Problems of identification in information systems

UDC 004. 932:531.87-028.23 **doi:** https://doi.org/10.20998/2522-9052.2026.2.01

Vitalii Serdechnyi, Olesia Barkovska, Andriy Kovalenko

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

CONTEXT-ADAPTIVE METHOD FOR OBJECT DETECTION IN VIDEO STREAMS

Abstract. **The work is devoted** to the development of a context-adaptive method for object detection in video streams that dynamically responds to environmental conditions. The relevance of the topic is explained by the need to increase the reliability of assistive systems for visually impaired people and other real-world applications, where variable weather and lighting conditions significantly reduce detection accuracy. **The subject** of the article is the study of multimodal fusion of acoustic, video, and LiDAR data for object recognition tasks. The goal of this paper is to propose and experimentally validate a method of adaptive preprocessing activation triggered by acoustic artifact classification. **The task** of this work is to analyze state-of-the-art preprocessing approaches (derain, defog, low-light enhancement), select appropriate acoustic classification models (e.g., PANNs, YAMNet), integrate LiDAR for spatial complementarity, and evaluate the impact of different preprocessing chains on detection metrics. **Methods** such as comparative analysis, experimental benchmarking of YOLO and DETR models, acoustic signal classification, and multimodal data fusion were applied. **The results of** the work include a confirmed increase in accuracy (mAP, Precision, Recall, IoU) and stability of detection under adverse conditions when using adaptive preprocessing pipelines, with YOLOv9m and YOLOv10m models showing the most balanced performance. **Further research** will focus on extending the model with full LiDAR integration, optimizing computational efficiency for mobile/embedded platforms, and scaling the approach for broader classes of environmental challenges such as fog, snow, and urban noise.

Keywords: context-adaptive detection; video preprocessing; acoustic analysis; object recognition; YOLO; multimodal system; assistance for visually impaired; low-light enhancement; derain; LiDAR.

Introduction

In modern intelligent object detection systems, increasing attention is given to the use of multimodal data, particularly from video cameras, laser rangefinders (LiDAR), and microphone arrays. Each of these sources has its own advantages and limitations: video streams provide a visual interpretation of the scene but are highly sensitive to changes in lighting and weather conditions; LiDAR ensures accurate depth estimation but is limited in recognizing object types; acoustic signals, in turn, contain contextual information about the environment that is difficult to obtain through visual means (Fig. 1). The complementary integration of these sources enhances system robustness to external influences, reduces the number of false detections, and enables the adaptation of processing to specific scene conditions.

Within this study, we propose an approach in which the classification of acoustic artifacts (e.g., rain, wind, traffic noise) serves as a trigger for the automatic selection of the appropriate video preprocessing method, while LiDAR data are used to improve the accuracy of spatial object localization. This approach represents a logical continuation of the trend of moving from isolated visual processing toward context-oriented adaptive multimodality.

State-of-the-art computer vision systems designed for real-world deployment often face degraded input quality from visual sensors due to the variability and unpredictability of the environment, particularly in outdoor scenarios, transportation infrastructure, or assistive systems for visually impaired people. Traditional approaches that rely solely on cameras demonstrate a significant drop in accuracy under atmospheric disturbances (fog, rain, snowfall), dynamic lighting (sudden sun/shadow transitions, glare), or complex scenes containing visual artifacts (reflections, shadows).

Therefore, there is a growing need for adaptive systems capable of automatically identifying the current scene context (e.g., presence of rain, fog, or low lighting levels) and dynamically adjusting preprocessing stages before feeding the data into the object detection model. The use of acoustic feature classification as an indicator of environmental conditions is one of the key tools for achieving such adaptability. The combination of video, audio, and LiDAR data enables the development of a context-aware system that demonstrates higher accuracy and performance in unpredictable real-world conditions. This makes the proposed context-adaptive method for object detection in video streams particularly relevant for real-world applications.

The complementary integration of audio sensors and LiDAR as sources of additional contextual information addresses this challenge through the use of:

– triggering functionality, where recognized acoustic artifacts (rain noise, wind, traffic) indicate adverse weather conditions that distort the quality of visual input data, initiating specialized filters (e.g., DeRain for rain, DeHaze for fog) to increase detection accuracy;

– automatic preprocessing adjustment, enabling adaptive tuning of filtering pipelines to environmental context;

– spatial complementarity, where LiDAR provides accurate distance information in scenarios where cameras fail (darkness, optical interference).

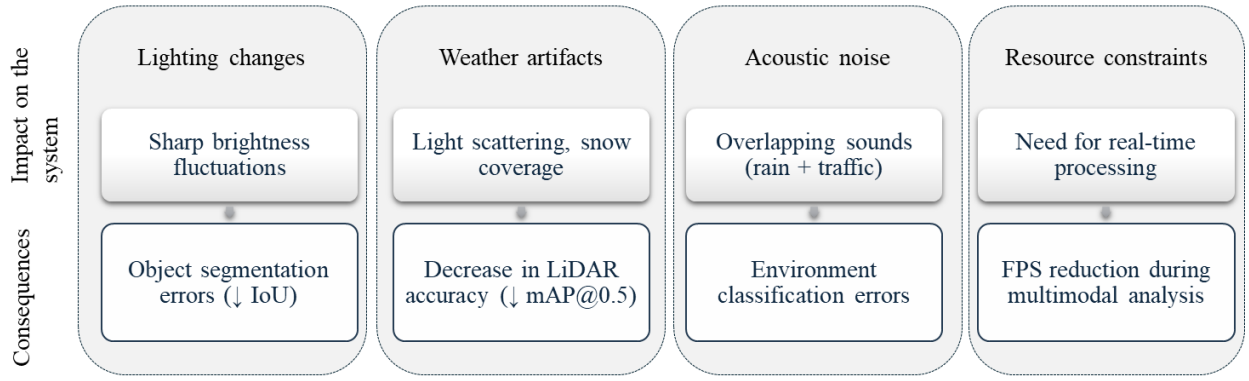


Fig. 1. Key challenges of multimodal systems

The integration of acoustic classification as a mechanism for dynamic activation of image enhancement filters makes it possible to increase object detection accuracy under challenging recording conditions, eliminate the need for manual intervention, and optimize the use of LiDAR through context-dependent data fusion. This approach paves the way toward the development of autonomous assistive systems capable of adapting to environmental changes without external commands.

Literature review and problem statement

To implement and study the preprocessing adaptation module under variable shooting conditions, a theoretical analysis of deraining methods was conducted according to the proposed criteria: image restoration accuracy, noise robustness, processing speed, architectural depth, real-time performance, and type of training (Table 1).

DerainNet is one of the first CNN-based deraining models, simple and fast, suitable for embedded systems. However, its accuracy is limited due to the lack of contextual analysis and multi-level processing. PReNet demonstrates significant improvement through recurrent processing and stage-wise loss. It provides good generalization but falls short of JORDER-E in contextual depth. JORDER-E ensures the highest quality by combining the deraining task with depth estimation. It delivers high accuracy and stability in complex backgrounds but is heavy and less suitable for real-time applications without optimization. EfficientDerain is specifically designed for speed and mobility. It is well-suited for real-time systems with limited resources, although its quality is somewhat inferior to JORDER-E and IADN. IADN is the most advanced among the reviewed methods, integrating attention mechanisms that allow better adaptation to complex rain structures. It represents a compromise between quality and computational cost. Median/Bilateral filtering are classical methods that provide low quality of object feature restoration, although they are fast. They are reasonable to use as baselines for comparison.

Table 1 – **Comparative analysis of deraining methods**

Method	Year	Method type	Architecture	Accuracy (PSNR, SSIM)	Noise robustness	Speed (FPS)	Key features
DerainNet [1]	2017	CNN, shallow	3 Conv. layers	PSNR ~27.3	Medium	High (real-time)	Simplest model; no complex contextual fusion
PReNet [2]	2019	Recurrent CNN	Multi-stage w/ residual	PSNR ~30.3; SSIM ~0.91	High	Medium	Recurrent processing + loss accumulation
JORDER-E [3]	2019	Multi-task CNN	Context aggregation + depth prediction	PSNR ~31.1; SSIM ~0.92	High	Moderate	Integrates depth map and guidance mask
Efficient Derain [4]	2020	Lightweight CNN	Encoder-decoder (Efficient)	PSNR ~29.7	Medium	High (mobile optimization)	Suitable for embedded systems
Median filter	-	Classical filter	-	Low (PSNR ~22)	Low	High	Simple; but aggressive, may lose details
Bilateral filter	-	Classical filter	Edge-aware	Medium (PSNR ~25)	Medium	Medium	Preserves edges; less effective against heavy rain

To evaluate the impact of preprocessing with respect to shooting conditions, real-time requirements, and object detection accuracy in a practical system, empirical testing should be conducted for a representative of each deraining method class: baseline (DerainNet), high-accuracy (IADN, JORDER-E), lightweight/fast (EfficientDerain), and classical methods (Median, Bilateral) as controls.

In the theoretical analysis of low-light image enhancement methods, the following criteria were considered: restoration accuracy, noise robustness, processing speed, architecture, and real-time performance (Table 2).

Zero-DCE is a flexible and efficient model that does not require paired “before/after” data, operates in real time, and preserves the naturalness of the image. It has a built-in mechanism for brightness and noise compensation.

LiteEnhanceNet demonstrates a very good balance between accuracy, visual quality, and speed. Thanks to

attention mechanisms, it performs better across scenes with variable illumination.

EnlightenGAN is one of the first GAN-based approaches for low-light enhancement. Although it produces natural-looking frames, it is resource-intensive and less suitable for real-time applications.

CLAHE (Contrast Limited Adaptive Histogram Equalization) is a classical and fast method, but it is prone to creating artificial contrasts and overexposure in certain regions.

Gamma correction is a basic fixed approach that improves overall brightness but does not adapt to heterogeneous lighting or noise. It is typically used as a simple preprocessor or baseline.

To evaluate the impact of image enhancement on YOLO object detection accuracy under low-light conditions and to select the optimal method for integration into the proposed context-adaptive system, it is advisable to test representatives of three categories: Zero-DCE (universal and lightweight, suitable for real time), LiteEnhanceNet (a deeper method with higher accuracy), and CLAHE or gamma correction as baselines.

For the theoretical analysis of image enhancement methods under foggy conditions, the following criteria were considered: restoration accuracy (PSNR, SSIM), noise robustness, processing speed, architectural complexity, and real-time applicability (Table 3).

Table 2 **– Comparative analysis of low-light image enhancement methods**

Method Name	Year	Method Type	Architecture	Accuracy (PSNR, SSIM)	Noise Robustness	Speed (FPS)	Features
Zero-DCE [5]	2020	Direct curve estimation (unsupervised)	Deep Curve Estimation (no GT)	PSNR ~24.0, SSIM ~0.83	High	High (~100 FPS)	Does not require GT, optimal for mobile devices
Lite Enhance Net [6]	2023	Lightweight supervised	Channel-spatial attention + feature fusion	PSNR ~25.8, SSIM ~0.87	High	High (~60 FPS)	Balanced trade-off between quality and speed, designed for real-time use
Enlighten GAN [7]	2019	GAN-based unsupervised	Generative Adversarial Network	PSNR ~22.7, SSIM ~0.80	Medium	Medium (~10–15 FPS)	Produces highly natural images, but computationally expensive
CLAHE	-	Classical method	Histogram equalization	PSNR ~18–22	Low	High (~150 FPS)	Suitable for high-contrast scenes, but prone to artifacts
Gamma correction	-	Classical method	Fixed nonlinear correction	PSNR ~19–21	Low	High	Simple, but does not account for scene context

Table 3 – **Comparative analysis of image enhancement methods under foggy conditions**

Method name	Year	Method type	Architecture	Accuracy (PSNR, SSIM)	Noise robustness	Speed (FPS)	Features
Dehaze Net [8]	2016	CNN (supervised)	Shallow CNN + feature extraction	PSNR ~26.9, SSIM ~0.82	Medium	High (30–50 FPS)	The first CNN for dehazing; uses transmission map
AOD-Net [9–10]	2017	All-in-One CNN	Lightweight end-to-end	PSNR ~27.5, SSIM ~0.84	Medium–High	High (40–60 FPS)	Without transmission map; suitable for mobile devices
Grid Dehaze Net [11]	2019	Deep CNN + attention	GridNet + multi-scale fusion	PSNR ~30.1, SSIM ~0.89	High	Moderate (~15–25 FPS)	Powerful attention-based architecture with better generalization

DehazeNet – one of the first CNN-based models that directly estimates the transmission map. Despite its limited depth, it demonstrates reasonable quality with high speed, making it suitable for resource-constrained systems.

AOD-Net (All-in-One Dehazing) – a lightweight, energy-efficient model that eliminates the need for estimating transmission or atmospheric light. Performs well in real time and is ideal for mobile applications. GridDehazeNet – a modern, powerful model with multi-level processing and attention mechanisms. Achieves high accuracy (PSNR >30), but requires considerable computational resources.

Residual Dehaze CNN – a recent model with deep residual blocks and local attention. Provides a balance between accuracy and speed, adapting well to variable fog conditions.

To evaluate the impact of image enhancement on YOLO-based object detection accuracy under foggy conditions and to select the optimal method for integration into the proposed context-adaptive system, it is recommended to conduct practical testing of the following methods: AOD-Net – as a baseline lightweight and fast model; GridDehazeNet – as a high-accuracy method for analyzing quality improvements; DehazeNet – assess the baseline effectiveness of CNNs under limited-resource conditions.

The concept of the proposed improvements to the process of detecting dynamic objects in video streams includes the use of audio signal processing and analysis, whereby atmospheric sounds such as rain, wind, and traffic noise are classified. Detecting noise of a specific class enables the automated application of the corresponding visual filter, which in turn improves the accuracy of object detection.

The diagram (Fig. 2) illustrates the typical life cycle of audio signal processing in Environmental Sound Recognition (ESR) systems and provides the rationale for selecting methods at each stage. The system input is a raw audio signal, typically in WAV format.

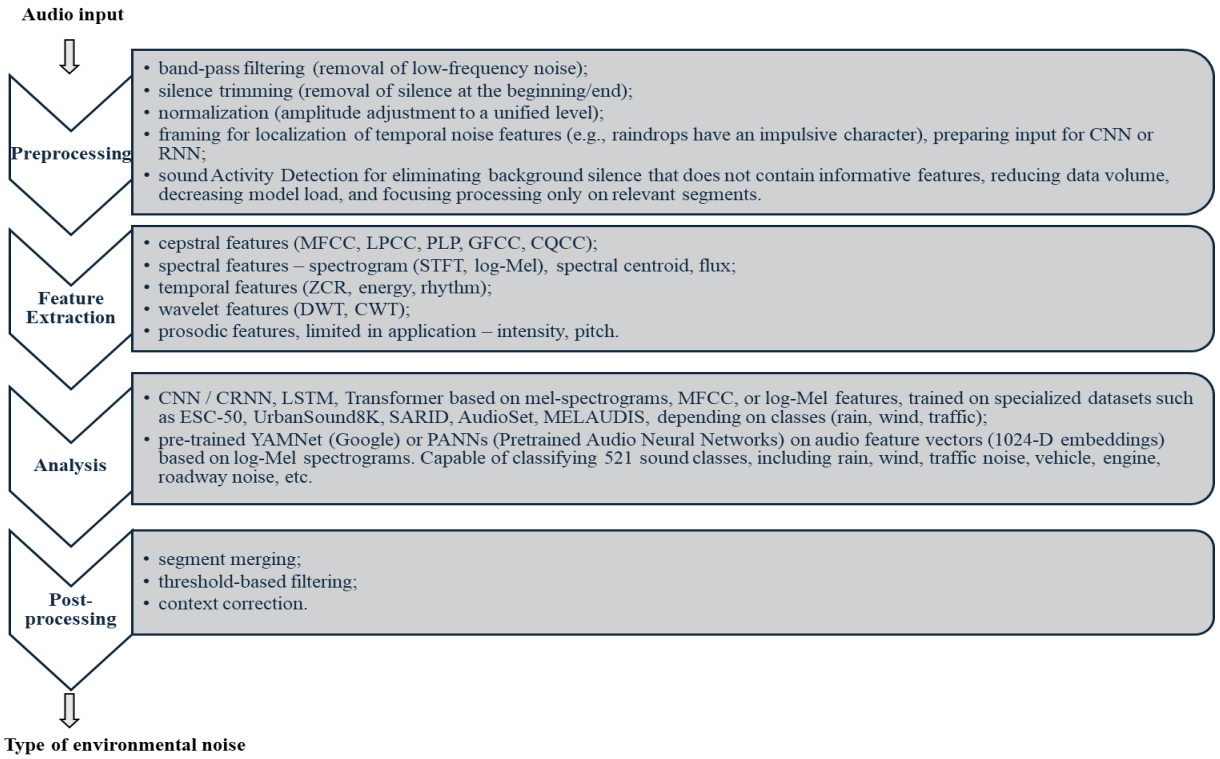


Fig. 2. Audio signal processing pipeline in the context of environmental sound recognition (ESR)

The first stage is preprocessing, which ensures high-quality preparation of the signal for subsequent analysis. This stage includes operations such as volume normalization, silence removal using Voice Activity Detection (VAD), band-pass filtering, and segmentation of the signal into frames with fixed windows (e.g., 25 ms with 10 ms overlap) to enable stable spectral analysis. The output of this stage is a cleaned and structured signal, which is then used at the next step – feature extraction – to compute characteristic acoustic descriptors (e.g., MFCC, CQCC, log-Mel spectrogram, Zero Crossing Rate). At this stage, the choice of feature type depends on the specific class of sound to be recognized. Thus, the quality of preprocessing directly affects the performance of the entire system. The methods adopted in this study (VAD, band-pass filtering, framing) are well-established in Environmental Sound Classification (ESC) tasks and ensure sufficient selectivity and consistency of the spectral structure under background noise typical for outdoor environments.

At the feature extraction stage, the prepared audio signal is transformed into a numerical representation suitable as input to machine learning models. The input is the preprocessed, framed signal considering activity regions. The output is a feature matrix that captures both spectral and temporal properties of the signal. The choice of features directly impacts classification accuracy, since different sound classes have distinct spectral structures and dynamics. This study employs a set of the most informative descriptors: cepstral coefficients (MFCC, GFCC, CQCC), spectral characteristics (log-Mel spectrogram, spectral centroid, roll-off), wavelet transforms (DWT, CWT), and, where necessary, prosodic features (intensity, rhythm). Such a selection is motivated by the need to achieve robust recognition of acoustic events in dynamic environments, particularly rain, wind, and traffic noise, which often share similar spectral components. Using a combined feature space increases the model's sensitivity to subtle spectral differences and enhances the discrimination of acoustic classes in noisy conditions.

The analysis stage is the core of an ESR system, where the audio signal is classified based on the extracted features. The module input is the spectral or cepstral representation of the signal (e.g., log-Mel spectrogram or MFCC), preserving the time-frequency structure. The output is either a probabilistic class distribution or a discrete decision about the type of acoustic event (e.g., rain, wind, traffic, speech). In this study, two approaches were applied: training deep models (CNN/CRNN, LSTM, or Transformer) on specialized datasets (ESC-50, AudioSet, SARID), and using a pretrained model, YAMNet (developed by Google), which operates on feature vectors and can classify 521 acoustic event classes, including weather and industrial noise [12–14]. This selection ensures high recognition accuracy under varying environmental conditions, as well as flexibility and adaptability to new classes without full retraining. Incorporating attention mechanisms or recurrent layers in CRNN and Transformer architectures enables temporal context modeling and focusing on informative regions of the spectrum, which is especially critical for distinguishing acoustically similar events.

An alternative powerful tool is the PANNs (Pretrained Audio Neural Networks) library, trained on AudioSet. It provides feature embeddings (e.g., 2048-dimensional CNN-based vectors) that can be directly input into a classifier or fine-tuned. PANNs support classification of a broad range of acoustic events, including rain, wind, engine, vehicle, car passing, and air

conditioner sounds, making them highly suitable for outdoor sound recognition tasks. They are also noted for their adaptability in low-resource scenarios.

The post-processing stage plays a crucial role in practical ESR applications. The module input consists of classification results obtained from the analysis stage – either probabilistic outputs or final frame-level predictions. The output is an aggregated decision, which may be represented as a single target class or as a trigger for launching specific actions in downstream modules. In this study, post-processing functions as a signaling mechanism (trigger) for activating the preprocessing module with the appropriate filter (e.g., derain, denoise). This approach ensures context-aware system responses to incoming acoustic events, enhancing adaptability and reducing the risk of false activations in real-world environments.

In this work, the pretrained models YAMNet and PANNs are investigated. The choice between them is determined by the priority between recognition accuracy and computational efficiency, which allows adapting the system to specific requirements regarding performance, mobility, or sensitivity to complex acoustic scenes (Table 4).

Table 4 – **Comparison of PANNs and YAMNet for environmental sound analysis**

Criterion	PANNs [15]	YAMNet [16]
Architecture	Deep convolutional networks (CNN14, Wavegram-CNN), optimized for audio	MobileNetV1 (lightweight architecture for fast inference)
Training data	AudioSet (527 classes, 5000+ hours of audio)	AudioSet (521 classes, 2 million annotations)
Accuracy (mAP)	0.439 (Wavegram-Logmel-CNN)	0.389
Supported classes	Rain, vehicle noise (Vehicle, Car horn, Engine)	Rain, traffic (Traffic noise, Car horn)
Transfer learning	Supports fine-tuning for specific sounds (e.g., different types of rain)	Capability of extracting embedding vectors for classification
Computational demands	High (CNN14: 81M parameters); for mobile apps optimized as E-PANNs (36% more efficient)	Low (3.7M parameters), ideal for real-time use
Integration	TensorFlow / PyTorch (implementation on GitHub)	Ready-to-use model in TensorFlow Hub

Within the study of environmental sound classification, an important step is the justified selection of audio datasets used for training or fine-tuning the models. The table presents a comparative description of the most common open datasets in terms of their suitability for recognizing rain noise, traffic sounds, as well as implicit estimation of the distance to the sound source (e.g., roads).

AudioSet is the most large-scale and universal source of training data – more than 2.1 million 10-second clips distributed across 527 classes, including detailed categories such as “Rain”, “Heavy rain”, “Vehicle”, “Car horn”, “Motorcycle”, etc. (Table 5). This dataset was used as the basis for training the YAMNet and PANNs models, making it a key element for the implementation and testing of the proposed approach. In turn, SARID is a narrowly specialized high-quality dataset, focused exclusively on the classification of rain with different intensity levels. It can be used for precise fine-tuning of models or for building a dedicated rain detection block; however, it has a limited size and does not include background sounds. MELAUDIS is a valuable resource for the analysis of traffic noise. It provides recordings with different acoustic perspectives (including information on road surface type and microphone distance), which can be useful for further refinement of the audio context.

Finally, ESC-50 and UrbanSound8K are baseline datasets that are often used for testing or benchmarking models. They include classes partially overlapping with this study (e.g., Rain, Engine, Car horn), but their size is significantly smaller and they do not cover a broad range of environments.

Table 5 **– Comparative table of acoustic datasets for environmental sound analysis [17]**

Dataset	Rain recognition	Traffic recognition	Distance to road	Key characteristics
SARID	Specialized: – different intensities (light rain, heavy rain); – clean recordings without background.	Not included	Not included	– Narrowly specialized; – high-quality recordings; – limited size.
AudioSet	Included: – classes "Rain", "Heavy rain"; – 28K+ samples; – background noise (wind, thunder).	Detailed: – sub-classes (car, motorcycle, brakes, horns); – 112K+ samples.	Implicit: through analysis of loudness/spectra	– 2.1M recordings (10 sec); – 527 classes; – hierarchical ontology.
MELA-UDIS	Partially: rain as background noise.	Optimal: – recordings at different distances; – different types of surfaces (asphalt, soil).	Present: – distance labels; – sound reflection effects.	– Focus on transport noise; – information about recording environment.
ESC-50 / Urban Sound8K	Basic: – class "Rain" (ESC-50); – 40 samples (limited).	Standard: – class "Car horn", "Engine" (UrbanSound8K); – background urban noise.	Absent	– Small size (5K samples); – standard for baseline testing.

Aims and tasks

The aim of this study is to develop a method for visual object detection in video streams that ensures improved accuracy and stability of system performance under variable environmental conditions by means of automatic selection of image preprocessing parameters based on the classification of acoustic artifacts and the use of spatial information from LiDAR sensors. To achieve this aim, the following objectives must be addressed:

– analyze methods for visual object recognition under different weather conditions;

– review methods for acoustic artifact classification;

– develop a model for visual object detection under variable external factors;

– conduct empirical analysis of video preprocessing methods depending on environmental conditions;

– provide a comparative evaluation of the efficiency of the proposed visual object detection method using the metrics F1-score, FPS, IoU, Precision, Recall, mAP@.5:.95, and mAP@0.5.

Materials and methods

The hypothesis of this study is that the integration of acoustic artifact classification as a trigger for the automatic activation of image enhancement filters will improve the accuracy of object detection under adverse weather conditions; ensure dynamic system adaptation without the need for manual intervention; and, in the longer term, enhance object localization through fusion with LiDAR data.

To achieve the anticipated outcomes – namely, improving the accuracy of object detection and recognition in challenging weather and lighting conditions (such as darkness, rain, fog, snow, or environmental noise pollution) – this work proposes a context-oriented approach to the adaptive preprocessing of input visual data. Unlike traditional systems that rely on standard or manually tuned filtering and enhancement parameters, the proposed model automates this selection by performing preliminary classification of the external environment, for instance, through acoustic analysis of the surroundings of a person using the intelligent assistance system for people with visual impairments. For example, the detection of rain noise automatically activates derain filtering of the visual data currently being fed into the system.

Let:

– Xv - raw video stream;

– Xa - input audio signal;

– $l_a = C_a(X_a) \epsilon \alpha_a$ - class of weather or noise conditions in the environment, determined by the audio classifier (e.g., rain, road noise, wind);

– $l_l = C_l(X_v) \epsilon \alpha_l$ - lighting level determined by video analysis or a light sensor (e.g., low-light or normal);

– $P(l_a, l_l)$ - module of context-adaptive video preprocessing that activates one or more filters: derain, defog, denoise, gamma correction, etc;

– $X_v^{`} = P(X_v, l_l, l_a)$ - video stream after preprocessing;

– dvision - distance to the object estimated based on the processed image;

– dLIDAR - distance to the object from LiDAR;

– α∈[0,1] - confidence coefficient of the visual module, determined based on the noise level;

– $d_i = \alpha \times d_{vision} + (1 - \alpha) \times d_{LIDAR}$ - combined distance estimation;

– $Y = D(X_v')$ - set of detected objects using the detection model (YOLOv11n).

Based on the above formalized hypothesis representation, the general functional model of the system is defined as:

$$Y = D(P(X_v, C_a(X_a), C_l(X_v)));$$

$$d_i = \alpha \times d_{vision} + (1 - \alpha) \times d_{LIDAR}.$$

The representation of the proposed method is shown in Fig. 3.

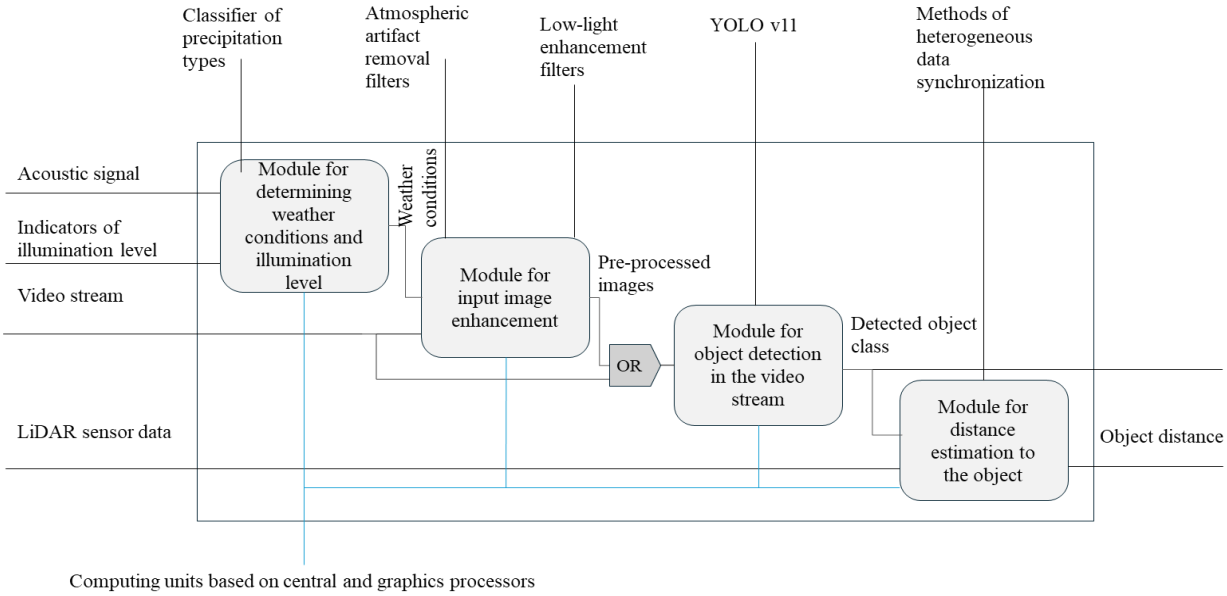


Fig. 3. Model of the context-adaptive method for object detection in video streams

In this study, the initial modules of the proposed method are examined, since the processing of LiDAR sensor data represents a further development of this research direction in the context of the Intelligent Assistance System for People with Visual Impairments [18].

The system architecture implements multi-level processing:

1. The audio analysis level (module C_a) is designed to classify acoustic artifacts (in particular, features of rain, road noise, and wind) using a neural network model based on audio features (e.g., MFCC, spectrogram CNN). The

output of this stage of the proposed method is $l_a \epsilon \alpha_a = \{rain, traffic, wind, none\}$

2. The lighting evaluation level (module C_l) serves to determine the degree of scene illumination. The output of this stage of the proposed method is $l_l \epsilon \alpha_a = \{low - light, normal\}$

3. The context-based activation level of preprocessing filters (module *P*) operates on the results of the two previous levels. Depending on the combination of l_a та l_l the corresponding filters are activated:

– Prain: derain (e.g., DerainNet, SPANet);

– Pfog: defog (e.g., DCPDN);

– Plow-light: gamma correction, histogram equalization, or Zero-DCE;

– Pdenoise: in case of road noise detection.

The processed video output is represented as $X_v^{`} = P(X_{v,} l_l, l_a)$.

The next and final level is the *Object Detection* stage based on YOLOv11n. It is used for the final detection of objects in the image, including their type, coordinates, and class.

Results of the study. Discussion of the results

To evaluate the baseline performance of the proposed object detection model under different environmental conditions, a series of experiments was conducted both without image preprocessing and after applying preprocessing pipelines. The results are presented in the table below for three typical scenes – sunny, rainy, and snowy under twilight conditions (Fig. 4).

The reported metric values characterize object detection quality (IoU, Precision, Recall, F1-score), performance (FPS), and overall accuracy (mAP@0.5 and mAP@0.5:0.95) (Table 6).

Analysis of the results presented in the table allows us to conclude that, despite the generally acceptable performance of modern YOLO-series models under good visibility conditions, the quality of object detection significantly decreases under worsening weather and lighting conditions. In particular:

- for YOLOv8m, the mAP@0.5 metric decreases from 0.348 (sunny weather) to 0.046 (winter twilight);

- for YOLOv9m, a similar decrease is observed – from 0.561 to 0.039, which represents more than 90% loss;

- the YOLOv10m model, which demonstrates the highest IoU (up to 0.918), also loses almost 40% of its F1-score between the sunny and winter scenes (0.804 → 0.538);

- the DETR model, despite its lower baseline efficiency, also demonstrates sensitivity to conditions, with Recall falling to 0.098 and F1-score to 0.107.

These results indicate a significant negative impact of atmospheric conditions (rain, snow, low illumination) on the quality of object detection. Particularly critical are the mAP@0.5:0.95 values, which for all models under twilight conditions do not exceed 0.039, indicating low accuracy of object boundary localization.

a b c

Fig. 4. Input frames for experiments: a – sunny weather, b – cloudy rainy weather, c – winter twilight

In this regard, there is a ned to apply additional methods of improving the quality of the input image, in particular such as derain, defog, denoise, and light-enhancement filtering. It is expected that the use of context-oriented preprocessing, which will be activated only under conditions of reduced visibility (for example, based on signs of rain or fog detected in audio), will allow improving the visibility of object contours; improving the quality of input images without excessive distortion; reducing the number of false detections; ensuring system adaptation without manual intervention. Thus, the next experimental stage should be aimed at studying the impact of preprocessing methods on detection accuracy under adverse conditions, with a detailed comparison of results before and after preprocessing (Table 6, Fig. 5).

Table 6 – **Object detection results without preprocessing**

Model	IoU	F1-score	Precision	Recall	FPS	mAP@0.5	mAP@0.5:0.95
Sunny weather							
YOLOv8m	0,893	0.807	0.8	0.815	19.0	0.348	0.278
YOLOv9m	0,887	0.862	0.855	0.87	14.0	0.561	0.458
YOLOv10m	0,911	0.804	0.854	0.759	16.0	0.447	0.406
YOLOv11s	0,88	0.775	0.754	0.796	9.0	0.404	0.335
DETR	0,8398	0.699	0.623	0.796	18.0	0.299	0.213
Cloudy rainy weather							
YOLOv8m	0.855	0.726	0.697	0.758	16.0	0.363	0.291
YOLOv9m	0.857	0.714	0.714	0.714	14.0	0.398	0.317
YOLOv10m	0.871	0.701	0.721	0.681	17.0	0.328	0.265
YOLOv11s	0.851	0.677	0.692	0.677	16.0	0.390	0.327
DETR	0.791	0.506	0.411	0.659	25.0	0.252	0.153
Winter twilight							
YOLOv8m	0.899	0.472	0.471	0.473	32.0	0.046	0.039
YOLOv9m	0.893	0.513	0.516	0.511	27.0	0.043	0.035
YOLOv10m	0.918	0.538	0.664	0.453	31.0	0.068	0.059
YOLOv11s	0.891	0.452	0.470	0.435	42.0	0.036	0.029
DETR	0.821	0.107	0.118	0.098	28.0	0.009	0.005

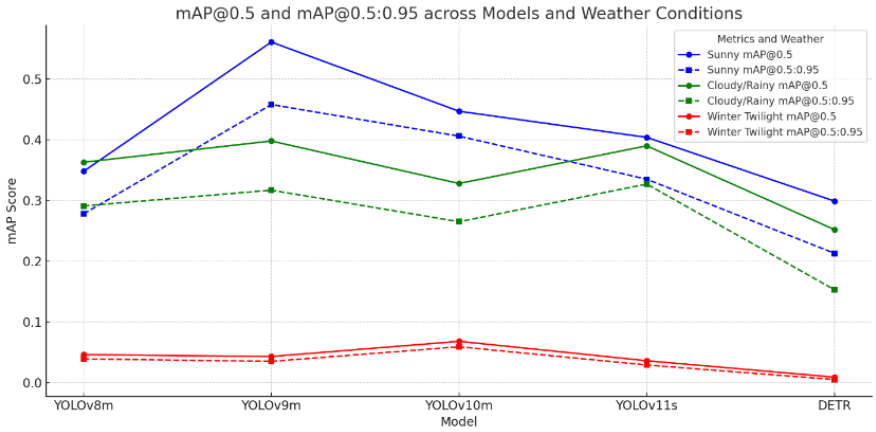


Fig. 5. mAP results under different weather conditions (without preprocessing)

As can be seen from the graph, YOLOv9m demonstrates the highest mAP@0.5 accuracy in sunny (0.561) and rainy (0.398) conditions. In twilight, all models show a noticeable drop in quality: the highest mAP@0.5 result is only 0.068 (YOLOv10m). DETR significantly lags behind in all conditions, especially in twilight (0.009). The study conducted a statistical analysis of the impact of various preprocessing chains on improving object detection accuracy and the speed of obtaining results. The analysis examined the influence of geometric and color correction methods of input images on the resulting object detection performance in images obtained under sunny weather conditions in an urban environment (Table 7).

Based on the average values of mAP@0.5 and mAP@0.5:0.95 for each of the tested preprocessing chains (including the option without preprocessing) for sunny weather, a graph was plotted (Fig. 6).

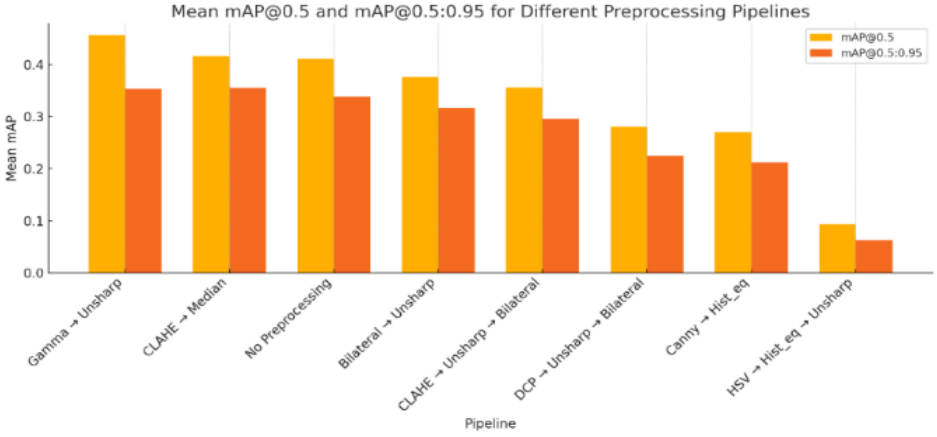
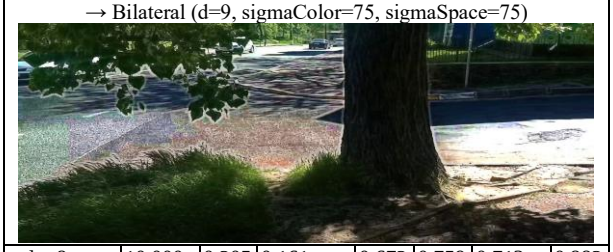
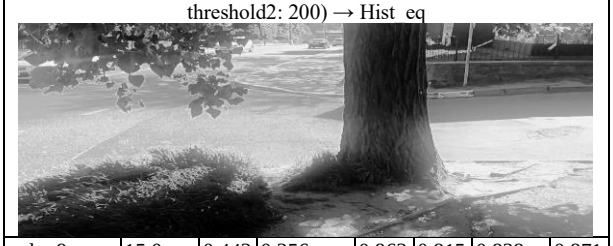


Fig. 6. Impact of different preprocessing strategies on object detection accuracy for urban scene images in sunny weather

Table 7 –**Statistical indicators of the impact of preprocessing methods on metrics for evaluating object detection accuracy and the speed of obtaining results. Sunny weather**

Model	FPS	mAP @0.5	mAP@ 0.5:0.95	Preci -sion	Re- call	F1- score	IoU
Without preprocessing							
yolo_8_m	19.000	0.348	0.278	0.800	0.815	0.807	0.893
yolo_9_m	14.000	0.561	0.458	0.855	0.870	0.862	0.887
yolo_10_m	16.000	0.447	0.406	0.854	0.759	0.804	0.911
yolo_11_s	9.000	0.404	0.335	0.754	0.796	0.775	0.880
detr	18.000	0.299	0.213	0.623	0.796	0.699	0.838
CLAHE (clipLimit=2, tileGridSize=8×8) → Unsharp (amount=1, kernelSize=5, sigma=1) → Bilateral (d=9, sigmaColor=75, sigmaSpace=75) 							
yolov8_m	12.000	0.293	0.244	0.740	0.759	0.766	0.893
yolov9_m	11.000	0.477	0.390	0.810	0.870	0.839	0.890
yolov10_m	13.000	0.460	0.383	0.860	0.796	0.827	0.890
yolov11_s	16.000	0.410	0.362	0.830	0.722	0.772	0.897
detr	26.000	0.139	0.100	0.485	0.611	0.541	0.822
Gamma (g: 1.25) → Unsharp (amount: 1, kernelSize: 5, sigma: 1) 							
yolo_8_m	18.000	0.422	0.334	0.833	0.818	0.885	0.885
yolo_9_m	15.000	0.587	0.468	0.870	0.870	0.870	0.880
yolo_10_m	18.000	0.458	0.370	0.843	0.796	0.819	0.884
yolo_11_s	19.000	0.414	0.333	0.782	0.796	0.789	0.883
detr	26.000	0.401	0.262	0.656	0.778	0.712	0.821
DCP (atmospheric_light_top_percent=0.1, omega=0.95, t0=0.1, window_size=15) → Unsharp (amount=1, kernelSize=5, sigma=1) → Bilateral (d=9, sigmaColor=75, sigmaSpace=75) 							
yolo_8_m	10.000	0.205	0.161	0.672	0.759	0.713	0.882
yolo_9_m	10.000	0.398	0.308	0.827	0.796	0.811	0.876
yolo_10_m	9.000	0.312	0.274	0.864	0.704	0.776	0.906
yolo_11_s	11.000	0.243	0.196	0.691	0.704	0.697	0.882
detr	24.000	0.247	0.183	0.532	0.611	0.569	0.820
CLAHE (clipLimit=2, tileGridSize=8x8) → Median (k=5) 							
yolo_8_m	17.000	0.386	0.331	0.815	0.815	0.805	0.898
yolo_9_m	14.000	0.568	0.481	0.839	0.870	0.855	0.889
yolo_10_m	15.000	0.455	0.413	0.894	0.778	0.827	0.888
yolo_11_s	19.000	0.461	0.386	0.827	0.796	0.805	0.882
detr	24.000	0.214	0.166	0.648	0.648	0.648	0.829
Canny (L2gradient: False, apertureSize: 3, threshold1: 100, threshold2: 200) → Hist_eq 							
yolo_9_m	15.0	0.443	0.356	0.863	0.815	0.838	0.871
yolo_8_m	16.0	0.289	0.226	0.741	0.796	0.768	0.872
yolo_10_m	17.0	0.254	0.209	0.927	0.704	0.800	0.881
yolo_11_s	21.0	0.217	0.189	0.809	0.704	0.752	0.872
detr	25.0	0.148	0.082	0.610	0.463	0.526	0.772
To_hsv (from: bgr, to: hsv) → Hist_eq → Unsharp (amount: 1, kernelSize: 5, sigma: 1) 							
yolo_9_m	14.0	0.112	0.075	0.523	0.426	0.469	0.816
yolo_8_m	18.0	0.106	0.074	0.524	0.407	0.458	0.818
yolo_10_m	17.0	0.118	0.091	0.786	0.407	0.537	0.835
yolo_11_s	19.0	0.075	0.049	0.465	0.370	0.412	0.822
detr	21.0	0.055	0.021	0.341	0.278	0.306	0.702
Bilateral (d: 9, sigmaColor: 75, sigmaSpace: 75) → Unsharp (amount: 1, kernelSize: 5, sigma: 1) 							
yolo_8_m	13.000	0.422	0.352	0.834	0.833	0.818	0.886
yolo_9_m	12.000	0.444	0.372	0.836	0.852	0.844	0.890
yolo_10_m	14.000	0.472	0.418	0.860	0.796	0.827	0.902
yolo_11_s	17.000	0.362	0.314	0.784	0.741	0.762	0.890
detr	26.000	0.181	0.128	0.654	0.630	0.642	0.805

The graph shows that the highest average accuracy (mAP@0.5) was achieved by the chains Gamma → Unsharp, CLAHE → Median, and Bilateral → Unsharp.

These methods significantly outperform the baseline accuracy without preprocessing and other chains.

The lowest values of mAP@0.5 and mAP@0.5:0.95 are observed for HSV → Hist_eq → Unsharp, meaning these methods hardly improve and sometimes even worsen the results.

Bilateral → Unsharp demonstrates the optimal balance between performance (FPS) and recognition quality.

The analysis of average FPS and IoU values for each preprocessing chain is shown in Fig. 7.

According to the constructed graphs, a number of generalizations can be made regarding the impact of preprocessing methods on IoU and FPS metrics, which

are critically important for real-time navigation support for visually impaired people. The highest IoU values (up to 0.902) were observed for the chains Gamma → Unsharp, CLAHE → Median, as well as in the baseline case without preprocessing, indicating the limited effectiveness of complex filter combinations in terms of spatial consistency of predictions. In contrast, the lowest IoU values (~0.702–0.822) were shown by certain DETR model configurations, especially when using color transformations (HSV → Hist_eq → Unsharp), which may be due to the high sensitivity of the architecture to uncontrolled changes in pixel distribution.

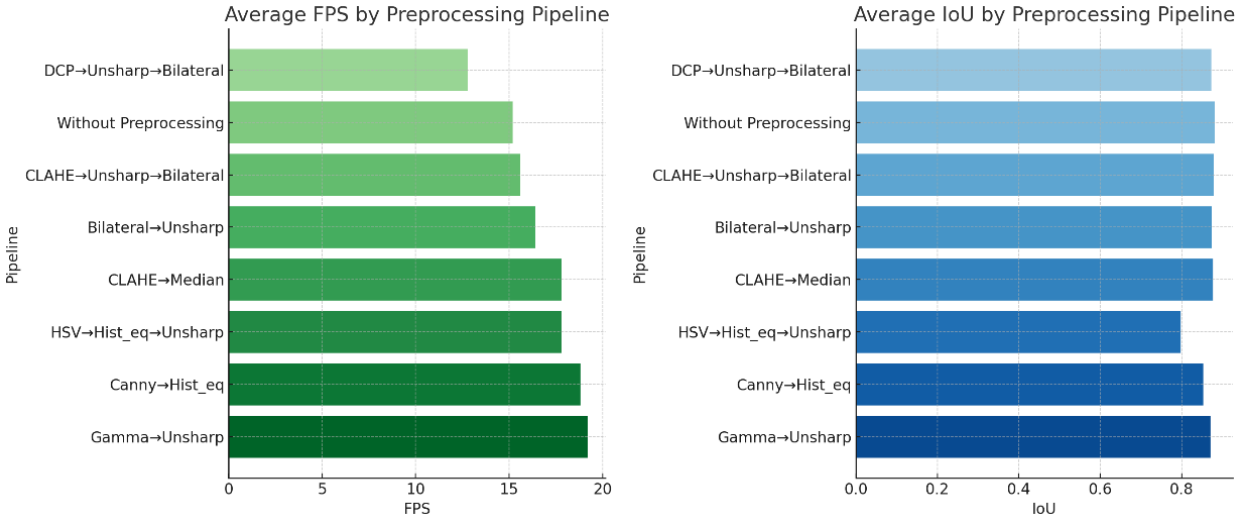


Fig. 7. Analysis of average FPS and IoU metrics for each of the 8 preprocessing chains for images in an urban environment under sunny weather

In terms of performance, the highest FPS values (~18–19 frames/s) were achieved by the chains Gamma → Unsharp, CLAHE → Median, and Canny → Hist_eq, making them suitable for use on mobile devices. At the same time, YOLOv11_s and DETR provided the highest speed, but the DETR model in most cases demonstrated lower accuracy (F1-score, IoU), making it less attractive for deployment in safety-critical tasks. Conversely, YOLOv9_m consistently showed the best compromise between speed and recognition quality, even in filtered variants – in particular, with Gamma → Unsharp, the model achieved mAP@0.5 = 0.587 and IoU = 0.880 at FPS = 15, which is quite acceptable for portable real-time systems. Thus, considering all factors, the optimal choice for developing a navigation support system for visually impaired people under bright daylight conditions is the YOLOv9_m model in combination with the preprocessing chain Gamma → Unsharp or CLAHE → Median, which provide high spatial accuracy, good Precision/Recall balance, and acceptable FPS for mobile device operation.

The obtained statistical results on the impact of preprocessing methods on object detection accuracy in an urban environment under cloudy rainy weather are presented below (Table 8).

Based on the average values of mAP@0.5 and mAP@0.5:0.95 for each of the tested preprocessing chains (including the variant without preprocessing) under cloudy rainy weather conditions, a graph was constructed (Fig. 8).

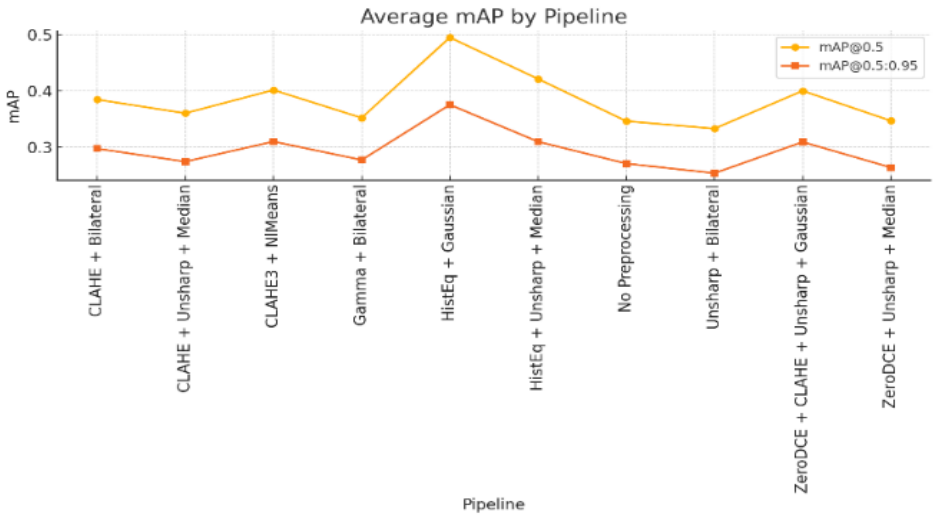
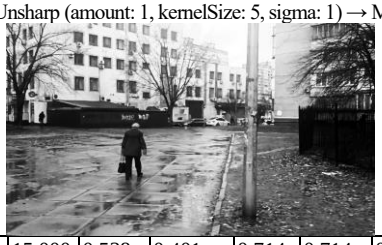
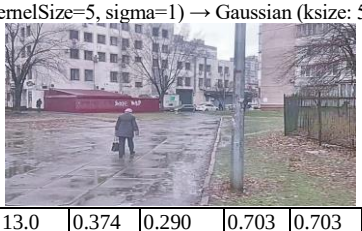


Fig. 8. Impact of different preprocessing strategies on object detection accuracy for images in an urban environment under cloudy rainy weather

Table 8 – **Statistical indicators of the impact of preprocessing methods on metrics for evaluating object detection accuracy and processing speed. Cloudy rainy weather**

Model	FPS	mAP @0.5	mAP@ 0.5:0.95	Preci-sion	Recall	F1-score	IoU
Without preprocessing							
yolo_8_m	16.000	0.363	0.291	0.697	0.758	0.726	0.855
yolo_9_m	14.000	0.398	0.317	0.714	0.740	0.714	0.857
yolo_10_m	17.000	0.328	0.265	0.721	0.681	0.701	0.871
yolo_11_s	16.000	0.390	0.327	0.643	0.692	0.677	0.851
detr	25.000	0.252	0.153	0.411	0.659	0.506	0.791
Clahe (clipLimit=2, tileGridSize=8x8) → Bilateral (d=9, sigmaColor=75, sigmaSpace=75) 							
yolo_8_m	10.000	0.388	0.306	0.643	0.791	0.709	0.847
yolo_9_m	10.000	0.411	0.313	0.691	0.714	0.703	0.842
yolo_10_m	12.000	0.430	0.350	0.716	0.692	0.704	0.874
yolo_11_s	14.000	0.461	0.384	0.645	0.659	0.652	0.862
detr	12.000	0.233	0.133	0.478	0.473	0.475	0.703
CLAHE (clipLimit: 2, tileGridSize: 8×8) → Unsharp (amount: 1.5, kernelSize: 5, sigma: 1) → Median (k: 5) 							
yolo_8_m	15.000	0.418	0.327	0.607	0.747	0.670	0.840
yolo_9_m	13.000	0.445	0.340	0.700	0.692	0.696	0.853
yolo_10_m	17.000	0.422	0.338	0.729	0.681	0.705	0.875
yolo_11_s	19.000	0.322	0.271	0.625	0.669	0.647	0.851
detr	25.000	0.194	0.094	0.493	0.396	0.439	0.735
Clahe (clipLimit: 3, tileGridSize: 8x8) → Nl_means (h: 10, hColor: 10, isColor: true, searchWindowSize: 21, templateWindowSize: 7) 							
yolo_8_m	3.000	0.389	0.321	0.631	0.716	0.670	0.865
yolo_9_m	4.000	0.481	0.357	0.680	0.725	0.702	0.844
yolo_10_m	3.000	0.359	0.301	0.750	0.662	0.703	0.881
yolo_11_s	6.000	0.555	0.449	0.648	0.648	0.648	0.852
detr	23.00	0.224	0.121	0.481	0.429	0.453	0.772
Unsharp (amount: 1.5, kernelSize: 5, sigma: 1) → Bilateral (d: 9, sigmaColor: 75, sigmaSpace: 75) 							
yolo_8_m	13.000	0.335	0.266	0.615	0.736	0.670	0.849
yolo_9_m	12.000	0.436	0.338	0.678	0.670	0.674	0.859
yolo_10_m	14.000	0.293	0.240	0.708	0.692	0.700	0.875
yolo_11_s	17.000	0.355	0.287	0.625	0.659	0.642	0.851
detr	25.000	0.245	0.136	0.471	0.538	0.503	0.775
Gamma (g: 1.25) → Bilateral (d: 9, sigmaColor: 75, sigmaSpace: 75) 							
yolo_8_m	14.000	0.338	0.257	0.689	0.703	0.696	0.842
yolo_9_m	11.000	0.449	0.347	0.695	0.725	0.710	0.841
yolo_10_m	12.000	0.327	0.305	0.735	0.703	0.719	0.868
yolo_11_s	13.000	0.367	0.305	0.630	0.670	0.649	0.854
detr	14.000	0.279	0.171	0.465	0.521	0.472	0.772
Hist_eq → Gaussian (ksize: 5x5, sigma: 0) 							
yolo_8_m	17.000	0.438	0.334	0.706	0.767	0.736	0.870
yolo_9_m	15.000	0.628	0.464	0.713	0.726	0.720	0.856
yolo_10_m	18.000	0.688	0.535	0.783	0.714	0.747	0.867
yolo_11_s	22.000	0.473	0.386	0.663	0.456	0.556	0.864
detr	26.000	0.249	0.158	0.405	0.437	0.525	0.782
Hist_eq → Unsharp (amount: 1, kernelSize: 5, sigma: 1) → Median (k: 5) 							
yolo_9_m	15.000	0.538	0.401	0.714	0.714	0.714	0.852
yolo_10_m	18.000	0.571	0.440	0.729	0.681	0.705	0.862
yolo_11_s	21.000	0.421	0.320	0.702	0.648	0.674	0.850
yolo_8_m	17.000	0.435	0.311	0.725	0.725	0.725	0.841
detr	26.000	0.142	0.076	0.374	0.374	0.374	0.753
Zero_dce → Unsharp (amount: 1, kernelSize: 5, sigma: 1) → Median (k: 5) 							
yolo_9_m	14.000	0.352	0.280	0.651	0.678	0.664	0.846
yolo_10_m	16.000	0.352	0.281	0.726	0.670	0.697	0.856
yolo_11_s	20.000	0.453	0.363	0.694	0.648	0.670	0.854
yolo_8_m	16.000	0.329	0.250	0.725	0.725	0.725	0.872
detr	4.000	0.248	0.144	0.429	0.429	0.436	0.774
Zero_dce → Clahe (clipLimit=2, tileGridSize=8×8) → Unsharp (amount=1, kernelSize=5, sigma=1) → Gaussian (ksize: 5×5, sigma: 0) 							
yolo_9_m	13.0	0.374	0.290	0.703	0.703	0.703	0.850
yolo_10_m	15.0	0.389	0.307	0.789	0.615	0.697	0.872
yolo_11_s	21.0	0.541	0.437	0.741	0.692	0.716	0.846
yolo_8_m	16.0	0.429	0.346	0.734	0.758	0.746	0.848
detr	4.0	0.268	0.165	0.417	0.714	0.526	0.773

The best results among all preprocessing chains were demonstrated by Hist_eq → Gaussian, providing the highest average values of mAP@0.5 = 0.591 and mAP@0.5:0.95 = 0.424. High performance was also shown by CLAHE → NLM (0.437 / 0.330) and Zero-DCE → CLAHE → Unsharp → Gaussian (0.452 / 0.374).

Thus, the combination of adaptive histogram equalization (CLAHE or Hist_eq) with smoothing filters (Gaussian or NLM) significantly improves both baseline accuracy (IoU ≥ 0.85) and extended accuracy (AP@0.5:0.95), which is critical for detecting objects of different scales and under partially obscured conditions (Fig. 9).

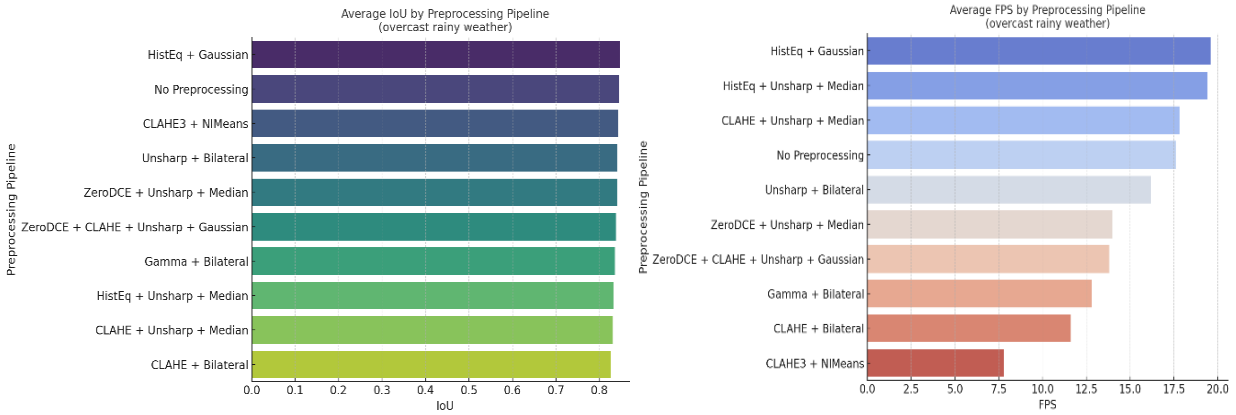


Fig. 9. Analysis of average FPS and IoU indicators
for each preprocessing chain for images in an urban environment under cloudy rainy weather

For the task of assisting visually impaired people outdoors, real-time system performance is a key criterion. The highest average FPS was achieved by Hist_eq → Gaussian (19.6 FPS), Hist_eq → Unsharp → Median (19.2 FPS), no preprocessing (15.4 FPS), and CLAHE → Unsharp → Median (16.2 FPS). In contrast, combinations with NLM and Bilateral (especially after CLAHE or Gamma) significantly reduce performance (down to ~10 FPS), which may be critical in field conditions.

The IoU metric reflects spatial accuracy of segmentation or detection. All strategies, except for Hist_eq → Gaussian, showed an average IoU in the range of 0.84–0.86, indicating stable localization quality. However, Hist_eq → Gaussian reached IoU = 0.862, while CLAHE → NLM and CLAHE → Unsharp → Median also achieved IoU ≥ 0.858. In the no-preprocessing variant, IoU was only 0.857, indicating minimal gains in localization.

The most balanced preprocessing strategies under cloudy rainy weather are:

- Hist_eq → Gaussian – provides maximum accuracy (mAP), high performance (FPS), and precise localization (IoU). It works particularly well with the YOLOv10_m model, which achieved the highest results (mAP@0.5 = 0.688, mAP@0.5:0.95 = 0.535) at 18 FPS;

- Zero-DCE → CLAHE → Unsharp → Gaussian: demonstrates high accuracy with YOLOv11_s (mAP@0.5 = 0.541) and YOLOv8_m, maintaining IoU > 0.84, though FPS performance is slightly lower (~16–21);

- CLAHE → Unsharp → Median: delivers stable results across all YOLOv models, especially YOLOv10_m and YOLOv9_m, where average IoU ≈ 0.86 and FPS remains at 15–18.

From the model perspective, YOLOv10_m and YOLOv9_m show the best balance between accuracy, localization, and speed across almost all preprocessing chains. YOLOv8_m is also stable, though slightly behind in accuracy. YOLOv11_s demonstrates high FPS but overall lower accuracy, performing better after aggressive contrast enhancement (e.g., CLAHE, Hist_eq).

DETR consistently performs the worst in all metrics except FPS and is not recommended for real-time use or low-light conditions.

For real-time operation, it is critical to avoid combinations with NLM and overly heavy filters like Bilateral with large parameters, as they reduce FPS to 10 or less, which is unacceptable for dynamic street environments.

The least effective strategy is Unsharp → Bilateral, which yields low accuracy (mAP ≤ 0.33), slow processing, and no advantages in localization. Similarly, the no-preprocessing variant performed worse than most simple chains using CLAHE or Hist_eq.

The obtained statistical results on the impact of preprocessing methods on object detection accuracy in urban environments at night are presented below (Table 9).

The diagram (Fig. 10) shows that simple Bilateral smoothing proved to be the most effective for nighttime scenes. More aggressive chains with multiple filters did not provide improvements and in some cases even reduced accuracy.

The highest FPS was achieved by the chain Hist_eq → Unsharp → Median (~34 frames/s), which indicates its high speed despite lower accuracy. The slowest chain was Gamma → CLAHE → Unsharp → Bilateral. Thus, if real-time speed is a priority, Hist_eq → Unsharp → Median should be used, though with accuracy limitations (Fig. 11).

All chains demonstrated high IoU values (~0.89–0.91). The highest average IoU was obtained with Hist_eq → Unsharp → Median.

That is, regardless of preprocessing, the quality of prediction overlays on GT masks remained stable. This may indicate the consistency of geometric object localization in YOLO.

The YOLOv10_m model is the most stable across different preprocessing chains.

The YOLOv11_s model shows the highest performance (FPS) but the lowest accuracy.

Table 9 – **Statistical indicators of the impact of preprocessing methods on metrics for evaluating object detection accuracy and processing speed. Nighttime**

Model	FPS	mAP @0.5	mAP@ 0.5:0.95	Preci-sion	Re-call	F1-score	IoU
Without preprocessing							
yolo_8_m	32.000	0.046	0.039	0.471	0.473	0.472	0.899
yolo_9_m	27.000	0.043	0.035	0.516	0.511	0.513	0.893
yolo_10_m	31.000	0.068	0.059	0.664	0.453	0.538	0.918
yolo_11_s	42.000	0.036	0.029	0.470	0.435	0.452	0.891
detr	28.000	0.009	0.005	0.118	0.098	0.107	0.821
Bilateral (d: 9, sigmaColor: 75, sigmaSpace: 75)							
							
yolo_8_m	16.000	0.047	0.040	0.439	0.460	0.449	0.900
yolo_9_m	15.000	0.052	0.043	0.547	0.515	0.530	0.894
yolo_10_m	17.000	0.060	0.051	0.699	0.443	0.542	0.916
yolo_11_s	19.000	0.045	0.038	0.491	0.428	0.457	0.894
detr	29.000	0.017	0.010	0.183	0.209	0.195	0.794
Hist_eq → Unsharp (amount: 1, kernelSize: 5, sigma: 1) → Median (k: 5)							
							
yolo_10_m	36.000	0.030	0.027	0.528	0.350	0.421	0.917
yolo_9_m	30.000	0.017	0.014	0.326	0.372	0.347	0.907
yolo_8_m	33.000	0.020	0.017	0.320	0.359	0.339	0.902
yolo_11_s	44.000	0.024	0.019	0.292	0.348	0.318	0.902
detr	27.000	0.004	0.002	0.154	0.090	0.113	0.821
Gamma (g: 1.3) → Clahe (clipLimit: 2, tileGridSize: 8x8) → Unsharp (amount: 1, kernelSize: 5, sigma: 1) → Bilateral (d: 9, sigmaColor: 75, sigmaSpace: 75)							
							
yolo_11_s	20.000	0.027	0.023	0.365	0.389	0.377	0.896
yolo_10_m	14.000	0.038	0.033	0.563	0.428	0.486	0.908
yolo_9_m	15.000	0.029	0.024	0.404	0.498	0.446	0.897
yolo_8_m	15.000	0.027	0.023	0.334	0.389	0.360	0.899
detr	22.000	0.013	0.008	0.115	0.123	0.119	0.802

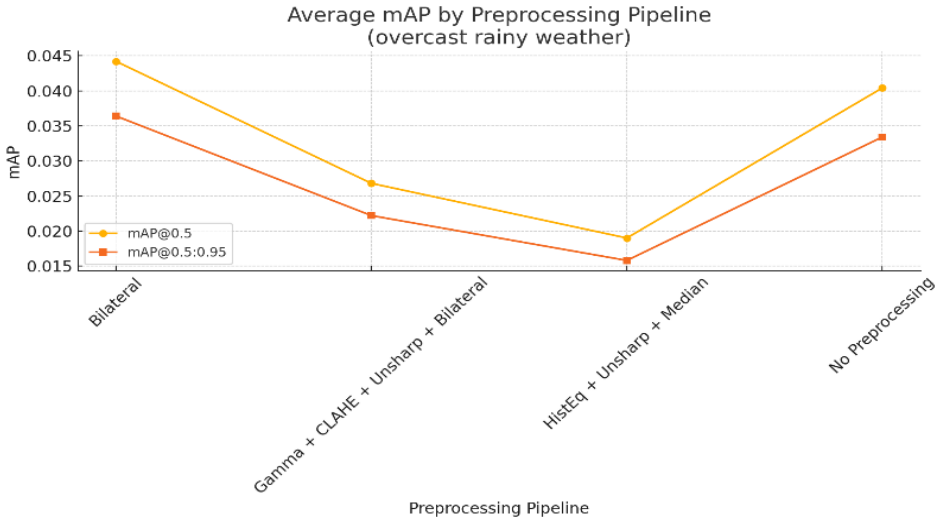


Fig. 10. Impact of different preprocessing strategies on object detection accuracy for images in an urban environment at night

DETR – although it supports high speed – catastrophically fails in terms of accuracy and is unsuitable for use without further adaptations.

The best compromise between speed and accuracy is achieved by the YOLOv10_m model with Bilateral preprocessing.

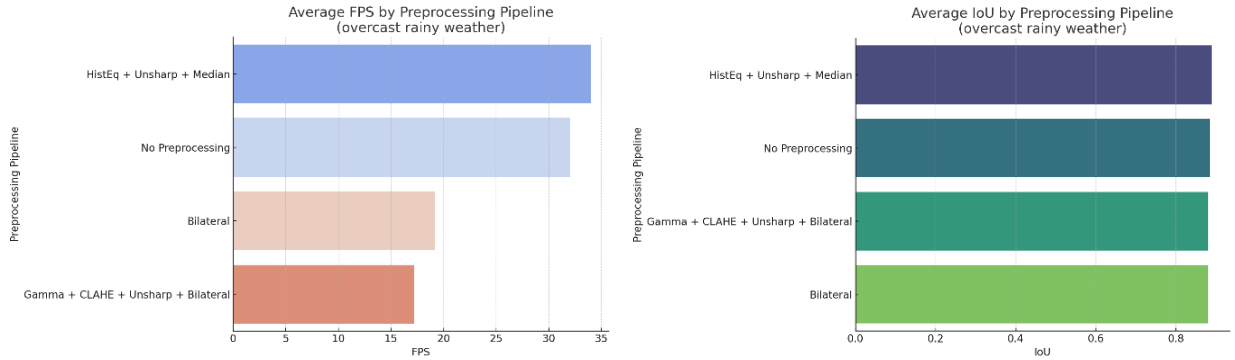


Fig. 11. Analysis of average FPS and IoU values for each preprocessing chain for images in an urban environment at night

Conclusion

The scientific novelty of the proposed method lies in the integration of audio analysis as an additional information channel for selecting the video preprocessing method; dynamic management of the processing pipeline based on recognized context; and adaptability to variable external factors without direct user intervention.

The practical novelty and significance of the proposed method consist in addressing the technical challenges faced by modern object detection systems operating under changing external conditions.

A context-adaptive video stream processing method has been proposed, where acoustic features of the environment are used as triggers for the activation of preprocessing filters. This makes it possible to improve the accuracy of object detection in complex weather and lighting conditions.

A comparative analysis of preprocessing chains including CLAHE, Gamma correction, Bilateral filtering, Median filtering, and others was carried out. It was found that the greatest improvement in accuracy (mAP, F1-score), with acceptable performance (FPS), is achieved by combinations such as Gamma → Unsharp, CLAHE → Median, and Hist_eq → Gaussian (particularly in rainy conditions). The YOLOv10_m model demonstrated the highest stability and efficiency across all scenarios (rain, night, day), showing the best compromise between speed (up to 31 FPS) and localization accuracy (IoU ~ 0.91). The DETR model, despite good performance, showed the lowest accuracy metrics and is not recommended for use in real-time or safety-critical applications.

The results confirm the feasibility of a multimodal approach, where the combination of audio, video, and LiDAR data enhances the reliability of computer vision systems without the need for manual adaptation to shooting conditions.

Conflicts of interest. The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Use of artificial intelligence. The authors confirm that they did not use artificial intelligence technologies when creating the current work.

REFERENCES

1. Yao, Y., Shi, Z., Hu, H., Li, J., Wang, G.; and Liu, L. (2023), "GSDerainNet: A Deep Network Architecture Based on a Gaussian Shannon Filter for Single Image Deraining", *Remote Sens*, 2023, vol. 15, doi: https://doi.org/10.3390/rs15194825
2. Pourali, A., Boukani, A. and Khazaei, H. (2025), "PreNeT: Leveraging Computational Features to Predict Deep Neural Network Training Time", *Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering*, pp. 81–91, doi: https://doi.org/10.1145/3676151.3719373
3. Yu, Yi, Yang, W., Tan, Y.–P. and Kot, A. C. (2022), "Towards Robust Rain Removal Against Adversarial Attacks: A Comprehensive Benchmark Analysis and Beyond", *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR), IEEE, pp. 6003–6012, doi: https://doi.org/10.1109/CVPR52688.2022.00592
4. Guo, Q., Sun, J., Juefei-Xu, F., Ma, L., Xie, X., Feng, W., Liu, Y., and Zhao, J. (2021), "Efficient De Rain: Learning Pixel-Wise Dilation Filtering for High-Efficiency Single-Image Deraining", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 1487–1495, doi: https://doi.org/10.1609/aaai.v35i2.16239
5. Mu, W., Liu, H., Chen, W. and Wang, Y. (2022), "A More Effective Zero-DCE Variant: Zero-DCE Tiny", *Electronics*, vol. 11, no. 17, doi: https://doi.org/10.3390/electronics1117275 0
6. Zhang, S., Zhao, S., An, D., Li, D. and Zhao R. (2024), "LiteEnhanceNet: A Lightweight Network for Real-Time Single Underwater Image Enhancement", *Expert Systems with Applications*, vol. 240, Apr. 2024, article number 122546, doi: https://doi.org/10.1016/j.eswa.2023.122546
7. Jiang, Y., Gong, X., Liu, D., Cheng, Yu, Fang, C. and Shen X. (2021), "EnlightenGAN: Deep Light Enhancement Without Paired Supervision, *IEEE Trans. on Image Proc.*, vol. 30, pp. 2340–2349, doi: https://doi.org/10.1109/TIP.2021.3051462
8. Ullah, H., Muhammad, K., Irfan, M., Anwar, S., Sajjad, M. and Imran, A. S. (2021), "Light-DehazeNet: A Novel Lightweight CNN Architecture for Single Image Dehazing", *IEEE Transactions on Image Processing*, vol. 30, pp. 8968–8982, doi: https://doi.org/10.1109/TIP.2021.3116790
9. Zhang, L., Zhao, J., Lang, Z. and Fang L. (2024), "Vehicle detection algorithm for foggy based on improved AOD-Net", *Transactions of the Institute of Measurement and Control*, vol. 46, issue 14, pp. 2696–2705, doi: https://doi.org/10.1177/01423312241248490

10. Guo, Z., Zhang, X. and Yu, S. (2024)," Image Defogging Based on Improved AOD-Net Network ", *Image Processing, Electronics and Computers*, IOS Press, pp. 211–222, doi: https://doi.org/10.3233/ATDE240472
11. Liu, X., Shi, Z., Wu, Z., Chen, J. and Zhai, G. (2023), "GridDehazeNet+: An Enhanced Multi-Scale Network With Intra-Task Knowledge Transfer for Single Image Dehazing", *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 870–884, doi: https://doi.org/10.1109/TITS.2022.3210455
12. Kholiev, V., and Barkovska, O. (2023), "Comparative analysis of neural network models for the problem of speaker recognition", *Innovative Technologies and Scientific Solutions for Industries*, vol. 2 (24), pp. 172–178, doi: https://doi.org/10.30837/ITSSI.2023.24.172
13. Barkovska, O., Holovchenko, O., Storchai, D., Kostin, A., and Lehezin, N. (2025), "Investigation of computer vision techniques for indoor navigation systems", *Innovative Technologies and Scientific Solutions for Industries*, vol. 2 (32), pp. 5–15, doi: https://doi.org/10.30837/2522-9818.2025.2.005
14. Tsalera, E., Papadakis, A. and Samarakou, M. (2021), "Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning", *Journal of Sensor and Actuator Networks*, vol. 10, no. 4, doi: https://doi.org/10.3390/jsan10040072
15. Arshdeep, S., Haohe, L. and Plumbley, M. D. (2023), "E-PANNs: Sound Recognition Using Efficient Pre-Trained Audio Neural Networks", *INTER-NOISE and NOISE-CON Congress and Conference Proceedings*, vol. 268, no. 1, pp. 7220–7228, doi: https://doi.org/10.3397/IN_2023_1083
16. Valliappan, N. Harihara, Pande, S. D. and Vinta, S. R. (2024), "Enhancing Gun Detection with Transfer Learning and YAMNet Audio Classification", *IEEE Access*, vol. 12, pp. 58940–58949, doi: https://doi.org/10.1109/ACCESS.2024.3392649
17. Turskis, T., Teleiša, M., Buckiūnaitė, R. and Čalnerytė, D. (2023), "Mixed-type data augmentations for environmental sound classification", *IVUS 2023: Information society and university studies 2023*, CEUR workshop proc. of the 28th int. conf. on information society and university studies (IVUS 2023), Kaunas, Lithuania, May 12, 2023, CEUR-WS, 3575, pp. 184–194, available at: https://ceur-ws.org/Vol-3575/Paper20.pdf
18. Barkovska, O. and Serdechnyi, V. (2024), "Intelligent Assistance System for People with Visual Impairments", *Innovative Technologies and Scientific Solutions for Industries*, vol. 2(28), pp. 6–16, doi: https://doi.org/10.30837/2522-9818.2024.28.006

Received (Надійшла) 10.12.2025
Accepted for publication (Прийнята до друку) 18.03.2026

ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

Сердечний Віталій Сергійович – аспірант кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Vitalii Serdechnyi – PhD student of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: vitalii.serdechnyi@nure.ua; ORCID Author ID: http://orcid.org/0009-0007-8828-5803;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57210288841.

Барковська Олеся Юріївна – кандидат технічних наук, доцент, доцент кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Olesia Barkovska – Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: olesia.barkovska@nure.ua; ORCID Author ID: http://orcid.org/0000-0001-7496-4353;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=24482907700.

Коваленко Андрій Анатолійович – доктор технічних наук, професор, завідувач кафедри електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна;
Andriy Kovalenko – Doctor of Technical Sciences, Professor, Head of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine;
e-mail: andriy.kovalenko@nure.ua; ORCID Author ID: https://orcid.org/0000-0002-2817-9036;
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=56423229200.

Контекстно-адаптивний метод детекції об'єктів у відеопотоках

В. С. Сердечний, О. Ю. Барковська, А. А. Коваленко

Анотація. Робота присвячена розробці контекстно-адаптивного методу детекції об'єктів у відеопотоках, який динамічно реагує на умови навколишнього середовища. Актуальність теми пояснюється необхідністю підвищення надійності асистивних систем для людей із порушеннями зору та інших прикладних сфер, де змінні погодні та світлові умови суттєво знижують точність розпізнавання. **Предметом** є дослідження мультимодальної ф'южн-інтеграції акустичних, відео- та LiDAR-даних для задач розпізнавання об'єктів. **Метою роботи є** пропозиція та експериментальна валідація методу адаптивної активації попередньої обробки, що ініціюється класифікацією акустичних артефактів. **Завданнями** дослідження є аналіз сучасних підходів до попередньої обробки (дерейн, дефог, покращення зображень при низькому освітленні), вибір моделей акустичної класифікації (наприклад, PANNs, YAMNet), інтеграція LiDAR для просторової комплементарності та оцінка впливу різних ланцюжків препроцесінгу на метрики детекції. Використано **методи** порівняльного аналізу, експериментального тестування моделей YOLO та DETR, класифікації акустичних сигналів і мультимодальної ф'южн-інтеграції даних. **Результати** роботи включають підтверджене підвищення точності (mAP, Precision, Recall, IoU) та стабільності детекції в несприятливих умовах при використанні адаптивних пайплайнів попередньої обробки, причому моделі YOLOv9m і YOLOv10m продемонстрували найбільш збалансовані показники. **Подальші дослідження** будуть спрямовані на повну інтеграцію LiDAR, оптимізацію обчислювальної ефективності для мобільних та вбудованих платформ і масштабування підходу для ширшого класу середовищних викликів, таких як туман, сніг та міський шум.

Ключові слова: контекстно-адаптована детекція; препроцесінг відео; акустичний аналіз; об'єктне розпізнавання; YOLO; мультимодальна система; супровід незрячих; покращення в темряві; антидощ; LiDAR.