

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЗАЛІЗНИЧНОГО
ТРАНСПОРТУ

Кваліфікаційна наукова праця
на правах рукопису

Садовников Борис Ігорович

УДК 621.39:004.6:004.9:004.272

ДИСЕРТАЦІЯ
МЕТОДИ КЛАСИФІКАЦІЇ НЕЛІНІЙНО ЗМІНЮВАНИХ ОБРАЗІВ У
ВІДЕОПОТОЦІ В УМОВАХ СКЛАДНОЇ ЗАВАДОВОЇ ОБСТАНОВКИ

Спеціальність 172 – Телекомунікації та радіотехніка

Галузь знань 17 – Електроніка та телекомунікації

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

 Б.І. Садовников

Науковий керівник:
Жученко Олександр Сергійович,
кандидат технічних наук, доцент

Харків – 2025

АНОТАЦІЯ

Садовников Б.І. Методи класифікації нелінійно змінюваних образів у відеопотоці в умовах складної завадової обстановки. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії (PhD) за спеціальністю 172 – Телекомунікації та радіотехніка. – Український державний університет залізничного транспорту, Україна, Харків, 2025.

У дисертаційній роботі вирішується актуальне науково-технічне завдання підвищення достовірності та ефективності виявлення і класифікації нелінійно змінюваних образів у відеопотоці в умовах впливу зосереджених завад, за допомогою розробки методів, моделей та алгоритмів обробки, векторизації, нейронної класифікації та трекінгу, здатних забезпечувати стабільну роботу в реальному часі та масштабованість до зміни параметрів сцени.

Об’єктом дослідження є процес виявлення і класифікації образів у відеопотоці в складних завадових умовах при обмежених обчислювальних можливостях.

Предметом дослідження є методи і моделі обробки, детекції, класифікації та трекінгу образів у відеопотоці, спрямовані на підвищення достовірності та ефективності пошуку та розпізнавання образів у відеопотоці в складних завадових умовах.

Метою дисертаційної роботи є підвищення точності, стійкості та обчислювальної ефективності виявлення і класифікації нелінійно змінюваних образів у відеопотоці в умовах впливу зосереджених завад, шляхом розробки і впровадження методів і моделей на основі нейронних мереж для застосування на ресурсно обмежених пристроях у режимі реального часу.

У **вступі** обґрунтовано актуальність підвищення достовірності та ефективності виявлення і класифікації об’єктів у відеопотоці в умовах впливу зосереджених завад та обмежених обчислювальних ресурсів. Зазначено, що стандартні алгоритми глибокого навчання демонструють високу точність лише в

умовах стабільного фону та потужної апаратної підтримки, що унеможливило їх застосування в умовах впливу зосереджених завад. Наведено наукову новизну роботи, яка полягає у розробці методів, імітаційно-математичних моделей і алгоритмів детекції, класифікації та трекінгу нелінійно змінюваних образів на основі гібридного підходу, що поєднує класичну міжкадрову обробку з використанням згорткових нейронних мереж. Визначено практичне значення отриманих результатів для підвищення завадостійкості, зниження затримок обробки та зменшення енергоспоживання при реалізації методів на пристроях з обмеженими ресурсами. Надано інформацію про особистий внесок здобувача та перелік публікацій з результатами дисертаційного дослідження.

Перший розділ присвячений аналізу сучасних підходів і методів виявлення, класифікації та трекінгу об'єктів у відеопотоці, з урахуванням впливу зосереджених завад і обмежених обчислювальних ресурсів. Розглянуто класичні підходи детекції руху (віднімання фону, оптичний потік, міжкадрова різниця), глибинні згорткові архітектури (YOLO, SSD, Faster R-CNN), а також гібридні моделі, що поєднують переваги обох груп. Проведено класифікацію методів за критеріями точності, продуктивності, стійкості до завад та придатності до реалізації в умовах реального часу. Обґрунтовано, що попри високу точність, глибинні моделі є ресурсоемними, тоді як класичні методи – є менш точними, але ефективними на пристроях із низьким енергоспоживанням. За результатами аналізу обґрунтовано доцільність розробки нового методу, який поєднує швидкодію класичних підходів і точність глибинних моделей, адаптованих до роботи в умовах завад і нестабільного фону.

У **другому розділі** представлено порівняльний аналіз існуючих методів виявлення та класифікації об'єктів у відеопотоці, що покладено в основу розробленого гібридного методу DeltaTrack. Запропонований підхід поєднує переваги класичного виявлення змін на основі трьохкадрової міжкадрової різниці (дельта) із точністю згорткової нейронної мережі MobileNetV1, яка використовується для класифікації виділених регіонів. До складу методу також входять морфологічна обробка, подвійне порогоування та трекінг на основі алгоритму MOSSE. Метод DeltaTrack орієнтовано на роботу в реальному часі на пристроях з обмеженими ресурсами, з мінімальним споживанням відеопам'яті та

навантаженням на GPU. У розділі обґрунтовано вибір базового підходу, побудовано покроковий алгоритм реалізації, запропоновано імітаційну модель і структурну схему. Проведено експериментальне порівняння з методами SSD (представник глибоких CNN) та MoG (класичний підхід), що підтвердило ефективність DeltaTrack за критеріями швидкодії, точності та ощадності ресурсів. Доведено його придатність до використання в умовах змінного фону, завад і обмежень, характерних для мобільних та вбудованих відеоаналітичних систем.

У **третьому** розділі представлено формалізовану імітаційну модель методу DeltaTrack, яка описує повний цикл обробки відеопотоку з урахуванням параметрів сцени, динаміки фону та обмежень обчислювальних ресурсів. Модель побудовано у вигляді послідовності математичних операторів, що забезпечує можливість аналітичного аналізу, адаптації до змінних умов середовища та програмної реалізації на малопотужних пристроях. Результати аналітичного профілювання доповнено експериментальним моделюванням, що дозволило оцінити часові витрати, точність детекції та стабільність роботи моделі в різних умовах. Експериментально підтверджено, що застосування формалізованої моделі DeltaTrack забезпечує стабільну роботу в реальному часі без потреби у великій кількості ресурсів GPU. Встановлено, що основне навантаження припадає на морфологічну обробку та класифікацію, а ключові параметри моделі впливають на її масштабованість і продуктивність.

У **четвертому** розділі дисертаційного дослідження удосконалено загальний метод розпізнавання об'єктів за рахунок інтеграції адаптивного механізму обмеження області класифікації на основі теплової карти та подальшого застосування еволюційного підходу до оптимізації параметрів у багатокритеріальному середовищі. Запропоновані удосконалення були використані для алгоритму розпізнавання об'єктів DeltaTrack з використанням еволюційного підходу. Еволюційний підхід базується на модифікованому алгоритмі RVEA з динамічним оновленням довідкових векторів, адаптивною селекцією та мутацією, орієнтованою на Парето-фронт. Проведено експериментальну оцінку ефективності обох етапів удосконалення, підтверджено зростання показників точності розпізнавання, зменшення часу обробки кадру та

стабільність роботи в умовах зосереджених завад та обмежених обчислювальних ресурсів.

У висновках до дисертаційного дослідження роботи узагальнено основні результати, які вирішують актуальне науково-технічне та розв'язують часткові задачі, поставлені в роботі.

Результати, отримані в результаті виконання дисертаційної роботи, відображено у вигляді наступних пунктів **наукової новизни**:

1. Вперше запропоновано метод виявлення та класифікації об'єктів у відеопотоці (DeltaTrack), який поєднує операції над образами та трекінг, що забезпечує підвищення швидкодії та зниження обчислювального навантаження на нейромережу в умовах роботи в реальному часі.

2. Вперше запропоновано формалізовану імітаційну модель реалізації методу DeltaTrack у вигляді послідовності математичних операторів, яка описує повний цикл обробки відеопотоку з урахуванням параметрів сцени та ресурсних обмежень, що забезпечує покращення аналітичної оцінки ефективності пошуку та розпізнавання образів у відеопотоці та можливість подальшої адаптації моделі при заданій точності пошуку та розпізнавання образів у відеопотоці.

3. Удосконалено загальний метод розпізнавання об'єктів на відео шляхом застосування адаптації до відеоданих та формування теплової карти регіонів фільтрації неінформативних ділянок, що дозволяє підвищити точність та швидкодію пошуку і розпізнавання образів у відеопотоці.

Практичні результати, отримані в дисертації, полягають у розробці та верифікації методів, моделей і алгоритмів, які підвищують ефективність та стійкість виявлення і класифікації нелінійно змінюваних образів у відеопотоці в умовах дії зосереджених завад, а саме:

– на основі аналізу класичних та згорткових методів виявлення і класифікації об'єктів у відеопотоці обґрунтовано доцільність розробки гібридних методів, які поєднують швидкість традиційних підходів з точністю нейромережових, що забезпечує стійкість до завад і ефективну роботу в реальному часі;

– розроблено алгоритм і програмну реалізацію гібридного методу DeltaTrack, який поєднує переваги міжкадрової різниці (FD) та згорткової класифікації, що дозволило досягти середнього часу обробки одного кадру 5,4 мс, що на 41% швидше, ніж у класичного методу MoG (9,2 мс), і на 8% швидше, ніж у згорткового SSD (5,8 мс) в режимі реального часу;

– удосконалено алгоритм виявлення та класифікації об'єктів за рахунок впровадження гібридної архітектури (FD, морфологічні операції, MobileNetV1, трекінг MOSSE), що дозволило зменшити споживання відеопам'яті до 390 МБ (на 57% менше, ніж у SSD) та підтвердило придатність методу до роботи в реальному часі на пристроях з обмеженими ресурсами;

– удосконалено алгоритм обчислювального ланцюга формалізованої імітаційної моделі DeltaTrack за рахунок включення параметризованих етапів з адаптацією до змін середовища (освітлення, шум, кількість об'єктів), в результаті чого досягнуто стабільну роботу пристроїв при FPS >14 навіть у умовах гамма-шуму ($k=5$) та динамічної сцени з $K=30$;

– реалізовано механізм адаптації методу DeltaTrack до динаміки сцени шляхом автоматичного формування теплової карти активних регіонів, що дозволило зменшити кількість класифікаційних операцій і, як наслідок, скоротити середній час обробки одного кадру на 17% – з 5,3 мс до 4,4 мс при збереженні стабільної роботи в режимі реального часу;

– удосконалено алгоритм розпізнавання об'єктів у відеопотоці за рахунок інтеграції адаптивного фільтрування регіонів і оптимізації ресурсоспоживання, що забезпечило підвищення точності класифікації з 71,4% до 72,2%, зменшення використання GPU на 10,9% і відеопам'яті на 8 МБ без статистично значущого зростання навантаження на CPU та RAM;

– проведено оптимізацію методу DeltaTrack за допомогою модифікованого алгоритму RVEA, що адаптує параметри до змін середовища і динаміки кадрів та забезпечує прискорення обробки кадру на 2,3 %, підвищення точності розпізнавання на 1,1 % і зменшення використання GPU на 2,0 %, при цьому статистична значущість отриманих результатів підтверджена t-критерієм Стюдента.

Всі результати, отримані в дисертаційному дослідженні, мають науково-практичну цінність у сфері інтелектуальної обробки відеоданих, зокрема для підвищення ефективності виявлення та класифікації об'єктів у реальному часі в умовах завад та обмежених ресурсів.

Ключові слова: нейронні мережі, інтелектуальні комп'ютерні системи, телекомунікаційні системи, самонавчання, відеопотік, людино-машинний інтерфейс, керування та розпізнавання рухомих об'єктів, декомпозиція, аномалії, реальний час, адаптивність, моделювання, прийняття рішень, обробка даних, комп'ютерний зір, еволюційний підхід, оптимізація, класифікація зображень, YOLOv5, обмежені ресурси, IoT, завадостійкість, енергоефективність.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

Наукові праці, в яких опубліковано основні наукові результати

1. Sadovnykov B.I., Lysechko V.P., Komar O.M., Zhuchenko O.S. A research of the latest approaches to visual image recognition and classification. 2024, *National University «Zaporizhzhia Polytechnic». Radio Electronics, Computer Science, Control*, 1(68), P. 140-147 DOI 10.15588/1607-3274-2024-1-13 (**Web Of Science** – 2024).
2. Sadovnykov, B.I., Zhuchenko O.S. (2025) Method for Object Detection and Recognition in Video Streams Using Interframe Delta Computation// *National University «Yuri Kondratyuk Poltava Polytechnic». Control, Navigation and Communication Systems*, Vol. 2, № 80 (2025), P.P. 249-254, <https://doi.org/10.26906/SUNZ.2025.2.249>.
3. Sadovnykov B.I., Zhuchenko O.S. Mathematical model for object detection and recognition in video streams using inter-frame difference analysis. National Aviation University. Science-intensive Technologies. Series: «Electronics, telecommunications and radio engineering», Kyiv, 2025. Vol. 66, № 2, PP. 181-189, <https://doi.org/10.18372/2310-5461.66.20281>.
4. Sadovnykov, B., & Lysechko, V. (2025) Adaptive behaviour tuning of a neural network-based method for moving object recognition in video streams// Computer-integrated technologies: education, science, production/ Lutsk National Technical University. – Lutsk. – 2025. (№60), P.P. 53-62, <https://doi.org/10.36910/6775-2524-0560-2025-60-05>.

Опубліковані праці апробаційного характеру:

5. Sadovnykov B. I., Lysechko et al. Experimental study of optimized face recognition algorithms for resource – constrained//Academic journal: Mechanics Transport Communications. – 2023. – Vol. №21, Issue 1, article №2343. – P. 89-95
6. Садівников Б.І., Мазурова О.О. Дослідження методів оптимізації алгоритмів розпізнавання обличчя для роботи у системах з обмеженими

ресурсами //Наукові досягнення та відкриття сучасної молоді: матеріали міжнародної науково-практичної конференції, 28 квітня 2021 року. – Покровськ: ДВНЗ «ДонНТУ», 2021, с. 88-90.

7. Sadovnykov B., Zhuchenko O., Perets K. Overview of state-of-the art image object detection and classification approaches// Інформаційно-керуючі системи на залізничному транспорті: тези доповідей та виступів учасників 36-ї Міжнародної науково-практичної конференції «Інформаційно-керуючі системи на залізничному транспорті». – 2023. – № 3 (додаток). – С. 9-10.

8. Sadovnykov, B.I. Development of an algorithm for detecting moving objects in images from a real-time video stream/ B.I Sadovnykov, O.S. Zhuchenko// Інформаційно-керуючі системи на залізничному транспорті: тези доповідей та виступів учасників 37-ї Міжнародної науково-практичної конференції «Інформаційно-керуючі системи на залізничному транспорті» (Харків, 10-11 жовтня, 2024 р.). – 2024. – № 3 (додаток). – С. 80-81.

9. Sadovnykov B.I., Lysechko V.P. Increasing the speed of recognizing and classifying visual images in video by means of image transformations// Перспективи розвитку озброєння та військової техніки сухопутних військ. Збірник тез доповідей Міжнародної науково-технічної конференції (Львів 15-16 травня 2024). Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, Issue 30 P. 307-308.

10. Sadovnykov B.I., Zhuchenko O.S. Ways to solve the main disadvantages of the method of searching and classifying objects in a video stream using the difference between frames// Тези доповіді за матеріалами науково-практичної конференції «Сектор безпеки і оборони України на захисті національних інтересів: актуальні проблеми та завдання в умовах воєнного стану». Національна академія Державної прикордонної служби України імені Богдана Хмельницького, 22 листопада 2024, С. 699-702.

11. Sadovnykov B.I., Lysechko V.P. Modern approaches to visual object recognition and classification// XXI Міжнародна наукова конференція «Новітні технології для захисту повітряного простору». Харківський національний

університет повітряних сил ім. І. Кожедуба. Харків: 9 – 10 квітня 2025 року. С. 332-333.

12. Садовников Б.І., Жученко О.С., Шевченко О.О. Візуальне розпізнавання об'єктів у режимі реального часу на основі дельт-фільтрації// Перспективи розвитку озброєння та військової техніки сухопутних військ. Збірник тез доповідей Міжнародної науково-технічної конференції (Львів 14-15 травня 2025). – Національна академія сухопутних військ імені гетьмана Петра Сагайдачного. – Львів: НАСВ, 2025, С.89.

ABSTRACT

Sadovnykov B.I. Methods for Classifying Nonlinearly Varying Patterns in Video Streams under Complex Interference Conditions. Qualifying scientific work on manuscript rights.

Dissertation for obtaining a scientific doctor of philosophy (PhD) in specialty 172 – Telecommunications and Radio Engineering – Ukrainian State University of Railway Transport, Ukraine, Kharkiv, 2025.

The dissertation addresses a relevant scientific and technical problem of improving the reliability and efficiency of detection and classification of nonlinearly varying patterns in video streams under the influence of concentrated interference, through the development of methods, models, and algorithms for processing, vectorization, neural classification, and tracking, capable of ensuring stable real-time operation and scalability to changing scene parameters.

The object of research is the process of detecting and classifying patterns in video streams under complex interference conditions with limited computational resources.

The subject of research is methods and models for processing, detection, classification, and tracking of patterns in video streams, aimed at improving the reliability and efficiency of search and recognition in adverse environments.

The purpose of the dissertation is to enhance the accuracy, robustness, and computational efficiency of detecting and classifying nonlinearly varying patterns in video streams under concentrated interference, through the development and implementation of methods and models based on neural networks, applicable on resource-constrained devices in real-time.

The introduction substantiates the relevance of improving the reliability and efficiency of object detection and classification in video streams under concentrated interference and limited computational capacity. It is noted that standard deep learning algorithms achieve high accuracy only under stable backgrounds and powerful hardware, which makes them unsuitable under concentrated interference. The scientific

novelty of the work lies in the development of methods, mathematical models, and algorithms for detection, classification, and tracking of nonlinearly varying patterns using a hybrid approach that combines classical inter-frame processing with convolutional neural networks. The practical significance of the results is demonstrated in terms of increased noise immunity, reduced processing delays, and lower energy consumption when deployed on resource-limited devices. The introduction also provides information about the candidate's personal contribution and a list of publications with research results.

Chapter 1 analyzes modern approaches to detection, classification, and tracking of objects in video streams under concentrated interference and resource constraints. Considered are classical motion detection methods (background subtraction, optical flow, inter-frame difference), deep convolutional architectures (YOLO, SSD, Faster R-CNN), and hybrid models that combine their advantages. Methods are classified by accuracy, performance, robustness to interference, and real-time applicability. The analysis justifies the need for a new method that combines the speed of classical approaches with the accuracy of deep models, adapted to noisy and unstable backgrounds.

Chapter 2 presents a comparative analysis of existing detection and classification methods, forming the basis of the proposed hybrid method DeltaTrack. The approach combines three-frame inter-frame differencing (deltas) with MobileNetV1 for classification of extracted regions. Additional modules include morphological processing, dual-thresholding, and MOSSE-based tracking. DeltaTrack is designed for real-time operation on resource-limited devices with minimal GPU and video memory usage. The chapter substantiates the base approach, develops a step-by-step algorithm, presents the mathematical model and structural diagram, and reports experimental comparisons with SSD (deep CNN) and MoG (classical). Results confirm DeltaTrack's effectiveness in terms of speed, accuracy, and resource efficiency, proving its applicability in scenarios with background variability, interference, and constraints typical of mobile and embedded video analytics systems.

Chapter 3 presents a formalized mathematical model of DeltaTrack, describing the full video processing pipeline with parameters for scene dynamics, background changes, and resource limitations. The model is structured as a sequence of mathematical operators, enabling analytical evaluation, adaptation to environment changes, and implementation on low-power devices. Profiling results are supported with experimental simulations, evaluating time costs, detection accuracy, and stability. It is experimentally confirmed that the formalized DeltaTrack model ensures stable real-time operation without requiring large GPU resources. The computational load is concentrated in morphological processing and classification, while key parameters influence scalability and performance.

Chapter 4 improves general method of object recognition by integrating an adaptive heatmap-based region limitation mechanism and applying an evolutionary approach for multi-objective parameter optimization. The proposed improvements have been used for the DeltaTrack object recognition algorithm. The evolutionary approach is based on a modified RVEA with dynamic reference vector updates, adaptive selection, and Pareto-oriented mutation. Experimental evaluations confirm improved recognition accuracy, reduced frame processing time, and robust real-time performance under concentrated interference and resource limitations.

The conclusions summarize the main results, which address the relevant scientific and technical challenge and resolve specific tasks set out in the dissertation.

The results obtained in the dissertation are reflected in the following points of **scientific novelty**:

1. For the first time, a method for object detection and classification in video streams (DeltaTrack) has been proposed, which combines image operations with tracking, ensuring increased processing speed and reduced computational load on the neural network, while working in the real-time.

2. For the first time, a formalized simulation model of the DeltaTrack method has been proposed in the form of a sequence of mathematical operators describing the full cycle of video stream processing, taking into account scene parameters and resource constraints. This enables improved analytical evaluation of detection and recognition

efficiency, as well as the possibility of further model adaptation with a given recognition accuracy.

3. The general method of object recognition in video has been improved by applying adaptation to video data, by forming a heat map of regions of filtering uninformative areas, which allows to increase the accuracy and speed of image search and recognition in the video stream.

The practical results obtained in the dissertation include the development and verification of methods, models, and algorithms that improve the efficiency and robustness of detecting and classifying nonlinearly varying patterns in video streams under concentrated interference, namely:

- based on the analysis of classical and convolutional methods of detection and classification in video streams, the feasibility of developing hybrid methods has been substantiated, combining the speed of traditional approaches with the accuracy of neural networks, ensuring noise resistance and effective real-time performance;

- an algorithm and software implementation of the hybrid DeltaTrack method have been developed, combining the advantages of frame differencing (FD) and convolutional classification, which achieved an average frame processing time of 5,4 ms – 41% faster than the classical MoG method (9, 2 ms) and 8% faster than convolutional SSD (5,8 ms) in real-time;

- the detection and classification algorithm has been improved by implementing a hybrid architecture (FD, morphological operations, MobileNetV1, MOSSE tracking), which reduced video memory consumption to 390 MB (57% less than SSD) and confirmed real-time applicability on resource-limited devices;

- the computational pipeline of the formalized DeltaTrack model has been enhanced with parameterized stages adapted to environmental changes (lighting, noise, object count), ensuring stable operation with FPS >14 even under gamma noise ($k=5$) and dynamic scenes with $K=30$;

- a mechanism for adapting DeltaTrack to scene dynamics has been implemented through automatic heatmap generation of active regions, reducing the number of

classification operations and lowering average frame processing time by 17% – from 5,3 ms to 4,4 ms – while maintaining stable real-time performance;

- the object recognition algorithm has been refined by integrating adaptive region filtering and resource optimization, which increased classification accuracy from 71,4% to 72,2%, reduced GPU usage by 10,9% and video memory by 8 MB, without statistically significant growth in CPU and RAM load;

- the optimization of the DeltaTrack method was carried out using a modified RVEA algorithm, which adapts processing parameters to environmental changes and frame dynamics, ensuring a 2,3% reduction in frame processing time, a 1,1% increase in recognition accuracy, and a 2,0% decrease in GPU usage, with the statistical significance of the obtained results confirmed by Student's t-test.

All the results obtained in the dissertation have scientific and practical value in the field of intelligent video data processing, particularly for improving the efficiency of object detection and classification in real-time under interference and limited resources.

Keywords: neural networks, intelligent computer systems, telecommunication systems, self-learning, video stream, human–machine interface, control and recognition of moving objects, decomposition, anomalies, real-time, adaptability, modeling, decision-making, data processing, computer vision, evolutionary approach, optimization, image classification, YOLOv5, resource-constrained environments, IoT, noise immunity, energy efficiency.

LIST OF PUBLICATIONS OF THE ACQUIRER

Scientific works in which the main scientific results were published:

1. Sadovnykov B.I., Lysechko V.P., Komar O.M., Zhuchenko O.S. A research of the latest approaches to visual image recognition and classification. 2024, *National University «Zaporizhzhia Polytechnic». Radio Electronics, Computer Science, Control*, 1(68), P. 140-147 DOI 10.15588/1607-3274-2024-1-13 (**Web Of Science** – 2024).

2. Sadovnykov, B.I., Zhuchenko O.S. (2025) Method for Object Detection and Recognition in Video Streams Using Interframe Delta Computation// *National University «Yuri Kondratyuk Poltava Polytechnic». Control, Navigation and Communication Systems*, Vol. 2, № 80 (2025), P.P. 249-254, <https://doi.org/10.26906/SUNZ.2025.2.249>.

3. Sadovnykov B.I., Zhuchenko O.S. Mathematical model for object detection and recognition in video streams using inter-frame difference analysis. National Aviation University. Science-intensive Technologies. Series: «Electronics, telecommunications and radio engineering», Kyiv, 2025. Vol. 66, № 2, PP. 181-189, <https://doi.org/10.18372/2310-5461.66.20281>.

4. Sadovnykov, B., & Lysechko, V. (2025) Adaptive behaviour tuning of a neural network-based method for moving object recognition in video streams// *Computer-integrated technologies: education, science, production/ Lutsk National Technical University*. – Lutsk. – 2025. (№60), P.P. 53-62, <https://doi.org/10.36910/6775-2524-0560-2025-60-05>.

Published works of an approbation nature:

5. Sadovnykov B.I., Lysechko V.P. Experimental study of optimized face recognition algorithms for resource-constrained systems // *Academic journal: Mechanics Transport Communications*, 2023, Vol. 21, Issue 1, Article No. 2343, pp. 89–95.

6. Sadovnykov B.I., Mazurova O.O. Investigation of optimization methods for face recognition algorithms in systems with limited resources//Scientific Achievements and Discoveries of Modern Youth: Proceedings of the International Scientific and Practical Conference, April 28, 2021. – Pokrovsk: SEI “DonNTU”, 2021, pp. 88–90.

7. Sadovnykov B.I., Zhuchenko O.S., Perets K.G. Overview of state-of-the-art image object detection and classification approaches//Information and Control Systems in Railway Transport: Proceedings of the 36th International Scientific and Practical Conference, 2023, No. 3 (supplement), pp. 9–10.

8. Sadovnykov B.I., Zhuchenko O.S. Development of an algorithm for detecting moving objects in real-time video streams // Information and Control Systems in Railway Transport: Proceedings of the 37th International Scientific and Practical Conference (Kharkiv, October 10–11, 2024). – 2024, No. 3 (supplement), pp. 80–81.

9. Sadovnykov B.I., Lysechko V.P. Increasing the speed of recognizing and classifying visual images in video by means of image transformations // Prospects for the Development of Armaments and Military Equipment of the Land Forces: Proceedings of the International Scientific and Technical Conference (Lviv, May 15–16, 2024). – Hetman Petro Sahaidachnyi National Army Academy, Issue 30, pp. 307–308.

10. Sadovnykov B.I., Zhuchenko O.S. Solving the key limitations of the method for detecting and classifying objects in a video stream using frame differencing // Conference Proceedings: «Security and Defense Sector of Ukraine in Protection of National Interests: Current Problems and Tasks under Martial Law», National Academy of the State Border Guard Service of Ukraine named after B. Khmelnytskyi, November 22, 2024, pp. 699–702.

11. Sadovnykov B.I., Lysechko V.P. Modern approaches to visual object recognition and classification // 21st International Scientific Conference «Advanced Technologies for the Protection of Airspace», Kharkiv National University of the Air Force named after I. Kozhedub, April 9–10, 2025, pp. 332–333.

12. Sadovnykov B.I., Zhuchenko O.S., Shevchenko O.O. Real-time visual object recognition based on delta filtering // Prospects for the Development of Armaments and Military Equipment of the Land Forces: Proceedings of the International Scientific and Technical Conference (Lviv, May 14–15, 2025). – Hetman Petro Sahaidachnyi National Army Academy, Issue 31, P.89

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	21
ВСТУП	22
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ	30
1.1 Аналіз сучасних задач та вимог до методів обробки відео в умовах змінного фону	31
1.2 Класичні методи детекції руху: порівняння та обмеження	35
1.3 Глибинні згорткові методи обробки відео та їх обмеження	39
1.4 Обґрунтування доцільності розробки гібридного методу виявлення та класифікації об'єктів у відеопотоці	49
Висновки за розділом 1	51
РОЗДІЛ 2 РОЗРОБКА МЕТОДУ DELTATRACK ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ	53
2.1 Аналіз і обґрунтування вибору базового підходу до виявлення руху у відеопотоці	54
2.2 Архітектура та алгоритм реалізації методу DeltaTrack	60
2.3 Експериментальна оцінка ефективності методу DeltaTrack у порівнянні з SSD та MoG	72
Висновки за розділом 2	80
РОЗДІЛ 3. ФОРМАЛІЗОВАНА ІМІТАЦІЙНА МОДЕЛЬ МЕТОДУ ДЕТЕКЦІЇ ТА ТРЕКІНГУ DELTATRACK В УМОВАХ ДИНАМІЧНОГО ФОНУ	81
3.1 Структура та формалізація імітаційної моделі методу DeltaTrack	82
3.2 Аналітична оцінка обчислювальної складності параметрів формалізованої моделі	88
3.3 Адаптація імітаційної моделі методу DeltaTrack до змінних умов середовища	95

3.4 Експериментальна верифікація продуктивності та ефективності імітаційної моделі DeltaTrack	99
Висновки за розділом 3	103
РОЗДІЛ 4. УДОСКОНАЛЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ DELTATRACK НА ОСНОВІ ЕВОЛЮЦІЙНОГО ПІДХОДУ ДО АДАПТИВНОГО НАЛАШТУВАННЯ ПАРАМЕТРІВ	105
4.1 Інтеграція механізму адаптації у метод розпізнавання об'єктів DeltaTrack на відеопотоці	107
4.2 Експериментальна оцінка продуктивності удосконаленого методу DeltaTrack з механізмом адаптації до сцени	113
4.3 Розробка модифікації еволюційного алгоритму RVEA для задачі розпізнавання образів в умовах зосереджених завад	119
4.4 Порівняльна оцінка ефективності еволюційного підходу в задачі розпізнавання образів в умовах зосереджених завад	124
Висновки за розділом 4	128
ЗАГАЛЬНІ ВИСНОВКИ	130
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	132
ДОДАТОК А. Список публікацій здобувача за темою дослідження	148
ДОДАТОК Б. Акти впровадження результатів наукових досліджень дисертаційної роботи	151
ДОДАТОК В. Фрагмент програмної реалізації для експериментальної перевірки метрик методу	154
ДОДАТОК Г. Фрагмент програмної реалізації методу формування дельта регіонів з морфологічним очищенням	157

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- МКР – міжкадрова різниця
- МО – морфологічні операції
- ПО – порогове обчислення
- ТО – теплові області (теплова карта)
- ТР – трекінг об’єктів
- ОК – обчислювальне навантаження на кадр
- CPU – Central Processing Unit – центральний процесор
- GPU – Graphics Processing Unit – графічний процесор
- RAM – Random Access Memory – оперативна пам’ять
- VRAM – Video Random Access Memory – відеопам’ять
- IoU – Intersection over Union – метрика/показник перетину областей
- FD /3-FD – Frame Difference / Three-Frame Difference – міжкадрова різниця / трьохкадрова різниця
- MoG – Mixture of Gaussians – метод змішування гауссіанів
- SOBS – Self-Organizing Background Subtraction – самоорганізоване віднімання фону
- SSD – Single Shot Detector – згорткова нейронна мережа для розпізнавання об’єктів
- YOLO – You Only Look Once – згорткова нейронна мережа для швидкої детекції та класифікації об’єктів
- R-CNN / Faster R-CNN – Region-based Convolutional Neural Network – регіональні згорткові нейромережі для пошуку та класифікації об’єктів
- RVEA – Reference Vector Guided Evolutionary Algorithm – еволюційний алгоритм з довідковими векторами
- MOSSE – Minimum Output Sum of Squared Error – алгоритм трекінгу з мінімізацією середньоквадратичної похибки

ВСТУП

Актуальність теми дослідження. З розвитком інтелектуальних телекомунікаційних систем, зокрема у сфері відеоаналітики, систем безпеки, моніторингу критичної інфраструктури, транспортних комунікацій і мобільного зв'язку, зростає потреба в ефективному аналізі та обробці відеопотоків у реальному часі. Одним з основних викликів є розпізнавання та класифікація об'єктів у відеопотоці в умовах змінного середовища та дії зосереджених завад, які характерні для відкритих телекомунікаційних каналів, зокрема в розподіленій інфраструктурі або під час використання безпроводової передачі даних. Такі умови супроводжуються нестабільністю фону, динамікою сцени, фрагментацією зображення та обмеженнями пропускної здатності і ресурсів обчислення, що потребує застосування адаптивних, ресурсоощадних та стійких до завад методів обробки відеоінформації.

У результаті виникає проблема вибору між високоточними, але ресурсоємними згортковими нейронними мережами (SSD, YOLO, Faster R-CNN), які не завжди сумісні з обмеженнями телекомунікаційних пристроїв, та класичними методами аналізу руху (міжкадрові різниці, субстракція фону), що є легшими в реалізації, але менш стійкими до завад та зміни сцени.

Відтак, перспективним напрямом є створення гібридного підходу, що поєднує інструменти цифрової обробки сигналів, алгоритми імітаційно-математичного моделювання та легковагові нейромережеві архітектури з урахуванням специфіки телекомунікаційного середовища: обмеженої пропускної здатності, втрат пакетів, роботи в реальному часі та впливу завад. Такі рішення мають бути формалізовані, адаптивні та придатні до впровадження у вузли відеоспостереження, термінали мобільних мереж, платформи залізничного транспорту та інші компоненти телекомунікаційних систем.

З огляду на зазначені виклики, постає потреба у вирішенні актуального науково-технічного завдання дисертаційного дослідження, яке полягає у підвищенні достовірності та ефективності виявлення та класифікації нелінійно

змінюваних образів у відеопотоці в умовах впливу зосереджених завад, за допомогою розробки методів, математичних моделей та алгоритмів обробки, векторизації, нейронної класифікації та трекінгу, які забезпечують адаптивність до змін сцени, зменшення ресурсоспоживання та стабільну роботу.

Зв'язок роботи з науковими програмами, планами, темами.

Тема дисертаційного дослідження повністю відповідає характеристиці предметній області ОНП «Телекомунікації та радіотехніка» третього (освітньо-наукового) рівня, оскільки заснована на теоретичних принципах системного аналізу, методах обробки інформації та розпізнавання образів; використовує інструменти цифрових технологій, легковагових нейронних архітектур і засобів інтелектуальної обробки даних, що є актуальними для сучасних телекомунікаційних та радіотехнічних застосувань, включно з задачами відеоспостереження, моніторингу та автоматизованого управління.

Дисертаційна робота виконана на кафедрі «Транспортний зв'язок» Українського державного університету залізничного транспорту. Результати роботи впроваджено в навчальний процес Українського державного університету залізничного транспорту при викладанні дисциплін: «Нейромережеві технології в електронних комунікаціях», «Мультимедійні системи», «Інноваційні технології в інфокомунікаціях» (Додаток В). Науково-практичні результати, отримані в ході дисертаційного дослідження використовуються у службовій діяльності військової частини А7223 (Додаток Б).

Об'єктом дослідження є процес виявлення і класифікації образів у відеопотоці в умовах дії завад та обмежених обчислювальних ресурсів.

Предметом дослідження є методи, моделі та алгоритми обробки, виявлення змін, класифікації та трекінгу об'єктів у відеопотоці, спрямовані на підвищення достовірності та ефективності ідентифікації та аналізу зображень у реальному часі.

Метою дисертаційної роботи є підвищення достовірності, стійкості та ефективності виявлення і класифікації нелінійно змінюваних образів у відеопотоці в умовах впливу зосереджених завад, шляхом розробки методів, моделей і

алгоритмів, придатних до реалізації на ресурсно обмежених пристроях у режимі реального часу.

Для реалізації мети дисертаційної роботи було поставлено такі часткові наукові **задачі дослідження**.

1. Провести порівняльний аналіз класичних та згорткових методів виявлення і класифікації об'єктів у відеопотоці з урахуванням точності, ресурсоемності та стійкості до завад, та обґрунтувати доцільність розробки гібридного методу, адаптованого до роботи в умовах змінного фону та обмежених обчислювальних ресурсів.

2. Розробити метод виявлення і класифікації об'єктів DeltaTrack, що поєднує міжкадрову різницю, морфологічну фільтрацію, легковаговий нейронний класифікатор та адаптивний трекінг, і реалізувати його у вигляді ресурсоощадного алгоритму реалізації, придатного до використання в реальному часі.

3. Розробити алгоритм реалізації методу DeltaTrack на основі структурного поєднання класичних і нейронних підходів, з детальним описом послідовності операцій, забезпеченням обробки в реальному часі та оптимізацією споживання ресурсів для використання на пристроях з обмеженими можливостями.

4. Побудувати формалізовану імітаційну модель методу DeltaTrack у вигляді послідовності математичних операторів, що описують повний цикл обробки відеопотоку з урахуванням динаміки сцени та ресурсних обмежень, і провести аналітичну оцінку обчислювальної складності її основних етапів.

5. Розробити механізм адаптивного налаштування методу DeltaTrack до особливостей динамічної сцени шляхом автоматичної побудови теплової карти активних регіонів кадру, що дозволяє селективно застосовувати класифікатор та сприяє зменшенню часу обробки без втрати точності.

6. Удосконалити метод розпізнавання об'єктів у відеопотоці за рахунок впровадження адаптивного фільтрування регіонів та оптимізації ресурсного навантаження, з метою забезпечення підвищеної точності класифікації та зниження споживання GPU і VRAM при збереженні стабільної роботи на пристроях з обмеженими обчислювальними можливостями.

7. Розробити модифікацію еволюційного алгоритму RVEA для адаптивного налаштування параметрів методу розпізнавання у відеопотоці, що забезпечує зменшення середнього часу обробки кадру, підвищення точності та оптимізацію використання GPU без втрати стабільності в умовах зосереджених завад.

Методи дослідження. Для розв'язання задач, поставлених у дисертації було застосовано: методи математичного моделювання для побудови формалізованої моделі методу DeltaTrack, що описує етапи обробки відеопотоку: від кадрування та фільтрації до класифікації і трекінгу об'єктів; методи аналізу алгоритмічної складності для визначення обчислювального навантаження етапів обробки відеопотоку з урахуванням змінних параметрів сцени, глибини класифікатора та розміру зображення; методи оптимізації для вибору параметрів морфологічної обробки та структури класифікатора в умовах змінного фону та шумів; методи експериментального моделювання у програмному середовищі Python з використанням бібліотек OpenCV, NumPy та TensorFlow для практичної реалізації та тестування методу DeltaTrack у відеопотоці реального часу; методи статистичного аналізу (вибіркова статистика, середні значення, стандартне відхилення, регресійний аналіз) для верифікації результатів і оцінки стабільності обробки, точності класифікації та впливу шумів.

Наукова новизна отриманих результатів заснована на запропонованих методах та формалізованій імітаційній моделі, які забезпечують підвищення ефективності обробки відеопотоків у реальному часі при обмежених ресурсах. На основі цього було отримано такі наукові результати:

Результати, отримані в результаті виконання дисертаційної роботи, відображено у вигляді наступних пунктів **наукової новизни**:

1. **Вперше** запропоновано метод виявлення та класифікації об'єктів у відеопотоці (DeltaTrack), який поєднує операції над образами та трекінг, що забезпечує підвищення швидкодії та зниження обчислювального навантаження на нейромережу в умовах роботи в реальному часі.

2. Вперше запропоновано формалізовану імітаційну модель реалізації методу DeltaTrack у вигляді послідовності математичних операторів, яка описує повний цикл обробки відеопотоку з урахуванням параметрів сцени та ресурсних обмежень, що забезпечує покращення аналітичної оцінки ефективності пошуку та розпізнавання образів у відеопотоці та можливість подальшої адаптації моделі при заданій точності пошуку та розпізнавання образів у відеопотоці.

3. Удосконалено загальний метод розпізнавання об'єктів на відео шляхом застосування адаптації до відеоданих та формування теплової карти регіонів фільтрації неінформативних ділянок, що дозволяє підвищити точність та швидкодію пошуку і розпізнавання образів у відеопотоці.

Практичні результати, отримані в дисертації, полягають у розробці та верифікації методів, моделей і алгоритмів, які підвищують ефективність та стійкість виявлення і класифікації нелінійно змінюваних образів у відеопотоці в умовах дії зосереджених завад, а саме:

- на основі аналізу класичних та згорткових методів виявлення і класифікації об'єктів у відеопотоці обґрунтовано доцільність розробки гібридних методів, які поєднують швидкість традиційних підходів з точністю нейромережевих, що забезпечує стійкість до завад і ефективну роботу в реальному часі;

- розроблено алгоритм і програмну реалізацію гібридного методу DeltaTrack, який поєднує переваги міжкадрової різниці (FD) та згорткової класифікації, що дозволило досягти середнього часу обробки одного кадру 5,4 мс, що на 41% швидше, ніж у класичного методу MoG (9,2 мс), і на 8% швидше, ніж у згорткового SSD (5,8 мс) в режимі реального часу;

- удосконалено алгоритм виявлення та класифікації об'єктів за рахунок впровадження гібридної архітектури (FD, морфологічні операції, MobileNetV1, трекінг MOSSE), що дозволило зменшити споживання відеопам'яті до 390 МБ (на 57% менше, ніж у SSD) та підтвердило придатність методу до роботи в реальному часі на пристроях з обмеженими ресурсами;

– удосконалено алгоритм обчислювального ланцюга формалізованої імітаційної моделі DeltaTrack за рахунок включення параметризованих етапів з адаптацією до змін середовища (освітлення, шум, кількість об'єктів), в результаті чого досягнуто стабільну роботу пристроїв при FPS >14 навіть у умовах гамма-шуму ($k=5$) та динамічної сцени з $K=30$;

– реалізовано механізм адаптації методу DeltaTrack до динаміки сцени шляхом автоматичного формування теплової карти активних регіонів, що дозволило зменшити кількість класифікаційних операцій і, як наслідок, скоротити середній час обробки одного кадру на 17% – з 5,3 мс до 4,4 мс при збереженні стабільної роботи в режимі реального часу;

– удосконалено алгоритм розпізнавання об'єктів у відеопотоці за рахунок інтеграції адаптивного фільтрування регіонів і оптимізації ресурсоспоживання, що забезпечило підвищення точності класифікації з 71,4% до 72,2%, зменшення використання GPU на 10,9% і відеопам'яті на 8 МБ без статистично значущого зростання навантаження на CPU та RAM;

– проведено оптимізацію методу DeltaTrack за допомогою модифікованого алгоритму RVEA, що адаптує параметри до змін середовища і динаміки кадрів та забезпечує прискорення обробки кадру на 2,3 %, підвищення точності розпізнавання на 1,1 % і зменшення використання GPU на 2,0 %, при цьому статистична значущість отриманих результатів підтверджена t-критерієм Стьюдента.

Науково-практичні результати, отримані в ході дисертаційного дослідження, впроваджено в службову діяльність військової частини А7223, що підтверджено відповідним Актом реалізації, наведеним у додатку Б.

Особистий внесок здобувача полягає у розробці методів, математичних моделей, алгоритмів та програмних рішень, що забезпечують виконання поставлених у дисертаційному дослідженні завдань.

У роботах, опублікованих у співавторстві, здобувачу належать такі наукові результати: [1] – узагальнено сучасні підходи з розпізнавання образів, проведено порівняльний аналіз алгоритмів, обґрунтовано перспективи використання легковагових нейромережових моделей; [2] – розроблено метод виявлення та

класифікації об'єктів на основі міжкадрової дельти, що забезпечує зменшення ресурсоспоживання; [3] – побудовано імітаційну модель методу з формалізацією етапів попередньої обробки, векторизації, класифікації та трекінгу; [4] – запропоновано удосконалення методу DeltaTrack шляхом інтеграції механізму адаптації до вхідних даних; [5] – проведено експериментальне дослідження продуктивності алгоритмів розпізнавання у системах з обмеженими ресурсами; [6] – досліджено методи оптимізації алгоритмів розпізнавання образів та придатність їх застосування на вбудованих платформах; [7] – здійснено огляд сучасних методів виявлення об'єктів, визначено переваги гібридних підходів; [8] – реалізовано алгоритм детекції об'єктів у відеопотоці в реальному часі, проаналізовано швидкодію та стабільність; [9] – обґрунтовано застосування оптимізованих перетворень для підвищення швидкодії розпізнавання в умовах обмежених ресурсів; [10] – запропоновано шляхи подолання основних обмежень методу міжкадрової різниці у складних відеосценах; [11] – систематизовано сучасні підходи до візуального розпізнавання об'єктів; [12] – розроблено алгоритм методу дельта-фільтрації для стабілізації розпізнавання об'єктів у відеопотоці в умовах реального часу.

Апробація результатів. Основні наукові результати та положення дисертаційного дослідження були представлені автором на наступних конференціях, де пройшли апробацію та фахове обговорення:

- Міжнародна науково-практична конференція «Наукові досягнення та відкриття сучасної молоді», Покровськ: ДВНЗ «ДонНТУ», 2021, 28 квітня 2021;
- 36 Міжнародна науково-практична конференція «Інформаційно-керуючі системи на залізничному транспорті». Харків: УкрДУЗТ, 16-17 листопада 2023;
- 37 міжнародна науково-практична конференція «Інформаційно-керуючі системи на залізничному транспорті». Харків: УкрДУЗТ, 10-11 жовтня 2024;
- Міжнародна науково-технічна конференція «Перспективи розвитку озброєння та військової техніки сухопутних військ». Національна академія сухопутних військ імені гетьмана Петра Сагайдачного Львів 15-16 травня 2024;

– Науково-практична конференція «Сектор безпеки і оборони України на захисті національних інтересів: актуальні проблеми та завдання в умовах воєнного стану». Національна академія Державної прикордонної служби України імені Богдана Хмельницького, 22-23 листопада 2024;

– XXI Міжнародна наукова конференція «Новітні технології для захисту повітряного простору». Харківський національний університет повітряних сил ім. І. Кожедуба. Харків: 9 – 10 квітня 2025 року.

– Міжнародна науково-технічна конференція «Перспективи розвитку озброєння та військової техніки Сухопутних військ», 14-15 травня 2025 рік.

Публікації. Результати, отримані в межах даного дисертаційного дослідження було опубліковано автором у 12 наукових працях: 4 статтях у фахових виданнях України, у тому числі, 1 стаття у виданні внесеному до міжнародної наукометричної бази Web of Science та 8 публікаціях за матеріалами міжнародних науково-практичних конференцій.

Структура і обсяг дисертації. Дисертаційна робота складається з анотації двома мовами, вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Робота містить 150 сторінок основного тексту, у тому числі, 43 рисунка, 29 таблиць, 143 найменування у списку використаних джерел, 4 Додатка (А-Г). Загальний обсяг дисертаційної роботи займає 158 сторінок.

РОЗДІЛ 1

СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

Проблема розпізнавання та класифікації образів на фіксованому зображенні або у відеопотоці має широке прикладне значення не лише в галузі комп'ютерного зору, а також у сучасних телекомунікаційних технологіях. Вона актуальна для систем відеомоніторингу, автономного пересування, кіберзахисту, автоматизованого управління. Особливого значення проблема набуває в розподілених телекомунікаційних системах, де ефективність обробки відеопотоків безпосередньо впливає на якість та надійність передавання інформації, особливо в умовах обмежених обчислювальних ресурсів, завад та нестабільного зв'язку.

Аналіз публікацій за темою дослідження дозволив виділити наступні підходи до виявлення, локалізації та класифікації візуальних об'єктів.

1. Глибинні згорткові мережі, зокрема YOLO, SSD, Faster R-CNN, досліджено у роботах [13, 19, 23, 37, 40, 44, 78, 103, 104, 118]. Доведено, що вони забезпечують високу точність, але мають високе ресурсоспоживання. Наприклад, у [13] обґрунтовується вплив порогу NMS на роботу YOLO за низької якості зображень, а в [78] YOLOv8 адаптується для аерофотознімків, що є релевантним у телекомунікаційних задачах з використанням безпілотників.

2. В роботах [50, 58, 77] доведено, що легковагові архітектури, такі як MobileNet-SSD, розраховані на обмежене середовище, тобто на граничні обчислення у вузлах телекомунікаційної мережі. У [58] MobileNet застосовується в роботі з Raspberry Pi, що демонструє його придатність до використання на пристроях з малими ресурсами.

3. Гібридні методи, які поєднують класичні детектори (міжкадрову різницю, фонове моделювання) з нейромережами, описані у [8, 10, 14, 20, 84]. Обґрунтовано, що вони дозволяють зменшити обсяг обчислень без значної втрати

точності. У [8] поєднано міжкадрову різницю з фоном, а в [20] — спектральні зображення з CNN.

4. В роботах [60, 121, 120, 137] розглядається зменшення складності моделей SSD за рахунок використання GhostNet, глибоких роздільних згорток, координаційної уваги, дистиляції знань, що особливо важливо для реалізації в телекомунікаційних вузлах.

5. Класичні методи обробки відео на основі міжкадрової різниці, трекінгу, аналізу фону описано у [1, 2, 5, 6, 10, 21]. Вони демонструють стабільність і низьке навантаження, що є доцільним у попередньому аналізі відеопотоку перед передачею по каналах зв'язку.

Таким чином, аналіз досліджень з обраної теми показує, що в задачах, пов'язаних з телекомунікаціями, потрібні рішення, які враховують ресурсні обмеження, завадостійкість та здатність до роботи в реальному часі. Частина сучасних підходів не враховує ці умови повною мірою. Тому актуальним є розробка гібридного підходу, який поєднує прості методи попередньої обробки з ефективними засобами класифікації, з орієнтацією на використання у телекомунікаційних мережах.

1.1 Аналіз сучасних задач та вимог до методів обробки відео в умовах змінного фону

Обробка відео в телекомунікаційних застосуваннях потребує адаптації до умов, у яких фон може змінюватися через освітлення, погодні фактори, рух об'єктів або компресію даних. У таких середовищах відеоаналіз здійснюється в умовах нестабільної якості переданих даних, затримок і обмежень на ресурси обробки. Змінність фону призводить до хибнопозитивних спрацьовувань детекторів, втрати інформативності кадрів і загального зниження достовірності розпізнавання. Окрім того, обробка часто виконується на граничних вузлах мережі або мобільних пристроях, де обчислювальні та енергетичні ресурси обмежені [1-3, 28, 30, 42, 55].

У результаті, методи відеоаналізу, які застосовуються в телекомунікаційних середовищах, мають відповідати специфічним вимогам, тобто бути стійкими до завад, ефективними при низькій якості сигналу та здатними до роботи в режимі реального часу.

До основних задач, які постають при обробці відео в таких умовах, належать.

1. Виявлення руху та виокремлення динамічних об'єктів [1, 28, 51]

Задача полягає в точному виявленні змін у сцені, пов'язаних з переміщенням об'єктів, а не з фоновими коливаннями (тінями, гілками, дощем тощо). Помилкове трактування таких змін як об'єктів викликає зайве навантаження на канали зв'язку та знижує ефективність подальшої обробки. В телекомунікаційних мережах, де передається умовно важлива інформація, необхідно мінімізувати хибнопозитивні спрацьовування.

Для оцінки точності виявлення об'єктів широко застосовується показник IoU (Intersection over Union), який показує рівень перетину детектованої області з реальною (рис. 1.1). Розраховується за формулою [28, 124]:

$$IoU = \frac{S_{\text{перетину}}}{S_{\text{об'єднання}}} \quad 1.1$$

де $S_{\text{перетину}}$ – площа спільної частини між передбаченим і реальним об'єктом, а $S_{\text{об'єднання}}$ – площа, що охоплює обидва об'єкти.



Рисунок 1.1 – Показник IoU для оцінки точності виділення об'єктів

На рис. 1.1 зображено три рівні відповідності між виявленою та реальною областю: низький ($IoU = 0,35$), середній ($0,74$) та високий ($0,93$). У безпроводових мережах оптимальним вважається значення IoU понад $0,5$ – $0,7$, в залежності від задачі (моніторинг, трекінг або передача об'єктів через завадостійкий канал).

2. Стійка фільтрація завад при передачі відео [28, 42]

Безпроводові та мобільні канали зв'язку схильні до спотворень, втрат пакетів і компресійних артефактів (особливо в форматах H.264/H.265). Метод обробки має не реагувати на завади, що не змінюють реального змісту відео. Це особливо актуально для задач моніторингу, відеоконференцій та дистанційного керування.

Таблиця 1.1 – Формати стиснення відео та типові артефакти*

Формат стиснення	Типові артефакти	Поширеність у мережах
H.264	Блокування, розмиття, затримки між кадрами	Широко використовується в IP-відео
H.265	Втрати якості на швидких сценах, розриви фрагментів	Переважає в сучасних мобільних стрімах
MJPEG	Шум, зниження деталізації при високій компресії	Вебкамери, прості відеопотоки
VP9	Штучні контури, змазування дрібних деталей	Платформи YouTube, WebRTC, стримінг

**складено автором за аналізом характеристик форматів відеокодування [28, 42].*

Як видно з табл. 1.1, найбільш поширені формати H.264 та H.265 схильні до спотворень, що потребують додаткової фільтрації при обробці відео в телекомунікаційних застосуваннях.

3. Зниження обчислювального навантаження на вузлах[55]

У багатьох випадках відео обробляється не на потужних серверах, а на периферійних пристроях (Edge), наприклад, маршрутизаторах, камерах чи

мобільних вузлах. Енергоефективність та низьке використання ресурсів є визначальними факторами для вибору методу в телекомунікаційних мережах.

4. Забезпечення обробки в реальному часі [17]

Для практичного застосування у телеметрії, безпеці чи мобільному зв'язку важливо, щоб загальна затримка (від надходження кадру до прийняття рішення) не перевищувала 100–200 мс. Методи мають бути оптимізовані для швидкої обробки з урахуванням обмежень апаратної платформи та пропускної здатності мережі.

У табл. 1.2 показано вимоги до методів відеообробки, що застосовуються в умовах змінного фону. Ці параметри є базовими при розробці методів, які мають бути інтегровані у граничні вузли телекомунікаційних мереж.

Таблиця 1.2 – Вимоги до методів обробки відео в умовах змінного фону

Характеристика	Орієнтовні вимоги у телекомунікаціях	Значущість при проектуванні
Стійкість до змін освітлення	Алгоритм має адаптуватися до плавних і різких змін	Важлива
Робота в режимі реального часу	$FPS \geq 25$, затримка ≤ 200 мс	Основна вимога
Ресурсне навантаження	≤ 150 MB RAM, $\leq 20\%$ CPU/GPU	Обмеження при впровадженні
Сумісність із форматами кодування	Стабільна робота зі стисненими H.264/H.265 потоками	Бажана
Адаптивність до фонових коливань	Оновлення моделей фону, виявлення дрібних змін	Перевага

**узагальнено автором на основі практичного аналізу.*

У реальних умовах телекомунікаційних систем комбінація вимог до методів відеообробки може істотно відрізнятися залежно від задачі.

У табл. 1.3 наведено типові приклади застосувань, що показують співвідношення основних параметрів: затримки, завадостійкості, точності та ресурсної чутливості.

Таблиця 1.3 – Типові сценарії використання відеоаналізу*

Сценарій використання	Вимоги до затримки	Пріоритет точності	Вплив фону	Прикладний формат
Відеоспостереження	Середні (<300 мс)	Середній	Високий	H.264
Відеоконференція	Високі (<150 мс)	Високий	Середній	H.265
Відеоаналітика в IoT	Низькі (<500 мс)	Низький	Високий	MJPEG, VP9
Дистанційне керування/навігація	Дуже високі (<100 мс)	Високий	Високий	H.264/H.265

*узагальнено автором з урахуванням практики застосувань у телекомунікаційних системах

Як видно з табл. 1.3, методи відеообробки повинні бути адаптовані до специфіки телекомунікаційних систем. У задачах реального часу (наприклад, дистанційного керування) основну роль відіграє мінімальна затримка, тоді як у відеоспостереженні пріоритетною є стійкість до фонових змін і завад, що дозволяє уникати хибних спрацьовувань у складних умовах середовища.

Таким чином, обробка відео в телекомунікаційних системах вимагає врахування змінного фону, обмежених ресурсів і необхідності роботи в реальному часі. Розглянуті задачі, а саме: виявлення руху, фільтрація завад, мінімізація навантаження та затримок – визначають вимоги до сучасних методів відеоаналізу.

1.2 Класичні методи детекції руху: порівняння та обмеження

Класичні методи виявлення та аналізу руху ґрунтуються на порівнянні кадрів, оцінці змін у пікселях, моделюванні фону або векторному представленні змін. Вони не потребують попереднього навчання, мають низьку складність реалізації та добре підходять для використання в телекомунікаційних середовищах з обмеженими ресурсами. Це актуально для периферійних пристроїв, де ускладнене використання глибоких неймереж [1, 25, 28, 42, 135].

У табл. 1.4 наведено характеристику найбільш ефективних класичних методів відеоаналізу, а також їхні переваги та обмеження.

Таблиця 1.4 – Порівняння класичних методів детекції руху*

Метод	Суть	Переваги	Недоліки
FD	Двокадрова різниця	Простота, швидкість	Чутливість до шумів, немає врахування історії
3-FD	Трикадрова різниця	Краще врахування руху, висока швидкодія	Складніша реалізація, залежність від кадрування
MoG	Модель фону як суміш нормальних розподілів	Добре працює зі статичним фоном	Чутлива до завад, повільна адаптація
Eigen-BG	Виявлення фону за допомогою PCA (власні вектори)	Висока точність на стабільних сценах	Високе обчислювальне навантаження
Simp-SOBS	Адаптивний фон у кольоровому просторі	Хороша робота з кольоровими сценами	Висока складність, нестійкість до шуму

*узагальнено автором на основі [1-3,8]

Проте, характеристик недостатньо для вибору методу. Щоб оцінити функціональну придатність до задач відеоаналізу в телекомунікаціях, доцільно порівняти їх за кількісними показниками якості виявлення (Precision - точність, Recall-повнота, F1-міра), а також за показниками навантаження на ресурси (використання CPU та RAM).

Таке порівняння представлено в табл. 1.5, яке оснований на результатах експериментального моделювання з використанням відкритих відеоданих CDNet2014 для категорій із фоновими коливаннями та динамічними сценами. Розрахунок показників Precision, Recall та F1-міри проводився шляхом порівняння детектованих масок руху з еталонними розмітками кадрів. Витрати ресурсів (CPU, RAM) оцінювались на однопоточному виконанні в умовах обмеженого середовища, що емулює граничний вузол (CPU 1.6 GHz, RAM 512 MB).

Таблиця 1.5 – Порівняння методів детекції за показниками

Метод	Точність (Precision)	Повнота (Recall)	F1-міра	CPU Usage (%)	RAM Usage (MB)
FD	0,48	0,32	0,38	6	45
3-FD	0,71	0,55	0,61	9	50
MoG	0,65	0,61	0,63	25	110
Eigen-BG	0,83	0,68	0,75	40	200
Simp-SOBS	0,76	0,59	0,66	33	160

*розраховано автором

Отримані значення показують, що не дивлячись на високі показники точності та F1-міри у методі Eigen-BG, метод потребує значних ресурсів, що робить його складним для впровадження на Edge-пристроях. Simp-SOBS і MoG дають прийнятну якість, проте потребують більше оперативної пам'яті та часу для обчислень. Метод 3-FD, хоча і дещо поступається Eigen-BG за точністю, забезпечує прийнятний баланс між продуктивністю та ефективністю при мінімальному навантаженні на систему (рис. 1.2).

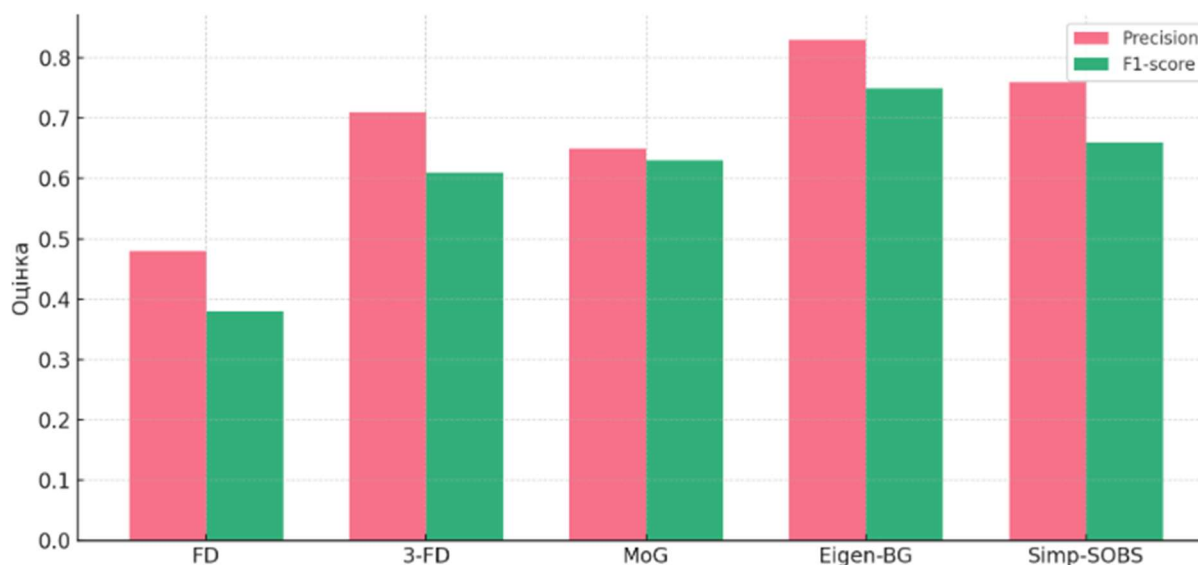


Рисунок 1.2 – Оцінка методів детекції за точністю та F1-мірою

Як видно з рис. 1.2, метод 3-FD досягає хорошого балансу між якістю ($F1=0,61$) та простотою реалізації (низьке навантаження на CPU/RAM). У порівнянні з MoG або Simp-SOBS, він потребує менше ресурсів і показує стабільні

результати на змінному фоні, тоді як Eigen-BG, хоча і показує високу точність, є надто ресурсоємним для телекомунікаційних пристроїв.

На рис. 1.3 представлено порівняння методів за ресурсним навантаженням на процесор та оперативну пам'ять, що дозволяє оцінити придатність до впровадження в умовах обмежених обчислювальних ресурсів.

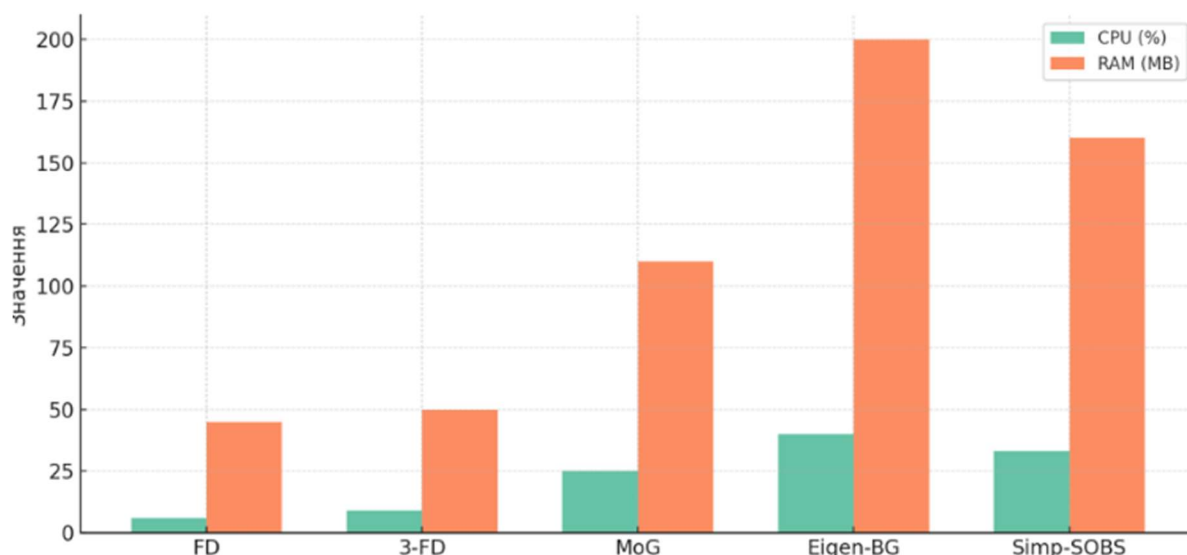


Рисунок 1.3 – Ресурсне навантаження класичних методів детекції руху

Як видно з рис. 1.3, методи Eigen-BG і Simp-SOBS мають високі показники споживання ресурсів, що ускладнює їх застосування. Натомість 3-FD показує нижче навантаження на CPU і RAM при збереженні прийняттого рівня точності виявлення, що підтверджує його практичну придатність для обробки відео в телекомунікаційних системах з обмеженими ресурсами.

Проведене порівняння класичних методів детекції показало, що 3-FD забезпечує оптимальне поєднання точності, стійкості до завад та економного використання ресурсів, що робить метод обґрунтованим вибором для попередньої обробки відео в телекомунікаційних системах. FD та 3-FD мають дуже схожий принцип роботи, але FD виконує менше операцій, тому потенційно може слугувати заміною у випадках, коли недостатньо швидкості роботи 3-FD.

1.3 Глибинні згорткові методи обробки відео та їх обмеження

Задача розпізнавання та класифікації візуальних образів на зображенні є комплексною і передбачає виконання кількох послідовних етапів, узагальнених на рис. 1.4.



Рисунок 1.4 – Блок-схема загального методу розпізнавання та класифікації об'єктів

Оскільки відео фактично є послідовністю окремих зображень (кадрів), які змінюються з певною частотою залежно від кількості кадрів у секунду, загальний підхід до обробки може бути однаковим як для статичних зображень, так і для відеопотоку [30, 42]. Для виконання такої задачі достатньо реалізувати пошук та класифікацію об'єктів на кожному кадрі. Водночас послідовні кадри можуть містити як мінімальні, так і значні відмінності, або взагалі не відрізнятися, що створює додаткові можливості для оптимізації обчислень і підвищення точності [28, 124].

Першим етапом більшості глибинних згорткових методів є попередня обробка вхідного зображення, що є загальноприйнятою практикою у сфері комп'ютерного зору [124]. Це пояснюється тим, що вихідні зображення часто містять надлишкові дані: різні формати пікселів, кольорові простори, кодування, які мають значення для відображення, але не впливають на якість аналізу. Тому доцільно приводити всі кадри до єдиного визначеного формату та розміру, що забезпечує стабільну роботу наступних етапів методів [28].

Наступним важливим етапом попередньої обробки є вилучення надлишкової інформації про кольори. У багатьох випадках переведення зображення у відтінки сірого дозволяє підвищити точність роботи алгоритмів, оскільки кольорова інформація не завжди несе корисні ознаки для розпізнавання (особливо у задачах детекції руху чи контурів) [28, 42]. Спрощення кольорової моделі також має ряд переваг з точки зору використання ресурсів: обчислення стають менш складними, зменшується навантаження на процесор та пам'ять завдяки зменшенню обсягу даних для аналізу.

Завершальним кроком попередньої обробки є фізична трансформація зображення, зокрема зміна розміру, перевероти або віддзеркалення кадру [30]. Використання єдиного фіксованого розміру для всіх зображень спрощує структуру моделі та забезпечує сталість параметрів вхідного шару. Масштабування кадрів до одного формату є доцільним, однак може супроводжуватися частковою втратою деталей, тому важливо застосовувати оптимальні алгоритми, що мінімізують спотворення навіть за умови незначного збільшення часу обробки [124].

«Обрізання» ділянок кадру є недоцільним, оскільки у задачах пошуку об'єктів неможливо гарантувати, що вилучена частина не містить важливих елементів сцени. Щодо переверотів і віддзеркалення, то ці операції можуть покращити стійкість алгоритму до змін ракурсу, проте кожна трансформація збільшує кількість обчислень, що критично для роботи у режимі реального часу [28, 30].

Наступним кроком після попередньої обробки є пошук потенційних об'єктів. Алгоритм, обраний для цього завдання, має відповідати кільком ключовим вимогам: забезпечувати достатню точність виявлення, можливість роботи у реальному часі, стабільність при обробці зображень низької якості або зменшеного розміру, а також стійкість до шумів, змін ракурсу та освітлення середовища [30, 42].

Потенційно для цього можуть використовуватися як класичні алгоритми комп'ютерного зору, так і методи машинного навчання.

Класичні підходи, як правило, складаються з фіксованого набору математичних операцій та трансформацій над пікселями кадру і залишаються незмінними незалежно від характеристик вхідних даних [28, 42]. Будь-які налаштування під специфічні умови або особливості середовища виконуються вручну. Для різних наборів даних нерідко доводиться розробляти кілька паралельних реалізацій алгоритму. Тобто відсутність автоматичної адаптації є суттєвим недоліком класичних методів при розв'язанні задач пошуку та класифікації відео об'єктів[124].

Альтернативним підходом до класичних алгоритмів комп'ютерного зору є використання методів машинного навчання, оскільки тут головний акцент зміщується з жорстко заданих операцій над пікселями на побудову адаптивної моделі [30, 42]. Основна перевага нейронних мереж полягає у гнучкості: вони здатні самостійно підлаштовувати параметри під характеристики вхідних даних, що суттєво розширює можливості практичного використання [124].

Одна і та сама архітектура моделі може змінювати ваги залежно від змін у середовищі, автоматично виявляючи приховані закономірності у даних. Це мінімізує потребу у ручному налаштуванні та зменшує кількість виняткових ситуацій. Завдяки такій адаптивності алгоритми глибинного навчання вважаються перспективними для вирішення задач пошуку та класифікації об'єктів на зображеннях і у відеопотоці [28]. Розглянемо їх більш докладно.

Згорткові нейронні мережі (CNN) – це спеціалізований клас нейронних мереж, розроблений для роботи із зображеннями [9]. Такі моделі складаються з послідовно з'єднаних шарів нейронів і імітують процеси, що відбуваються у зоровій корі людини під час обробки візуальної інформації [124].

Завдяки цій архітектурі вирішуються різноманітні задачі: розпізнавання облич, класифікація зображень, пошук аномалій на медичних знімках та інші [30, 42]. Головна особливість CNN полягає у тому, що вони автоматично виділяють найбільш інформативні ознаки у числовій формі з піксельних даних і використовують ці ознаки для прогнозу. Які саме характеристики будуть

вилучені, залежить від вхідних даних і завдання, що ставиться на етапі навчання моделі.

Кожен шар згорткової мережі працює з різним рівнем деталізації: перші шари найчастіше виявляють базові елементи, такі як контури або текстури, тоді як наступні шари опрацьовують дедалі більш абстрактні ознаки, формуючи високорівневі уявлення про об'єкт [30].

Задачу пошуку об'єктів на зображенні та їх класифікації можна розглядати як дві окремі, послідовно виконувані підзадачі, що ілюструється на блок-схемі загального методу (див. вище рис. 1.4). Результатом першого кроку є визначення координат об'єкта на зображенні, а наступним – це віднесення знайденого об'єкта до певного класу.

Сучасні алгоритми глибинного навчання поділяються на два типи залежно від того, чи виконують вони детекцію та класифікацію одночасно чи поетапно. Однопрохідні детектори (one-stage detectors), такі як YOLO або SSD, за один цикл аналізу кадру визначають положення об'єкта та його клас [13, 79]. Двохпрохідні (two-stage detectors), наприклад, сімейство моделей R-CNN – спочатку знаходять регіони, що потенційно містять об'єкти, а потім проводять їх класифікацію [44, 104]. Незалежно від підходу, обидва типи моделей фактично виконують другий і третій кроки схеми (рис. 1.5) у межах єдиної архітектури.

На сьогодні існує кілька модифікацій алгоритму R-CNN, зокрема Fast R-CNN та Faster R-CNN [44, 45]. Основною метою цих удосконалень стало значне прискорення роботи моделі при збереженні високої точності розпізнавання об'єктів. Початкова версія R-CNN обробляла одне зображення в середньому за 40 секунд, що робило її непридатною для задач реального часу [44]. Модифікація Fast R-CNN дозволила скоротити цей час до близько 2,5 секунд, а подальший розвиток у вигляді Faster R-CNN знизив час обробки одного кадру до приблизно 0,2 секунди [45, 104].

Відповідно, для завдань детекції та класифікації об'єктів на потоковому відео доцільно розглядати саме Faster R-CNN, як найбільш оптимізовану і ефективну версію цього сімейства алгоритмів.

На рис. 1.5 наведено спрощену архітектуру нейронної мережі Faster R-CNN, що демонструє її ключові структурні блоки.



Рисунок 1.5 – Структура мережі Faster R-CNN

З рис. 1.5 видно, що архітектура Faster R-CNN складається з двох основних модулів.

Першим модулем є глибока згорткова мережа із вбудованою мережею пропозицій регіонів – RPN (Region Proposal Network). Ця частина аналізує особливості, вилучені згортковими шарами, і формує припущення щодо областей кадру, які з найбільшою ймовірністю містять об'єкти. Завдяки цьому робота з піксельними даними безпосередньо стає необов'язковою, що дозволяє ефективно використовувати ресурси GPU [44, 104].

RPN стала принциповим поліпшенням у порівнянні з попередніми ітераціями, де для генерації регіонів використовувався алгоритм Selective Search на CPU. Хоча Selective Search базується на жадібному об'єднанні пікселів за низькорівневими ознаками, він працює значно повільніше за RPN [44]. Перехід до RPN дозволив суттєво прискорити детекцію: середня кількість запропонованих областей зменшилася з приблизно 2000 до 300 на наборі PASCAL VOC 2007, що скорочує кількість викликів класифікатора і загальний час обробки зображення.

Другим модулем є детектор Fast R-CNN, який обробляє запропоновані області та визначає клас об'єкта.

Важливою архітектурною особливістю є спільне використання згорткових шарів як мережею пропозицій регіонів, так і детектором. Це рішення

знижує вимоги до обсягу пам'яті та спрощує загальну структуру моделі. Разом з тим, воно ускладнює процес навчання, оскільки спільні шари повинні забезпечувати коректну роботу одразу для двох різних мереж. Розглянемо послідовність кроків, що реалізуються у Faster R-CNN.

Спочатку вхідне зображення, яке вже пройшло попередню обробку, передається до згорткової нейронної мережі для вилучення ознак і формування регіонів, що можуть містити об'єкти. На відміну від попередніх ітерацій R-CNN, пошук регіонів ґрунтується не на аналізі піксельних даних напряму, а на особливостях, які обчислюються згортковими шарами [44, 104].

Далі для кожного запропонованого регіону формується компактна мапа ознак, що подається до детектора Faster R-CNN. Цей детектор визначає наявність об'єкта у регіоні, його клас та уточнює межі, які займає об'єкт на зображенні.

Таким чином, завдяки розділенню процесу на два етапи: формування регіонів та класифікація, Faster R-CNN належить до двопрхідних детекторів [44]. З урахуванням часу обробки один кадр обробляється за $\approx 0,2$ секунди, що наближає модель до можливості роботи у реальному часі. Втім, за швидкодією вона все ще поступається однопрхідним рішенням на кшталт YOLO або SSD [45].

При обробці потокового відео можливе застосування додаткових оптимізацій, які можуть підвищити швидкість. Основною перевагою Faster R-CNN залишається висока точність локалізації та здатність виявляти дрібні об'єкти, що робить цей підхід актуальним для задач, де пріоритет має саме достовірність детекції.

Наступним перспективним алгоритмом для розв'язання задачі пошуку та класифікації об'єктів є YOLO (You Only Look Once) [104]. На відміну від Faster R-CNN, YOLO є монолітною однопрхідною моделлю, що одночасно визначає координати об'єкта та його клас за один цикл обробки зображення [13, 118].

З моменту появи першої версії YOLO алгоритм пройшов значну еволюцію. Зокрема, у YOLOv3 було впроваджено зміну каркасної моделі з Darknet-19 на Darknet-53, покращено стратегії навчання та налаштування

гіперпараметрів, що дозволило підвищити точність без суттєвої втрати швидкодії [13, 118]. Останні ітерації, такі як YOLOv8, пропонують кілька варіантів балансування між швидкістю роботи та точністю для різних типів пристроїв, включно з легковаговими версіями для вбудованих систем [78].

Ключова риса YOLO – це акцент на обробці в реальному часі, що робить цей підхід придатним для задач відеоспостереження та автономних систем. Разом з тим, до недоліків можна віднести обмеження, пов'язані зі схемою поділу зображення на клітинки $S \times S$: кожна клітинка прогнозує координати меж і клас лише одного об'єкта. Це знижує точність, якщо кілька дрібних об'єктів потрапляють у межі однієї клітинки [13].

Принцип роботи YOLO полягає у поділі зображення на решітку клітинок фіксованого розміру. Для кожної клітинки оцінюється ймовірність наявності об'єкта, координати його меж та належність до певного класу. На виході результати об'єднуються, і клітинки однакового типу зливаються, формуючи остаточні межі об'єктів (рис. 1.6).

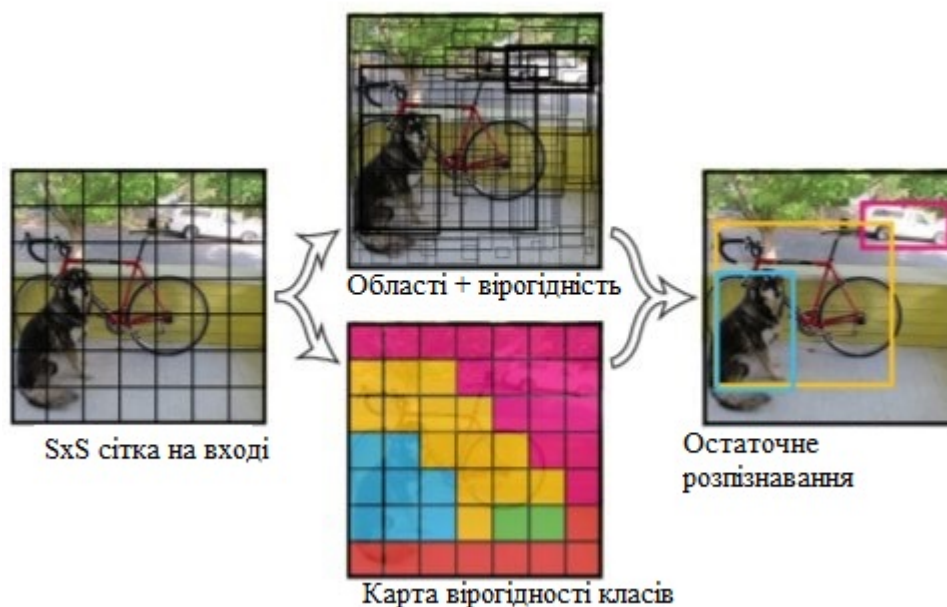


Рисунок 1.6 – Алгоритм роботи YOLO

З рис. 1.6 видно, що виявлення та класифікація об'єктів у YOLO виконуються за один прохід, тому цей підхід належить до однопрохідних

детекторів. Згідно з результатами досліджень [17, 118], останні версії YOLO, включаючи YOLOv8, забезпечують обробку понад 100 кадрів за секунду, що дозволяє використовувати їх для роботи у реальному часі навіть в умовах обмежених ресурсів.

SSD (Single Shot Detector) алгоритм для розпізнавання та класифікації візуальних образів на зображенні, який використовує згорткові нейронні мережі [79, 118]. Основною метою розробки алгоритму було підвищення швидкості роботи у порівнянні з попереднім передовим алгоритмом YOLO та покращити точність розпізнавання. Метою було досягти точності аналогічній двопрхідним детекторам як R-CNN або Faster R-CNN та надавати можливість роботи алгоритму у реальному часі. Розглянемо архітектуру моделі SSD на рис. 1.7.

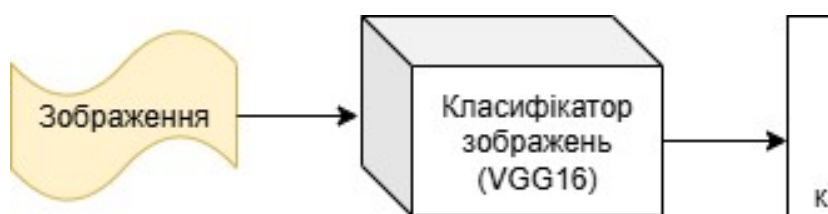


Рисунок 1.7 – Структура мережі SSD

З рис. 1.7 видно, що архітектура SSD структурно складається з двох основних компонентів.

Перший компонент – це базова глибока згорткова нейронна мережа, яка використовується для обчислення мап особливостей вхідного зображення. Для цього часто застосовують попередньо навчений класифікатор, з якого видаляють фінальний шар класифікації, щоб мати доступ до багаторівневих мап особливостей. Наприклад, у початковій реалізації SSD використовується VGG16 без вихідного шару класифікації [79].

Другий компонент – це додаткові згорткові шари SSD, які обробляють отримані мапи особливостей для локалізації меж об’єктів і їх класифікації.

SSD, як і YOLO, є однопрохідним детектором: зображення обробляється за один цикл без окремих етапів пошуку регіонів та класифікації.

На відміну від YOLO, SSD використовує систему «якірних меж» (anchor boxes). Алгоритм розбиває мапи особливостей на клітинки, кожна з яких прогнозує ймовірність наявності об'єкта та його клас. Для кожної клітинки визначено кілька якірних меж різного розміру й форми, що дозволяє виявляти об'єкти різних масштабів і кілька об'єктів всередині однієї клітинки [79].

Ключовою перевагою SSD є використання багаторівневих мап особливостей: чим глибше шар, тим більший розмір області зображення, яку він описує. Завдяки цьому SSD здатний точно детектувати об'єкти різного розміру, включаючи критично малі [80]. Це частково розв'язує обмеження YOLO, де кожна клітинка може описувати лише один об'єкт.

За результатами відомих досліджень, модель SSD300 при розмірі зображення 300×300 пікселів демонструє точність, що перевищує Faster R-CNN на 3% (на базі тієї ж VGG16) при швидкості обробки близько 59 кадрів за секунду, що дозволяє застосовувати її у режимі реального часу [79].

Для роботи з відеопотоком доцільно використовувати додатковий етап відстеження положення знайдених об'єктів, що дозволяє зменшити обчислювальне навантаження, уникаючи повторного пошуку на кожному кадрі. Такий підхід базується на поєднанні детекції та трекінгу, що буде детально розглянуто у подальших роботах.

Проведений експериментальний аналіз сучасних архітектур глибоких згорткових нейронних мереж для задач пошуку та класифікації візуальних об'єктів показав, що кожна з моделей має власний баланс між точністю та швидкістю обробки. На рис. 1.8 наведено порівняння точності моделей YOLOv8, SSD та Faster R-CNN, які були відібрані для подальших досліджень як найбільш перспективні з огляду на придатність до роботи у телекомунікаційних середовищах з обмеженими ресурсами.

Як видно з діаграми, Faster R-CNN демонструє найвищу точність (75 %), проте ця архітектура має значно більші обчислювальні затрати порівняно з YOLOv8, що забезпечує найнижчий показник точності (62 %) при найвищій

швидкодії. Модель SSD займає проміжну позицію з точки зору співвідношення точності (72 %) та вимог до ресурсів.

Отримані результати підтверджують доцільність комбінованого підходу, де менш ресурсоємні моделі можуть використовуватись у зв'язці з класичними методами для забезпечення стійкої роботи в умовах реального часу.

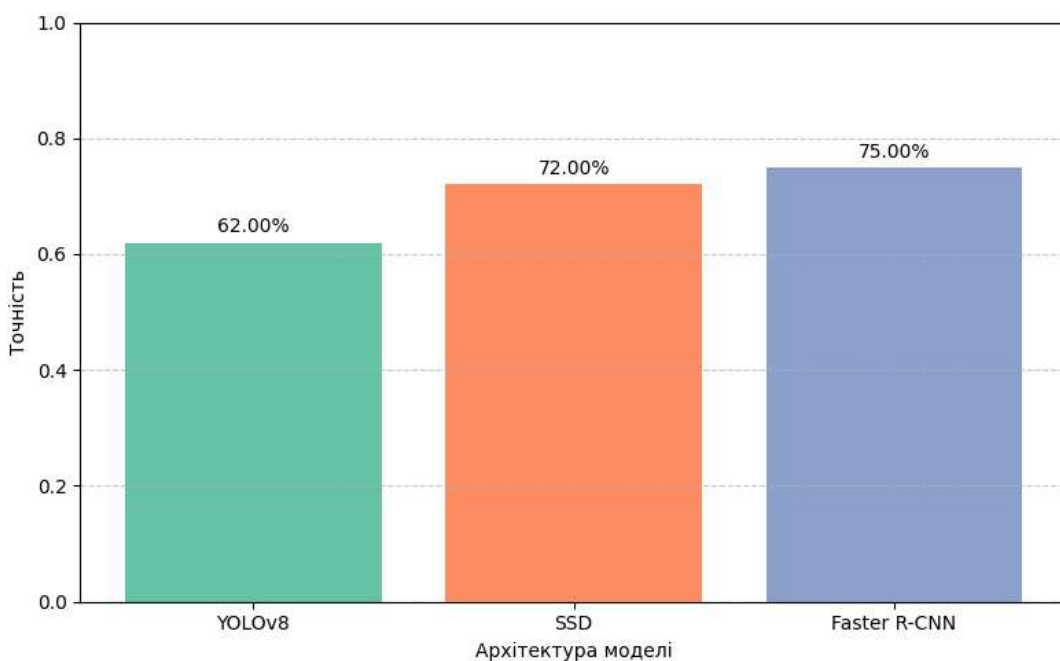


Рисунок 1.8 – Діаграма точності алгоритмів

Проте, точність не єдиний критерій, що враховується при виборі нейронної архітектури для використання у режимі реального часу. Для практичних телекомунікаційних задач важливою є швидкодія.

На рис. 1.9 наведено порівняння моделей за середнім показником обробки кадрів за секунду (FPS), що показує їх придатність до використання у режимі реального часу.

З рис. 1.9 видно, що модель YOLOv8 забезпечує найвищу швидкість обробки (понад 280 кадрів за секунду), що робить її придатною для використання у режимі реального часу. Натомість Faster R-CNN, хоча й демонструє високу точність (рис. 1.8), має найнижчий показник FPS, що обмежує його застосування на пристроях з обмеженими ресурсами.

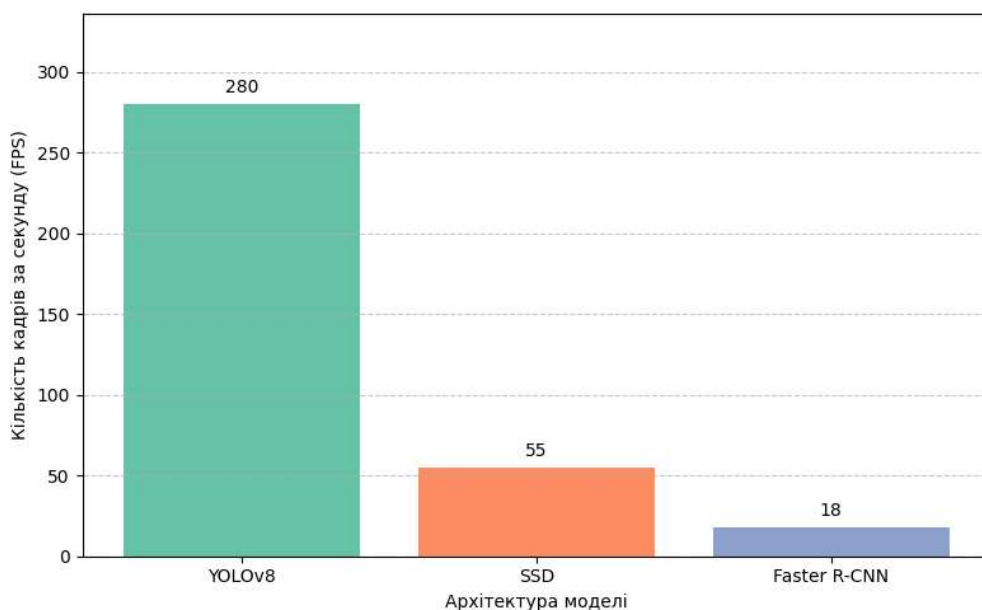


Рисунок 1.9 – Діаграма швидкості роботи алгоритмів

Модель SSD займає проміжну позицію, поєднуючи помірну швидкість з достатньою точністю, тому саме ця архітектура розглядається як базова для подальшого поєднання з класичними методами у запропонованому гібридному методі.

1.4 Обґрунтування доцільності розробки гібридного методу виявлення та класифікації об'єктів у відеопотоці

Проведений у Розділі 1 аналіз сучасних підходів до виявлення та класифікації об'єктів у відеопотоці показав, що більшість найбільш точних рішень ґрунтуються на використанні глибинних згорткових нейронних мереж. Такі моделі справді забезпечують високу якість розпізнавання об'єктів у стабільних умовах і на однорідному фоні, проте мають суттєві обмеження для використання у телекомунікаційних системах. По-перше, вони є ресурсоемними, що ускладнює їх реалізацію на граничних пристроях мережі. По-друге, глибинні згорткові моделі чутливі до різких змін умов зйомки: вплив завад, коливання фону, втрати якості відео через стиснення можуть призводити до помилок класифікації.

Класичні методи детекції руху, навпаки, відзначаються простотою, високою швидкістю та мінімальним використанням ресурсів, що робить їх придатними для застосування на ресурсно обмежених вузлах. Проте їхні результати часто є нестабільними в умовах динамічної сцени та за наявності завад.

Таким чином, окремо жоден з проаналізованих методів не забезпечує необхідного балансу між достовірністю, стійкістю та ресурсною ефективністю. Це обґрунтовує потребу у створенні гібридного методу, який поєднує переваги класичних детекторів і глибинних згорткових нейронних мереж для підвищення достовірності виявлення об'єктів у відеопотоці в режимі реального часу.

Отже, актуальним є вирішення науково-практичного завдання підвищення достовірності та стабільності розпізнавання об'єктів у відеопотоці за умов зосереджених завад і обмежених ресурсів шляхом розробки методу, який поєднує переваги класичних і нейромережових підходів та адаптується до змін навколишнього середовища.

Для реалізації мети дисертаційного дослідження необхідно розв'язати такі часткові задачі:

- провести порівняльний аналіз класичних та глибинних згорткових методів з урахуванням вимог телекомунікаційного середовища, оцінити їх точність, стійкість до завад та ресурсоємність;
- обґрунтувати доцільність створення гібридного методу на базі метода трикадрової різниці та легковагового нейрокласифікатора;
- розробити структуровану архітектуру алгоритму та формалізувати імітаційну модель, що описує повний цикл обробки відеопотоку в умовах змінного фону;
- дослідити механізми адаптації параметрів імітаційної моделі (пороги, фільтрація, класифікатор, трекер) до змін телекомунікаційного середовища та оцінити вплив адаптивності на продуктивність і стабільність роботи;
- провести аналітичну оцінку обчислювальної складності, моделювання та експериментальну перевірку функціонування методу на сценаріях, що

імітують роботу в умовах граничних пристроїв з обмеженими обчислювальними можливостями.

У сукупності, вирішення цих задач дозволить створити науково обґрунтований метод детекції та класифікації об'єктів, здатний забезпечити стабільну роботу в умовах динамічного фону, впливу завад і обмежених обчислювальних ресурсів.

Висновки за розділом 1

1. Проведено аналіз сучасних задач та умов обробки відео у телекомунікаційних системах, який показав, що основними викликами є змінність фону, вплив завад і обмеженість обчислювальних ресурсів (часто не більше 150 MB RAM та до 20 % CPU/GPU). Це дозволило сформулювати чіткі вимоги до методів відеоаналізу: стійкість до завад, стабільна робота при $FPS \geq 25$ та затримка не більше 100–200 мс.

2. Дослідження поширених форматів стиснення (H.264, H.265) та типових артефактів показало, що найбільш вразливими залишаються втрати якості на швидких сценах і блокування кадрів. Це обґрунтовує потребу у стійкій фільтрації та адаптивному оновленні моделей фону.

3. Порівняльна оцінка класичних методів детекції руху показала, що метод трикадрової різниці (3-FD) забезпечує F1-міру 0,61 при використанні лише 9 % CPU і близько 50 MB оперативної пам'яті. Це робить його оптимальним для впровадження на граничних вузлах з обмеженими ресурсами.

4. Аналіз глибинних згорткових моделей показав, що Faster R-CNN досягає найвищої точності (до 75 %), але потребує у 4–5 разів більше ресурсів у порівнянні з легковаговими моделями. Водночас YOLOv8 дозволяє обробляти відео зі швидкістю понад 280 кадрів за секунду, але за рахунок дещо меншої точності (≈ 62 –65 %).

5. Отримані результати аналізу дозволили зробити висновок, що жоден з методів окремо не забезпечує оптимального балансу між точністю, стійкістю до

завад та ефективністю використання ресурсів. Це обґрунтовує доцільність розробки гібридного методу, що поєднує переваги класичних детекторів руху та глибинних згорткових нейронних мереж.

6. Сформульовано завдання дисертаційного дослідження, спрямовані на розробку методу, здатного працювати у реальному часі з адаптацією до змін середовища та збереженням стабільної якості детекції і класифікації при обмежених ресурсах.

РОЗДІЛ 2

РОЗРОБКА МЕТОДУ DELTATRACK ДЛЯ ВИЯВЛЕННЯ ТА КЛАСИФІКАЦІЇ ОБ'ЄКТІВ У ВІДЕОПОТОЦІ

Аналіз сучасних методів виявлення та класифікації об'єктів у відеопотоці, проведений у Розділі 1, довів недостатню ефективність глибоких згорткових нейронних мереж в умовах обмежених обчислювальних ресурсів, а також їхню нестабільність у середовищах зі змінними характеристиками відеосигналу (зокрема освітлення, динамічний фон, наявність завад) [13, 16, 22, 37, 101, 102, 116, 124]. Встановлено, що точність моделей YOLO, SSD, Faster R-CNN значно знижується при обробці відео з низькою роздільною здатністю або за високого рівня завад, особливо на вбудованих пристроях реального часу [14, 23, 30, 40, 78].

Натомість методи, що базуються на аналізі міжкадрових різниць та фонового моделювання, виявили вищу продуктивність за умов обмежених обчислювальних ресурсів, хоча поступаються у стійкості до змін фону, освітлення та шумів [1, 2, 5, 6, 8, 10, 14, 15, 21, 28]. Найкращі результати в цьому класі методів показали гібридні підходи, які поєднують міжкадрову різницю з морфологічною фільтрацією [4, 6, 8, 10, 15, 19, 20, 26].

Незважаючи на наявні напрацювання, залишаються актуальними проблеми чутливості до варіацій освітлення, складності в ідентифікації перекритих об'єктів та висока частка хибних спрацьовувань у складних відеосценах. Окремим викликом є відсутність ефективних інтеграцій традиційних методів із сучасними моделями машинного навчання.

Таким чином, актуальною є розробка нового методу, який забезпечить адаптивне виявлення та класифікацію об'єктів у режимі реального часу, зберігаючи низькі обчислювальні витрати. При цьому метод повинен поєднувати простоту міжкадрового аналізу, стійкість до завад і можливість використання нейронної класифікації, що і стало предметом подальших досліджень в Розділі 2.

2.1 Аналіз і обґрунтування вибору базового підходу до виявлення руху у відеопотоці

Виявлення рухомих об'єктів у відеопотоці є основною задачею комп'ютерного зору, що визначає ефективність подальших етапів: сегментації, класифікації, трекінгу та аналізу поведінки.

Залежно від принципу виявлення змін у сцені, сучасні методи детекції руху поділяються на чотири класи [1, 3, 7] (табл. 2.1):

- віднімання фону (background subtraction) – ґрунтується на побудові та оновленні фонові моделі для кожного пікселя сцени, шляхом порівняння поточного значення з еталонним фоном для виявлення змін; застосовуються як статистичні моделі (GMM), так і нейроадаптивні методи (SOBS, Eigen-BG); підходить для стаціонарних камер, але є чутливим до динамічного фону та змін освітлення;

- оптичний потік (optical flow – OF) – використовує оцінку щільного або розрідженого поля векторів руху між кадрами, яке описує напрямок і швидкість переміщення кожного пікселя (алгоритми Lucas-Kanade, Horn-Schunck); забезпечує високу точність локалізації, але має високу обчислювальну складність та низьку придатність до застосування в режимі реального часу;

- міжкадрова різниця (FD, 3-FD) – базується на побудові абсолютної різниці між інтенсивностями пікселів у послідовних кадрах; може використовуватись у простій (двокадровій) або удосконаленій (трьохкадровій) формі; забезпечує високу швидкодію, не потребує фонові моделі, добре працює в умовах зміни сцени, але має обмежену повноту виявлення через втрату повільно рухомих або статичних об'єктів;

- глибинні згорткові нейронні мережі (CNN-based detectors) – застосовують попередньо навчені архітектури (YOLO, SSD, Faster R-CNN) для одночасної локалізації та класифікації об'єктів у кадрі; відзначаються високою точністю та адаптивністю до складних сцен, але вимагають значних обчислювальних ресурсів,

зазвичай потребують GPU, і мають обмежену продуктивність на вбудованих або енергоефективних пристроях.

Таблиця 2.1 – Порівняння підходів до виявлення руху у відеопотоці

Критерій	Віднімання фону (GMM, SOBS)	Оптичний потік	Міжкадрова різниця (FD, 3-FD)	Глибинні згорткові мережі (YOLO, SSD, Faster R-CNN)
Точність локалізації	Висока	Дуже висока	Середня	Дуже висока
Повнота виявлення	Середня	Висока	Низька – Середня	Висока
Стійкість до шуму	Середня	Висока	Середня – Висока	Низька – Середня
Обчислювальна складність	Висока	Дуже висока	Низька	Висока – дуже висока
Потреба в навчанні / моделі	Модель фону	Ні	Ні	Потребують навчання
Адаптивність до змін сцени	Середня	Висока	Висока	Залежна від якості навчання
Придатність до реального часу	Обмежена	Низька	Висока	Обмежена (залежить від GPU)
Особливості	Залежить від стабільного фону	Точний, дорогий	Швидкий, простий	Точний, ресурсоємний, навчений наперед

*розроблено автором

Як видно з табл. 2.1, методи віднімання фону забезпечують добру локалізацію об'єктів, але є чутливими до змін освітлення та динамічного фону і потребують складної ініціалізації. Оптичний потік гарантує високу точність локалізації, проте характеризується високою обчислювальною складністю, що обмежує його використання в умовах реального часу. Міжкадрова різниця забезпечує високу швидкодію та простоту реалізації без потреби у фоновій моделі, що є важливою перевагою для пристроїв з обмеженими ресурсами. Глибинні

згорткові мережі показують найвищу точність виявлення та класифікації, проте мають значне ресурсне навантаження, що ускладнює їх застосування у вбудованих або мобільних пристроях без апаратного прискорення.

У даному дослідженні запропоновано гібридний підхід, який поєднує міжкадрову різницю (3-FD, FD) як швидкий метод попереднього виявлення змін сцени та згорткову нейронну мережу для подальшої класифікації виявлених об'єктів.

Використання 3-FD, FD виправдане такими його перевагами.

1. Низька обчислювальна складність, що дає змогу працювати в реальному часі навіть на вбудованих пристроях.

2. Адаптивність до змін середовища, таких як освітлення, рухомий фон чи відеошум, без необхідності попереднього навчання.

3. Зменшення кількості регіонів, що передаються в нейронну мережу, що істотно знижує загальне навантаження на систему.

Для обґрунтування вибору цього підходу в табл. 2.2 та на рис. 2.1 наведено порівняльну оцінку класичних методів виявлення руху за показниками Precision, Recall, F-міри, а також FPR, FNR та PWC (Percentage of Wrong Classifications). Метрики для згорткових моделей наведено за даними з публікацій [106, 114, 123] та оцінками на наборах CDnet-2014/COCO /VOC.

Таблиця 2.2 – Порівняння методів виявлення руху

Метод	Повнота	Точність	F-міра	FPR	FNR	PWC
FD	0,25	0,61	0,22	0,030	0,75	3,4
3-FD	0,09	0,71	0,08	0,010	0,91	3,9
GMM	0,22	0,68	0,26	0,004	0,78	3,1
Simp-SOBS	0,45	0,62	0,32	0,015	0,55	4,2
RAF	0,37	0,60	0,24	0,010	0,63	4,8
Eig-Bg	0,59	0,48	0,38	0,060	0,41	6,1
SSD	0,81	0,79	0,80	0,020	0,19	2,3
YOLOv3	0,84	0,82	0,83	0,015	0,16	2,0
Faster R-CNN	0,87	0,85	0,86	0,012	0,13	1,8

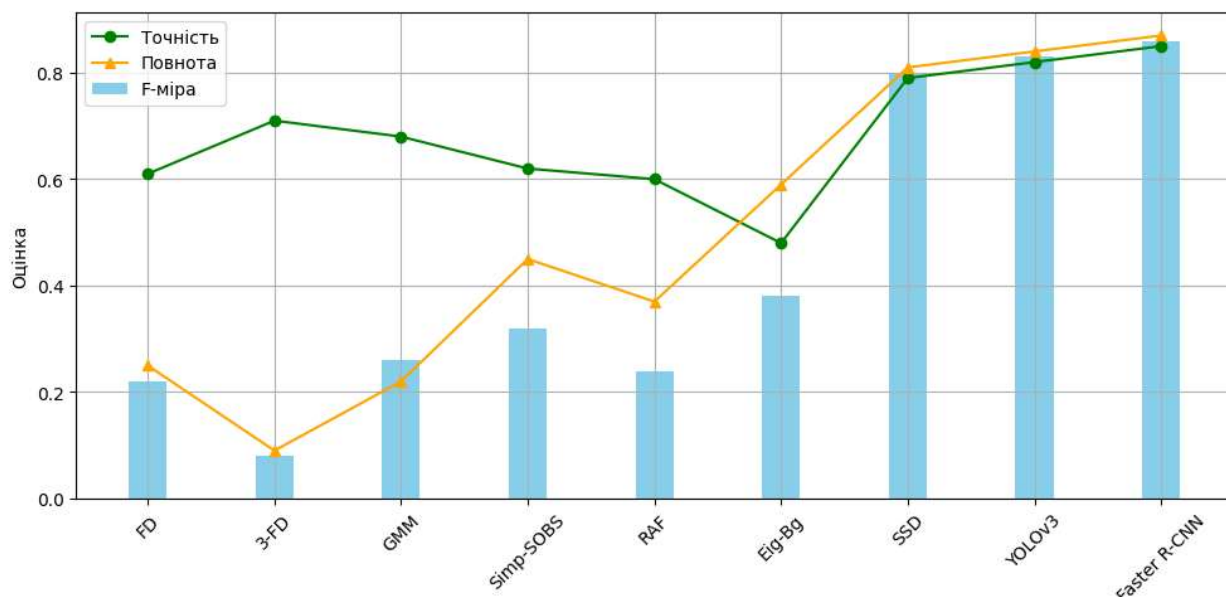


Рисунок 2.1 – Порівняльна оцінка методів виявлення руху

*розроблено автором

На рис. 2.1: точність позначена зеленою лінією з колами, повнота – помаранчевою лінією з квадратами, а F-міра – блакитними стовпчиками.

Як видно з табл. 2.1 та рис. 2,1, методи глибокого навчання (SSD, YOLOv3, Faster R-CNN) показують високі значення всіх основних показників. F-міра для Faster R-CNN досягає 0,86, що свідчить про високу узгодженість між точністю та повнотою. Водночас, класичні підходи, 3-FD, хоча і поступаються згортковим моделям за інтегральними метриками, мають значно меншу обчислювальну складність та швидше реагують на локальні зміни у відеопотоці. Наприклад, у 3-FD при високій точності (0,71) спостерігається низька повнота (0,09), що пояснюється жорсткістю фільтрації та спрощеними правилами детекції.

Таким чином, згорткові нейронні мережі забезпечують високу точність за рахунок багаторівневої обробки ознак, але потребують значних обчислювальних ресурсів. А прості методи, наприклад, 3-FD, є легкими для реалізації, придатними для попередньої обробки та швидкої локалізації змін, але мають обмежену здатність до узагальнення.

Зважаючи на те, що в даному дослідженні застосовуються методи і алгоритми в умовах обмежених ресурсів, доцільним є порівняння легких у

реалізації методів виявлення руху, які традиційно застосовуються в системах відеоспостереження реального часу. Серед них особливу увагу приділено трьом підходам, які найчастіше реалізуються на практиці: базовому FD, його модифікації 3-FD, а також моделі MoG, як прикладу складнішого та обчислювально затратного методу.

Таблиця 2.3 – Порівняння основних методів виявлення руху*

Метод	Точність	Повнота	F-міра
FD	0,47	0,23	0,19
3-FD	0,46	0,08	0,08
MoG	0,61	0,21	0,25

*Складено автором на основі даних з [106]

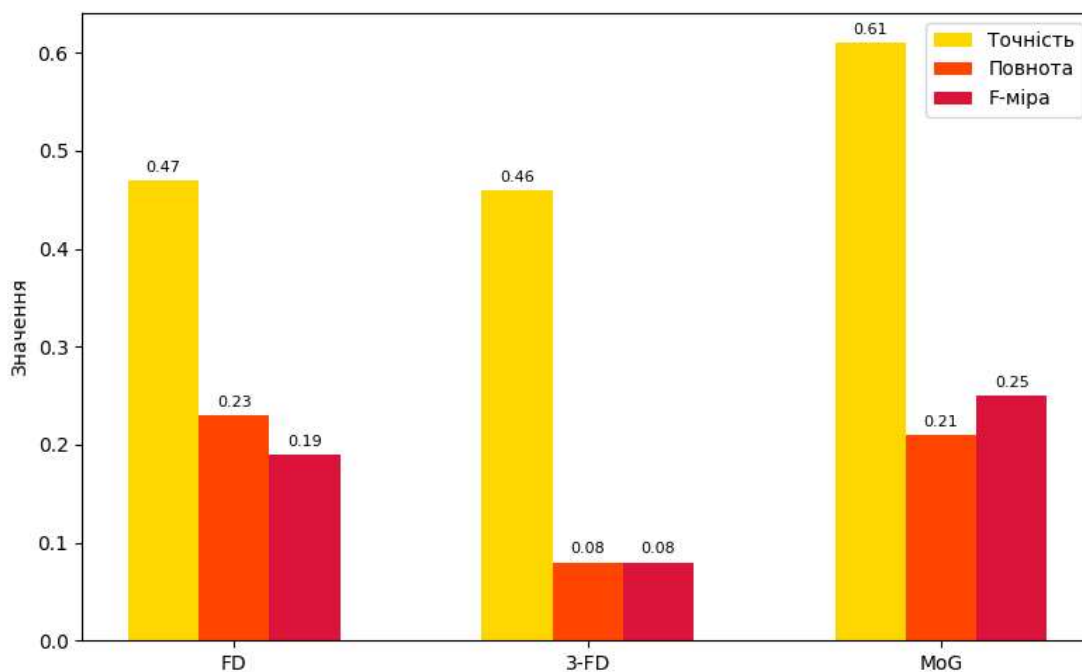


Рисунок 2.2 – Порівняння основних методів виявлення руху*

*Складено автором на основі даних з [106]

Як видно з розрахунків, хоча MoG має кращу F-міру, його реалізація є складною та ресурсоемною, що ускладнює використання в умовах реального часу. У той час як FD має прийнятні показники точності та просту реалізацію, що робить його придатним для вбудованих систем.

Разом з тим, серед простих методів особливо виділяється модифікований підхід 3-FD, який хоч і показує нижчі інтегральні результати роботи, проте дозволяє більш жорстко відсікати фонові коливання та забезпечує вищу локалізаційну точність змін. Це робить його перспективним для використання як попередній детектор змін, що має низьку обчислювальну складність і стабільну поведінку в шумових умовах.

На рис. 2.3 наведено структурну схему методу 3х кадрової міжкадрової різниці, в якій для кожної піксельної координати (x,y) обчислюється мінімальне значення між абсолютними різницями з попереднім і наступним кадрами. Саме така конфігурація лежить в основі порівняння FD і 3-FD у табл. 2.3.

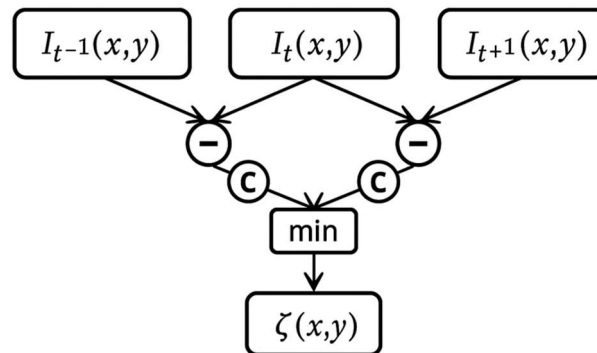


Рисунок 2.3 – Схема алгоритму міжкадрової різниці (3-FD)

Схема на рис. 2.3 структурує загальний процес обчислення змін між трьома послідовними кадрами: попереднім $I_{t-k}(x,y)$, поточним $I_t(x,y)$ і наступним – $I_{t+k}(x,y)$. Для кожної піксельної координати визначається мінімальне значення між абсолютними різницями з попереднім та наступним кадрами, що дозволяє ефективно пригнічувати випадкові флуктуації яскравості та зменшувати вплив шумів.

На основі цієї схеми побудовано базову формалізовану модель методу трьохкадрової міжкадрової різниці (удосконалена модель представлена в Розділі 3), яка враховує просторово-часову природу відеопотоку й описує основні етапи обробки змін. Математичне визначення маски змін представлено формулою:

$$D(x, y) = \min(|I_t(x, y) - I_{t-k}(x, y)|, |I_t(x, y) - I_{t+k}(x, y)|) \quad 2.1$$

Маска змін $D(x, y)$ порогово бінаризується за умовами:

$$B(x, y) = \begin{cases} 1, \text{ якщо } D(x, y) > T, \\ 0, \text{ інакше } D(x, y) < T \end{cases} \quad 2.2$$

На наступних етапах, отримана бінарна маска $B(x, y)$ обробляється морфологічними операціями, які описуються формулою:

$$B^{\circ}(x, y) = (B \circ K) \cdot K, \quad 2.3$$

де $\circ$ – це операція відкриття, $\cdot$ – операція закриття; K – структурний елемент.

У результаті формується цілісний регіон змін, який може бути переданий на подальші етапи класифікації та трекінгу. Згідно з проведеним експериментальним аналізом, базовий підхід міжкадрової різниці (FD) забезпечує оптимальне співвідношення між точністю виявлення змін і обчислювальною ефективністю в умовах обмежених апаратних ресурсів.

Удосконалена конфігурація – трьохкадрова різниця (3-FD) дозволяє підвищити стійкість до шумів та фонових флуктуацій, що робить її придатною для подальшого розвитку. У подальших підрозділах цей підхід буде модифіковано з урахуванням просторово-часової структури відеопотоку, характеру змін у сцені, а також шляхом підвищення достовірності виявлення об'єктів.

2.2 Архітектура та алгоритм реалізації методу DeltaTrack

Назва DeltaTrack є скороченням від Delta Frame Tracking і відображає дві особливості запропонованого методу.

1. Delta – використання міжкадрових різниць (трюхкадрова 3-FD–конфігурація) як швидкого способу виявлення змін у відеопотоці.

2. Track – впровадження компонента відстеження об'єктів, що активується після їх розпізнавання та дозволяє зменшити кількість повних циклів обробки.

Окрім цього, важливою складовою є вбудований модуль класифікації, реалізований на основі згорткової нейронної мережі (CNN), що дозволяє автоматично ідентифікувати об'єкти після їх сегментації. Така гібридна архітектура об'єднує обчислювальну ефективність методу 3-FD/FD з високою точністю, яку забезпечує CNN, і забезпечує адаптацію до умов реального часу без втрати достовірності.

Таким чином, DeltaTrack поєднує:

- швидке попереднє виявлення змін за допомогою 3-FD/FD;
- морфологічну фільтрацію й відсікання неінформативних регіонів;
- класифікацію знайдених об'єктів згортковою нейромережею;
- оптимізоване відстеження без необхідності повторної обробки кадрів.

Таке поєднання дає змогу адаптувати метод до різних сценаріїв відеоспостереження з урахуванням ресурсних обмежень та вимог до точності.

Порівняння DeltaTrack з базованими методами наведено в табл. 2.4.

Таблиця 2.4 – Порівняння базових та запропонованого DeltaTrack

Ознака	FD / 3-FD	DeltaTrack
Міжкадрова різниця	Так	Так
Використання трюх кадрів	Ні (в FD), Так (у 3-FD)	Так (гнучка адаптивна схема)
Морфологічна обробка	Обмежено (лише ерозія або дилатація)	Так (відкриття + закриття)
Порогова фільтрація за площею	Відсутня	Так
Адаптивна фільтрація шумів	Не передбачена	Так
Класифікація знайдених об'єктів	Зовнішня або неавтоматизована	Вбудований модуль класифікації (CNN)

Відстеження об'єктів між кадрами	Відсутнє	Так (зменшує кількість повних детекцій)
Можливість роботи в реальному часі	Частково (лише для FD)	Так
Обчислювальна ефективність	Висока	Висока (з оптимізацією переходів між етапами)

*розроблено автором

Загальна архітектура методу DeltaTrack реалізована у вигляді покрокового алгоритму, блок-схема якого представлена на рис. 2.4.



Рисунок 2.4 – Блок-схема алгоритму виявлення об'єктів у відеопотоці

*розроблено автором

Блок-схема відображає основні етапи обробки: попереднє перетворення вхідного кадру, виявлення змін на основі міжкадрової різниці, морфологічну фільтрацію, а також передавання сегментованих областей до модуля

класифікації. Якщо класифікація завершилася успішно, активується режим відстеження об'єкта; в іншому випадку виконується обробка наступного кадру.

Основні позначення та параметри алгоритму наведено в табл. 2.5.

Таблиця 2.5 – Основні позначення показників

Позначення	Характеристика параметра
I_t	Кадр отриманий в момент часу t
$Frame(t)$	Функція (або джерело) захоплення відеокадру з потоку в момент часу
$I_t(x, y)$	Значення пікселя у поточному кадрі
$I_{t-k}(x, y), I_{t+k}(x, y)$	Значення пікселя в сусідніх кадрах (для 3-FD)
$D(x, y)$	Абсолютна різниця між пікселями кадрів
$T - Threshold$	Порогове значення для виявлення значущих змін
$B(x, y)$	Порогова бінарна маска руху
$B^{\wedge}(x, y)$	Маска змін після морфологічної обробки
K	Структурний елемент (ядро) для морфологічних операцій
R_i	Виділений регіон зміни (область інтересу)
R_{valid}	Набір регіонів, що пройшли фільтрацію за розміром
A_{min}	Експериментально визначене мінімальне порогове значення (наприклад, 100 пікселей)
$O(B), C(B)$	Морфологічні оператори відкриття та закриття відповідно
w, h	Ширина і висота зображення у пікселях
C	Обчислювальне навантаження на кадр (кількість елементарних операцій)
T_{frame}	Час обробки одного кадру
γ	Середня кількість обчислень на один піксель (для етапу масштабування, перетворення тощо)
$\oplus$	Операція дилатації
$\ominus$	Операція ерозії

Розглянемо більш детально основні етапи та удосконалення запропонованого методу DeltaTrack відповідно до науково-практичних завдань дисертаційного дослідження.

Етап 1. Захоплення поточного кадру.

На початковому етапі здійснюється вилучення поточного кадру I_t з відеопотоку $I_t = \text{Frame}(t)$, який виступає вхідними даними для всіх подальших операцій. Цей процес виконується або у режимі реального часу, або з наперед заданим часовим інтервалом (наприклад, кожні 50 мс), що дозволяє забезпечити стабільний моніторинг сцени та оперативне реагування на появу нових об'єктів.

Надійність, частота та точність захоплення кадрів визначають ефективність роботи алгоритму в цілому, тобто від якості вхідного сигналу залежить точність виявлення змін і об'єктів.

Етап 2. Масштабування зображення до єдиного формату

Після захоплення кадру, його масштабують до стандартного розміру 1280x720 пікселів (формат 720p). Така уніфікація спрощує подальшу обробку, оскільки швидкість виконання більшості операцій безпосередньо залежить від кількості пікселів у кадрі. Крім того, це дозволяє уникнути варіацій продуктивності при зміні роздільної здатності вхідного відео. Більшість нейромережових моделей також орієнтується на фіксований розмір вхідного зображення, що оптимізує їх продуктивність. Стабільність розміру кадру особливо важлива для порівняння ефективності різних методів та для побудови масштабованих рішень у ресурсно обмежених середовищах. До того ж, уніфікований розмір є необхідною умовою для пакетної обробки зображень або розгортання на пристроях із низькою продуктивністю.

Швидкість виконання лінійних трансформацій прямо пропорційна кількості пікселів у кадрі, що можна розрахувати за формулою:

$$C = h \cdot \omega \cdot \gamma, \quad 2.4$$

Таким чином, час обробки одного кадру приблизно лінійно залежить від розміру вхідного зображення:

$$T_{frame} \sim \mathcal{O}(h \cdot \omega), \quad 2.5$$

Для порівняння продуктивності обробки кадрів з різною роздільністю можна оцінити коефіцієнт прискорення при переході до фіксованого розміру:

$$\text{Acceleration ratio} = \frac{h_{\text{BX}} \cdot \omega_{\text{BX}}}{1280 \cdot 720}, \quad 2.6$$

Без уніфікації розміру продуктивність алгоритму варіюється в залежності від якості відео, що ускладнює порівняння з іншими методами та знижує надійність. У випадках з обмеженими обчислювальними ресурсами можлива додаткова оптимізація, це зменшення вхідної роздільної здатності з метою прискорення обробки.

На Етапі 2 було проведено удосконалення, а саме.

На відміну від класичних реалізацій FD/3-FD, які працюють зі змінними розмірами кадрів, у методі DeltaTrack впроваджено жорстке масштабування до єдиного формату 1280×720. Це забезпечує стабільність обробки в реальному часі, прогнозованість ресурсного навантаження та придатність до роботи на апаратно обмежених пристроях (табл. 2.6).

Таблиця 2.6 – Особливості обробки розміру зображення в методах виявлення руху

Метод	Обробка змінного розміру	Уніфікований розмір	Стабільність обробки	Придатність до реального часу
FD / 3-FD	Так	Відсутній	Низька	Обмежена при великих кадрах
SSD, YOLO	Потребує масштабування	Наприклад, 300×300	Висока	Так, за умов оптимізації
DeltaTrack Net	Масштабування на вході	1280×720	Висока	Так (оптимізовано)

*розроблено автором

Етап 3. Перетворення кадру у відтінки сірого.

На цьому етапі кольорове зображення, представлене в просторі RGB, перетворюється у градації сірого. Це дозволяє знизити обчислювальну складність, оскільки замість трьох каналів обробляється лише один. Стандартна формула перетворення виглядає так:

$$I_t^{gray}(x, y) = 0,2989 \cdot R + 0,5870 \cdot G + 0,1140 \cdot B, \quad 2.7$$

Значення коефіцієнтів (0,2989, 0,5870, 0,1140) відповідають сприйняттю яскравості людським зором відповідно до стандарту ITU-R BT.601 [4].

Етап 4. Обчислення міжкадрової різниці

Після переходу до градацій сірого здійснюється обчислення змін між трьома послідовними кадрами (3-FD): попереднім $I_{t-k}(x, y)$, поточним $I_t(x, y)$ і обробляємим – $I_{t+k}(x, y)$, а також з застосуванням адаптивного коефіцієнта α ($\alpha, 0 < \alpha \leq 1$), що дозволяє враховувати динаміку сцени. Загальна формула (2.1) міжкадрової різниці [8, 51] набуває наступного аналітичного вигляду:

$$D(x, y) = \min(\alpha |I_t(x, y) - I_{t-k}(x, y)|, \alpha |I_t(x, y) - I_{t+k}(x, y)|) \quad 2.8$$

Етап 5. Формування бінарної маски змін

На основі отриманого поля різниць $D(x, y)$ виконується бінаризація з використанням подвійного порогування T , що дозволяє зменшити кількість хибнопозитивних пікселів у зонах незначних змін і підвищити надійність виділення істотних регіонів.

Бінарна маска формується за наступними умовами:

$$B(x, y) = \begin{cases} 1, \text{ якщо } D(x, y) > T_{high} \\ 0, \text{ якщо } D(x, y) < T_{low} \\ 1, \text{ якщо } T_{low} \leq D(x, y) \leq T_{high} \wedge \exists (x', y') \in N(x, y): B(x', y') = 1 \\ 0, \text{ інакше} \end{cases} \quad 2.9$$

де T_{high} та T_{low} – нижній та верхній пороги;

$N(x, y)$ – околиця пікселя (x, y) , наприклад 3×3

– умова для середнього порогу означає, що піксель включається до маски, якщо поруч вже є пікселі з маскою = 1 (ефект розширення контурів).

Виконання вищезазначених умов дозволяє надійно виділяти істотні зміни інтенсивності (у випадках, коли значення $D(x, y)$ перевищує верхній поріг T_{high}), ефективно пригнічувати слабкі флуктуації та шум (якщо $D(x, y) < T_{low}$, а також враховувати слабші зміни у випадках, коли вони просторово прилягають до вже виявлених активних пікселів (тобто $D(x, y)$ потрапляє між двома порогамі, але поруч є сусідній піксель із маскою 1). Значення порогів T_{high} та T_{low} підбираються експериментально з урахуванням рівня шуму у відеопотоці та характеру сцени.

Етап 6. Морфологічне очищення маски змін.

Внаслідок впливу шумів, часткового руху та неоднорідних меж об'єктів, сформовані регіони змін можуть містити артефакти, тобто внутрішні порожнини, розриви контурів, ізольовані пікселі тощо. Для усунення таких спотворень отримана маска $B(x, y)$ додатково обробляється послідовними морфологічними трансформаціями: відкриттям $O(\cdot)$ та закриттям $C(\cdot)$ [9]. Операція відкриття виконується за формулою:

$$O(B(x, y)) = (B(x, y) \ominus K) \oplus K, \quad 2.10$$

Операція закриття:

$$C(B(x, y)) = (B(x, y) \oplus K) \ominus K, \quad 2.11$$

Ерозія $B(x, y) \ominus K$ звужує об'єкти, видаляючи крайні пікселі, тоді як дилатація $B(x, y) \oplus K$, навпаки розширює їх, оскільки додає пікселі по

периметру. Відкриття дозволяє ефективно видалити дрібні шуми або ізольовані пікселі, які не належать до основного об'єкта. Закриття, навпаки, сприяє заповненню порожнин всередині виділених регіонів та поєднує фрагментовані частини.

Ці операції виконуються із використанням структурного елементу K . На практиці, найчастіше використовуються невеликі квадратні ядра (наприклад, розміром 3×3), які ковзають по зображенню. Розмір та форма ядра визначають силу морфологічного перетворення: надто мале ядро може не забезпечити достатнього ефекту, тоді як надто велике значно збільшує обчислювальне навантаження і ризикує змінити контури виділених об'єктів.

Операції відкриття та закриття застосовуються послідовно, формуючи очищену фінальну маску $B'(x,y)$:

$$B'(x,y) = C \left((O(B(x,y))) \right) = [(B(x,y) \ominus K) \oplus K] \oplus K \ominus K \quad 2.12$$

У випадку часткового руху об'єкта сформована маска може охоплювати лише його частину, що ускладнює подальшу класифікацію. Для компенсації цього ефекту застосовується дилатація з більшим розміром ядра, що дозволяє розширити отримані регіони змін, захопивши також найближчі околиці, включаючи нерухомі частини об'єктів, які належать до однієї логічної групи.

Візуалізація процесу формування маски змін на основі міжкадрової різниці та морфологічної обробки представлена на рис. 2.5.

На рис. 2.5 послідовно подано етапи обробки відеокادру в градаціях сірого: обчислення різниці між кадрами $D(x,y)$, бінаризація за порогом (T – Threshold), утворення первинної маски змін $B(x,y)$ та її подальша морфологічна обробка.

Синій шар позначає операцію відкриття $O(B(x,y))$, червоний – закриття $C(B(x,y))$, які усувають шуми та заповнюють незамкнені контури. У результаті

формується очищена бінарна маска $B'(x,y)$, що передається на наступні етапи аналізу відеопотоку.

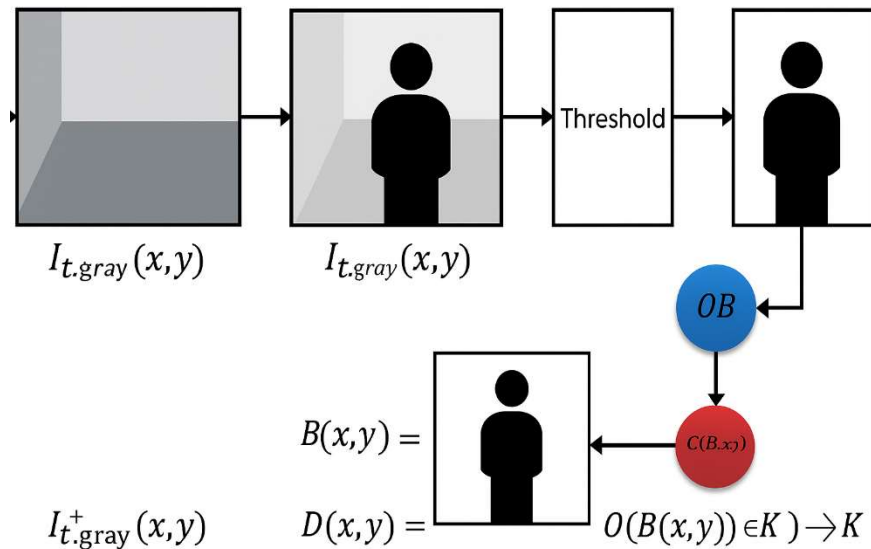


Рисунок 2.5 – Процес формування маски змін

Етап 7. Виділення зв'язних компонентів та фільтрація рухомих об'єктів

Після формування бінарної маски змін $B'(x,y)$, очищеної морфологічними операціями, наступним етапом є виділення зв'язних компонентів (регіонів), що утворюються з пікселів зі значенням 1, які є суміжними за 4- або 8-сусідством. Усі такі області формують множину R_i . Однак не всі з них є значущими з погляду детекції об'єктів. Часто серед компонент трапляються залишкові шуми або тіні, які, хоча і пройшли пороговання, не мають достатньої площі. Тому застосовується фільтрація за площею:

$$R_{valid} = \{R_i | area(R_i) \geq A_{min}\} \quad 2.13$$

Це дозволяє виключити неінформативні компоненти і передати на наступний етап лише потенційно значущі об'єкти. Така фільтрація знижує кількість хибнопозитивних спрацювань і підвищує точність класифікації.

Етап 8. Класифікація та трекінг об'єктів

Кожен валідований регіон $R_i \in R_{valid}$ масштабується до фіксованого розміру $W \times H$, нормалізується і передається на вхід згорткової нейронної мережі-класифікатора f_θ . Таким чином, CNN оперує не з повним зображенням, а лише з тими сегментами, які попередньо виявлені через метод міжкадрової різниці та морфологічну фільтрацію, що суттєво знижує обчислювальні витрати.

$$\hat{y}_i = f_\theta(R_i) \quad 2.14$$

У реалізації використовується модель на базі MobileNetV1, що забезпечує достатню точність при низькому обчислювальному навантаженні [118, 123]. Параметри згорткової нейронної мережі-класифікатора (MobileNetV1) представлені в табл. 2.7.

Модель на базі MobileNetV1 відповідає запропонованому гібридному методу DeltaTrack, де роль CNN зосереджена на остаточній валідації виявлених регіонів. При необхідності модель може бути адаптована під вимоги конкретної задачі за критеріями швидкодії, пам'яті чи точності.

Таблиця 2.7 – Параметри згорткової нейронної мережі-класифікатора (MobileNetV1)

Параметр	Значення / Коментар
Вхідний розмір зображення	$W \times H = 128 \times 128$
Нормалізація	Пікселі масштабуються до діапазону $[0, 1]$
Тип архітектури	MobileNetV1
Глибина мережі	28 шарів (включно з глибинно-сепарабельними згортками)
Кількість класів	Більше 1000 видів об'єктів, можливо розширення
Фінальна активація	Softmax
Оптимізатор	Adam, $\alpha = 10^{-4}$
Функція втрат	Крос-ентропія

Час обробки одного регіону	< 15 мс (на CPU, Intel i5), < 5 мс (на GPU)
Модельна вага	~4,3МВ
Формат виводу	Прогнозований клас $\hat{y}_l \in \{0,1\}$

**розроблено автором*

Після класифікації регіону як об'єкта, що потребує подальшого спостереження, активується модуль відстеження.

Для цього використовується MOSSE-трекер, що працює у частотній області та забезпечує високу продуктивність у режимі реального часу [59, 82]. Оновлення відбувається кожні N кадрів (де $N=10$ у базовій реалізації).

У разі втрати об'єкта трекер скидається, і запускається новий цикл виявлення. Порівняння властивостей трекерів у реальному часі представлено в табл. 2.8 [106].

Таблиця 2.8 – Порівняння трекерів для реального часу*

Алгоритм трекінгу	Швидкодія	Стійкість до завад	Пам'ять	Придатність до реального часу
MOSSE	Висока	Середня	Низька	Так
KCF	Середня	Висока	Середня	Так
CSRT	Низька	Висока	Висока	Обмежено

**Складено автором на основі даних з [106]*

Таким чином, запропонований алгоритм завершується поєднанням ефективної попередньої обробки, глибокої класифікації та адаптивного трекінгу, що забезпечує стійку роботу в реальному часі навіть у складних динамічних сценах.

У підрозділі 2.3 наведено результати експериментального моделювання, які дозволяють кількісно оцінити ефективність запропонованого методу.

2.3 Експериментальна оцінка ефективності методу DeltaTrack у порівнянні з методом SSD та MoG

Для верифікації ефективності розробленого методу DeltaTrack, що базується на трьохкадровій міжкадровій різниці, морфологічній обробці, класифікації та трекінгу, проведено серію експериментів з аналізу швидкодії, точності та ресурсного навантаження. Для порівняння за критеріями поширеності, репрезентативності класу та наявності незалежних реалізацій було обрано два аналогічні методи:

- SSD – як збалансований представник глибоких згорткових мереж для виявлення об'єктів у реальному часі (Faster R-CNN - повільний, але точний, YOLOv3 - швидший за SSD, але менш точний);
- MoG – найскладніший з класичних, але дуже часто використовується для порівняння зі згортковими мережами.

Експериментальна оцінка проводилася в однакових умовах: відеопотік з фіксованої камери, відсутність попереднього навчання на експериментальному датасеті, обробка в реальному часі.

Таке поєднання дозволяє оцінити DeltaTrack як у порівнянні з класичними методами, що не потребують навчання, так і з сучасними нейромережевими підходами. Основна увага приділялася метрикам точності (F-міра), швидкодії (FPS) та ресурсного навантаження при виконанні на пристроях із середнім рівнем продуктивності.

Додатково проведено узагальнене порівняння з іншими класичними методами (табл. 2.9 – складено автором за матеріалами [106]).

Як видно з табл. 2.9, кожен з підходів має свої особливості за параметрами: часова адаптивність, необхідність у фоновому моделюванні, тип вхідних даних, чутливість до шумів.

Зокрема, метод 3-FD, який покладено в основу запропонованого в даному дослідженні методі, має низьку обчислювальну складність, не потребує

попереднього навчання чи збереження фону, а також дозволяє адаптивно реагувати на зміну сцени.

Таблиця 2.9 – Основні підходи до виявлення руху

Підхід	Скорочення	Параметри	Реальний час	Зміни освітлення	Складність	Ресурси
Міжкадрова різниця	FD	–	Так	Низька	Низька	Низькі
Трикадрова різниця	3-FD	–	Так	Середня	Середня	Низькі
Ковзне середнє	RAF	$\alpha = 0.1$	Так	Низька	Середня	Помірні
Темпоральний морфол. градієнт	FMTG	$\alpha = 0,1$	Ні	Висока	Висока	Високі
Оцінка фону методом $\Sigma\Delta$	$\Sigma\Delta$	$N = 3$	Так	Низька	Низька	Низькі
Детекція руху на основі MRF	MRFM D	$\beta_s, \beta_p, \beta_f, \alpha$	Ні	Висока	Висока	Високі
Ентропія простору-часу	STEI	$3 \times 3 \times 5, Q = 100$	Ні	Середня	Висока	Високі
Різницева STEI	DSTEI	$3 \times 3 \times 5, Q = 100$	Ні	Середня	Висока	Високі
Гаусове згладжування	RGA	$\alpha = 0,01, D = 2,5$	Так	Висока	Середня	Помірні
Суміш Гауссіанів	MoG	α, T, D	Частково	Висока	Висока	Високі
Власне тло	Eig-Bg	$N = 28, M = 3$	Ні	Висока	Висока	Високі
Самонав. віднімання фону SOBS	Simp-SOBS	$\alpha_1 = 0,02, \alpha_2 = 0,01$	Частково	Середня	Висока	Помірні

На відміну від складніших моделей (наприклад, MoG, MRFMD, STEI), він не вимагає багатопараметричного налаштування та працює з мінімальними ресурсами, що критично важливо для реального часу.

Для оцінки запропонованого методу об'єктного аналізу відеопотоку було визначено основні показники, які дозволяють проаналізувати як якість виявлення об'єктів, так і ресурсну ефективність реалізації. Вибір показників обумовлений поширеністю в літературі з комп'ютерного зору, а також практичною значущістю в умовах обмежених обчислювальних ресурсів.

1. Середній час обробки одного кадру (мс) – показник, який дозволяє оцінити придатність алгоритму до роботи в реальному часі, тобто забезпечення обробки ≥ 25 кадрів на секунду.

2. Точність розпізнавання (%) – це частка правильно класифікованих об'єктів серед усіх виявлених, що свідчить про загальну якість моделі.

3. Використання центрального процесора (CPU), графічного процесора (GPU) та оперативної пам'яті (RAM) – відображає обчислювальні витрати алгоритму при виконанні, важливі для інтеграції в мобільні чи вбудовані пристрої.

4. Використання відеопам'яті (VRAM) – це критичний показник для пристроїв з обмеженим графічним ресурсом, особливо у випадках застосування нейронних мереж.

5. Кількість знайдених об'єктів – це кількісна характеристика чутливості алгоритму, яка враховує не лише точність, але й повноту виявлення.

У табл. 2.10 наведено порівняльні результати для трьох методів, а саме: класичного MoG, згорткової мережі SSD та запропонованого гібридного методу DeltaTrack. Оцінювання охоплює основні показники ефективності: швидкодію, точність, використання апаратних ресурсів.

Як видно з таблиці, DeltaTrack досягає найкращої швидкодії, забезпечуючи майже 41% вигаш у часі порівняно з MoG та випереджаючи SSD.

Таблиця 2.10 – Порівняння ефективності методів детекції об’єктів

Параметр	MoG	SSD	DeltaTrack	Коментар
Середній час обробки, мс	9,2	5,8	5,4	DeltaTrack найшвидший (на ~41% швидше за MoG)
Точність, %	68,4	74,0	71,2	SSD найточніший, але DeltaTrack лише на 2,8% відстає
Використання CPU, %	7,5	9,0	11,5	Вищий у DeltaTrack через попередню обробку
Використання RAM, МБ	350	445	468	Найменше у MoG; у DeltaTrack зростає через маски
Використання GPU, %	0,0	0,9	1,1	MoG не використовує GPU
Використання VRAM, МБ	0	610	390	DeltaTrack економніший за SSD на 57%

Незважаючи на дещо нижчу точність порівняно з SSD, гібридний метод DeltaTrack за розрахунками суттєво менше споживання відеопам’яті та стійкості до зміни сцени, що робить його перспективним для застосування в реальному часі на пристроях із обмеженими ресурсами.

На рис. 2.7–2.13 подано візуалізацію основних показників, наведених в табл. 2.10, що дозволяє наочно оцінити переваги та обмеження методу DeltaTrack у порівнянні з класичними та нейромережевими методами.

Як видно з рис. 2.7, метод DeltaTrack демонструє найнижчий середній час обробки одного кадру серед усіх порівнюваних підходів, що робить його найбільш придатним для задач реального часу. Попри те, що SSD має незначно вищу точність (що буде показано на наступних рисунках), запропонований гібридний метод досягає кращої швидкодії завдяки ефективному поєднанню просторово-часової фільтрації та селективної класифікації. Класичний метод MoG, хоча і широко використовується, поступається обом як за точністю, так і за продуктивністю, що обмежує його придатність у сучасних сценаріях.

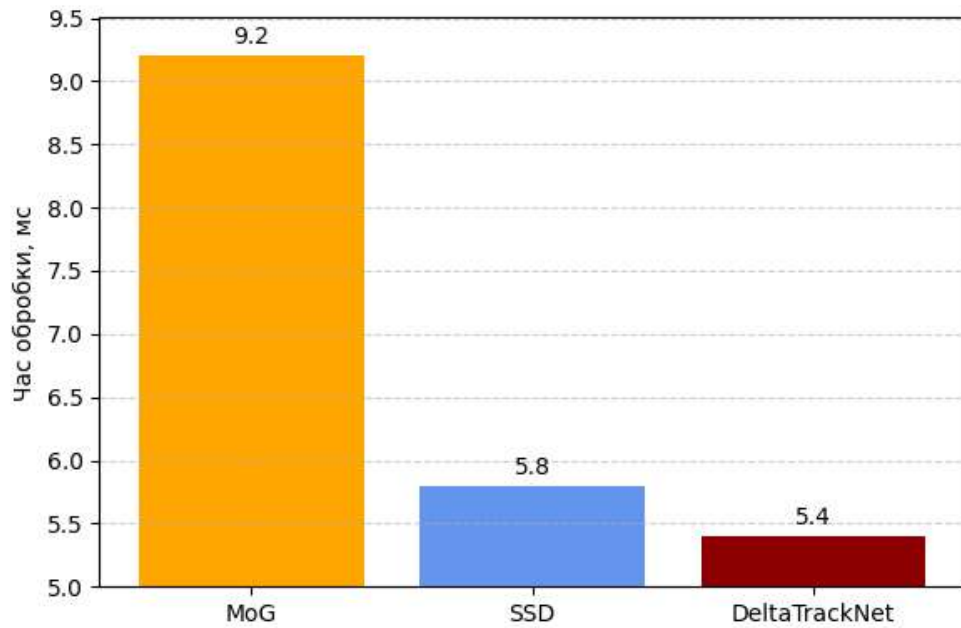


Рисунок 2.7 – Порівняння середнього часу обробки одного кадру

Крім того, DeltaTrack забезпечує обробку одного кадру в середньому за 5,4 мс, що на ≈ 0.4 мс (або 8%) швидше за SSD. Отже, за цим критерієм він краще підходить для розгортання у вбудованих системах, пристроях з обмеженим обчислювальним потенціалом або у застосуваннях, де критичною є мінімальна затримка обробки.

Рис. 2.8 ілюструє показники точності розпізнавання.

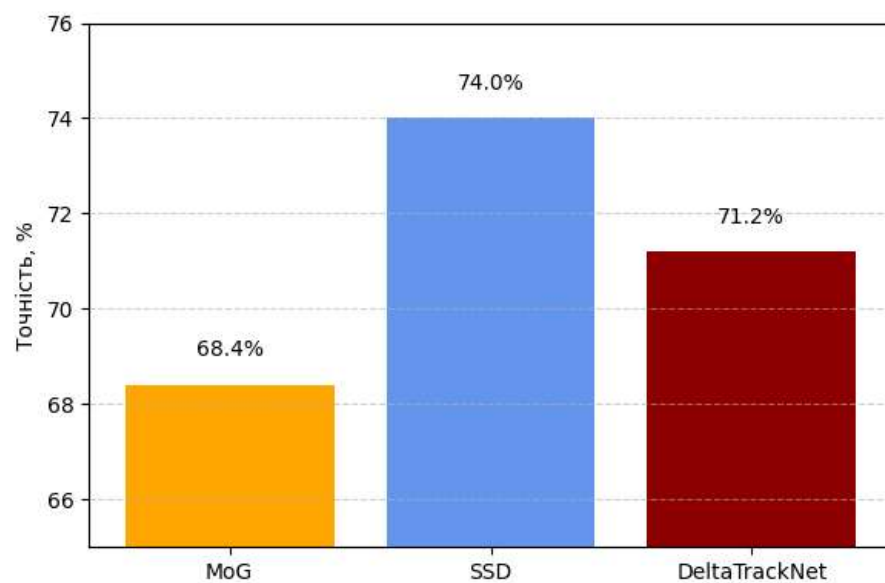


Рисунок 2.8 – Порівняння точності розпізнавання

Як свідчать результати експериментального порівняння, метод DeltaTrack досягає точності 71,2%, що лише на 2,8% поступається методу SSD. Зважаючи на істотний вииграш у швидкодії та зниженому ресурсному навантаженні, така різниця є допустимою та свідчить про ефективність використання DeltaTrack.

На рис. 2.9 представлено рівень використання процесора (CPU) під час обробки відеопотоку.

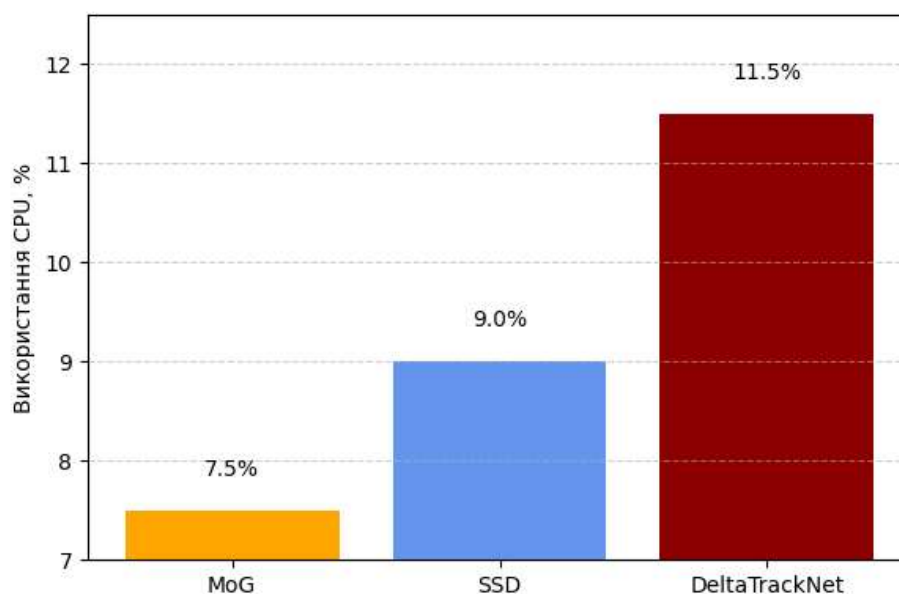


Рисунок 2.9 – Порівняння використання ресурсів процесору

Як видно з рис. 2.9, DeltaTrack показує дещо вищий рівень використання CPU (11,5%) порівняно з SSD і MoG, що зумовлено додатковими етапами попередньої обробки відеокадрів. Однак це навантаження залишається в межах, допустимих для систем реального часу. Важливо позначити що це прості лінійні операції, що знижує вірогідність проблем пов'язаних з сумісністю алгоритму з процесору\графічним прискорювачом приладу. Водночас така архітектура сприяє зниженню навантаження на графічний модуль та відеопам'ять, що є важливою перевагою в умовах обмежених апаратних ресурсів.

Підвищене залучення процесора в методі DeltaTrack, зокрема на етапах попередньої обробки зображень (видалення завад, морфологічна фільтрація),

обумовлює зростання обсягів проміжних даних, що зберігаються в оперативній пам'яті (рис. 2.10).

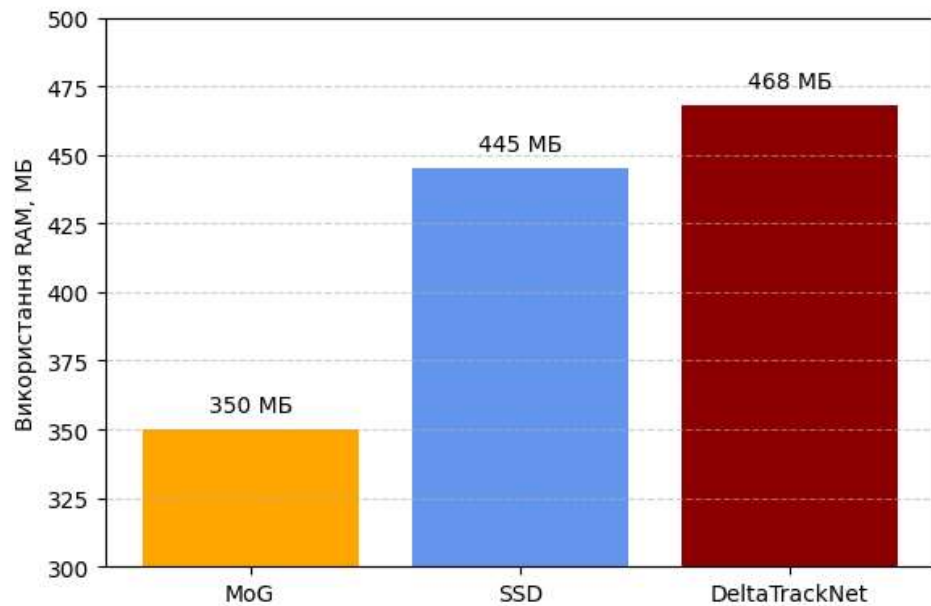


Рисунок 2.10 – Порівняння використання RAM

Як видно з рис. 2.10, DeltaTrack використовує 468 МБ RAM, що на 5,2% більше, ніж SSD. Це обумовлено збереженням масок змін та проміжних результатів на етапах попередньої обробки перед передаванням регіонів до класифікатора.

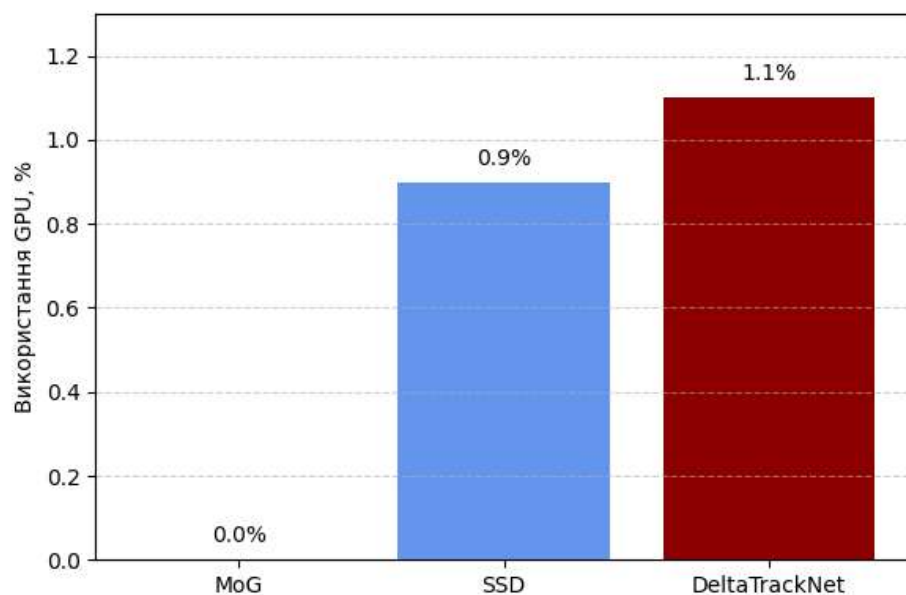


Рисунок 2.11 – Використання ресурсів GPU

На рис. 2.11 представлено порівняння використання обчислювальних потужностей графічного процесора (GPU), що залучаються при виконанні попередньої обробки кадрів.

Запропонований метод DeltaTrack показує незначно вищу активність GPU (1,1% проти 0,9% у SSD), що пов'язано з виконанням частини лінійних операцій над зображеннями саме на графічному процесорі. Така особливість дозволяє розвантажити CPU та підвищити загальну продуктивність, зокрема в умовах обмежених ресурсів.

На рис. 2.12 оцінено споживання відеопам'яті (VRAM).

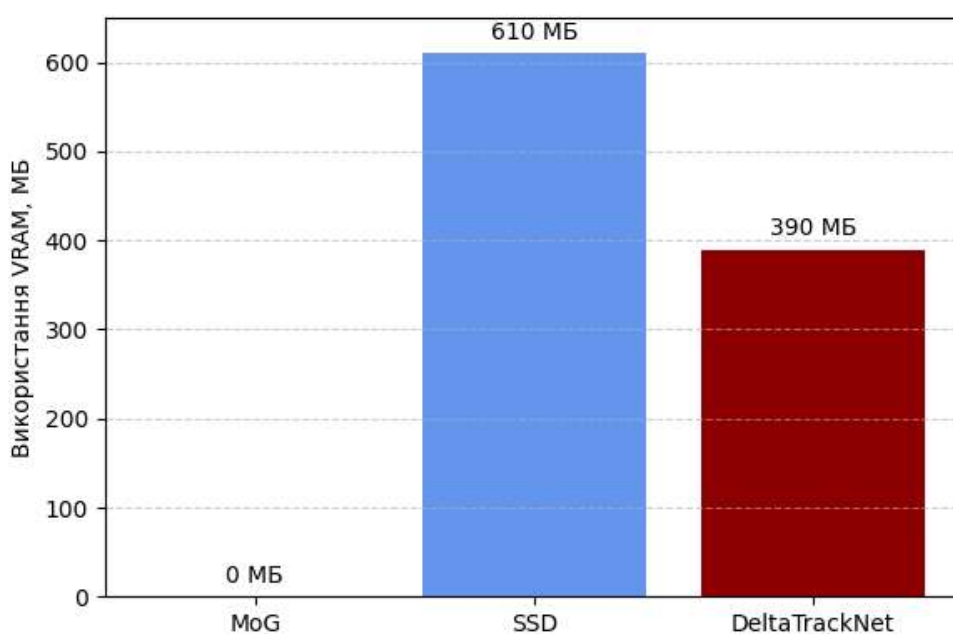


Рисунок 2.12 – Порівняння використання VRAM

Метод SSD споживає приблизно на 57% більше відеопам'яті, ніж DeltaTrack, що обумовлено використанням глибоких згорткових мереж, які потребують значних обсягів VRAM для зберігання проміжних тензорів і параметрів. У розробленому ж методі більшість обчислень виконується до етапу класифікації, що забезпечує економне використання графічних ресурсів і підвищує його придатність до впровадження у середовищах з обмеженим обсягом відеопам'яті, якими за звичай є Edge прилади.

Таким чином, результати експериментів підтверджують ефективність та практичну придатність запропонованого методу DeltaTrack.

По-перше, запропонований метод на основі трьохкадрової міжкадрової різниці, морфологічної фільтрації та селективного трекінгу забезпечує вищу швидкодію: середній час обробки становить 5,4 мс, що на $\approx 8\%$ менше, ніж у SSD (5,8 мс), і дозволяє стабільно працювати в умовах реального часу.

По-друге, DeltaTrack демонструє ефективніше використання відеопам'яті – 390 МБ проти 610 МБ у SSD, що становить виграш у 57%. Це особливо важливо для впровадження на вбудованих платформах або в середовищах з обмеженими графічними ресурсами.

Попри незначне зниження точності розпізнавання (71,2% проти 74,0% у SSD), метод забезпечує оптимальний баланс між точністю та обчислювальною ефективністю, з помірним зростанням навантаження на CPU (+2,5%) та RAM (+23 МБ).

Узагальнено, розроблений підхід є раціональним компромісом між продуктивністю, точністю та ресурсною економністю. Його застосування доцільне в задачах відеомоніторингу, інтелектуального відеоспостереження та автономної аналітики поточкових даних, зокрема на пристроях реального часу з обмеженим апаратним забезпеченням.

Висновки за розділом 2

1. Обґрунтовано вибір трикадрової міжкадрової різниці (3-FD) як базового підходу до виявлення руху у відеопотоці. У результаті порівняльного аналізу з FD, MoG, Simp-SOBS, Eigen-BG та іншими методами встановлено, що 3-FD забезпечує оптимальний баланс між точністю, швидкодією та обчислювальною ефективністю в умовах обмежених ресурсів. Звичайний метод міжкадрової різниці (FD) може бути використаний з метою ще більшого прискорення.

2. Розроблено метод DeltaTrack, який об'єднує поетапну обробку відеопотоку: масштабування кадру, перетворення в градації сірого, трикадрова різниця, застосування подвійного порогу, морфологічна обробка, класифікація (MobileNetV1) та трекінг (MOSSE). Метод не потребує попереднього навчання і демонструє стійкість до фонових змін та нестабільного освітлення.

3. Запропоновано застосування подвійного порогу при формуванні маски змін, що дозволяє надійно виділяти істотні об'єкти, пригнічувати шум і гнучко адаптуватися до слабких змін у сусідстві. Такий підхід підвищує точність сегментації без збільшення обчислювального навантаження.

4. Удосконалено етап морфологічного очищення маски змін шляхом послідовного застосування операцій відкриття та закриття, що дозволило значно зменшити кількість хибнопозитивних фрагментів і сформувати стабільні регіони для подальшої класифікації.

5. Реалізовано фільтрацію рухомих регіонів за площею та інтеграцію з класифікатором і трекером. Доведено, що поєднання MobileNetV1 із MOSSE-трекером дозволяє підтримувати відстеження об'єктів навіть у разі їх часткового зникнення з кадру, із мінімальним споживанням відеопам'яті.

6. Проведено експериментальне порівняння з методами SSD і MoG, що представляють згорткові та класичні підходи відповідно. Запропонований метод забезпечує менший час обробки (5,4 мс проти 5,8 мс у SSD та 9,2 мс у MoG), нижче споживання відеопам'яті (390 МБ проти 610 МБ у SSD) та порівнянну точність (71,2% проти 74,0% у SSD і 68,4% у MoG), демонструючи переваги гібридного підходу.

7. Застосування згорткової нейронної мережі MobileNetV1 на фінальному етапі класифікації дозволило поєднати високу точність розпізнавання з низьким обчислювальним навантаженням. Такий гібридний підхід вигідно відрізняється від повністю нейронних методів (як SSD), які потребують суттєво більше ресурсів і менш гнучкі до адаптації.

РОЗДІЛ 3

ФОРМАЛІЗОВАНА ІМІТАЦІЙНА МОДЕЛЬ МЕТОДУ ДЕТЕКЦІЇ ТА ТРЕКІНГУ DELTATRACK В УМОВАХ ДИНАМІЧНОГО ФОНУ

У сучасних дослідженнях в області відеоаналізу значна увага приділяється задачам виявлення, розпізнавання та трекінгу об'єктів у відеопотоці. Наукові дослідження [1, 3, 8, 10, 14, 21] обґрунтовують різні підходи до виявлення змін сцени: від простої міжкадрової різниці (FD, 3-FD) до складніших моделей фону та глибоких нейронних мереж (YOLO, SSD). Зокрема, у [1] показано переваги швидких методів локалізації змін у сцені, однак відсутня чітка імітаційна модель, яка б формалізувала всі етапи обробки.

Дослідження [5, 16, 22, 26] вказують на важливість точного та ресурсоощадного обчислення у вбудованих або обмежених середовищах, але обмежуються лише описом архітектур без формального обґрунтування їх дії. Наприклад, методи, засновані на YOLO [13, 101, 116], мають високу точність, але їх обчислювальна складність обмежує застосування в реальному часі на малопотужних пристроях. У роботах [14, 15, 20, 21] розглянуто використання трекерів (MOSSE, KCF, CSRT), проте їх аналіз не супроводжується налаштуванням імітаційної моделі оцінки навантаження та ефективності.

Автори [4, 6, 9, 12, 17] розглядають попередню обробку зображення (масштабування, перетворення у відтінки сірого, морфологічні операції), проте більшість з них подають лише алгоритмічні описи без формального математичного апарату. Аналогічно, у [28, 30, 42, 106] проводяться огляди методів, але бракує узагальнених моделей, які б охоплювали весь процес виявлення та трекінгу – від кадрювання до генерації мітки класу.

В роботах [23, 58, 84, 100] представлено легковагові нейромережеві архітектури для класифікації об'єктів. Проте навіть у цих дослідженнях відсутня поетапна формалізація всього процесу: немає загального виразу для обчислювального ланцюга та математичного описання інструментів і процедур.

У зв'язку з цим, у Розділі 3 розроблено формалізовану імітаційну модель методу DeltaTrack, яка поєднує обчислювально ефективні процедури (3-FD, морфологічні операції) з класифікатором на основі нейронної мережі та адаптивним трекером. Для реалізації поставленої мети вирішуються такі завдання:

- представлення методу у вигляді сукупності математичних функцій і логічних умов для кожного з етапів (попередня обробка, виявлення, фільтрація, класифікація, трекінг);
- визначення параметрів методу, що дозволяють адаптувати його до обмежених обчислювальних ресурсів та різної динаміки сцени;
- обґрунтування коректності та ефективності кожного з етапів, з урахуванням шумів, змін освітлення та особливостей відеопотоку;
- включення до моделі оцінок обчислювальної складності, часу обробки кадру та точності виявлення;
- забезпечення гнучкості моделі для її подальшої адаптації до нових умов, типів задач або джерел відео.

3.1 Структура та формалізація імітаційної моделі методу DeltaTrack

Формалізована модель DeltaTrack є науково обґрунтованим математичним представленням методу виявлення та трекінгу об'єктів у відеопотоці, на основі трьохкадрової різниці, морфологічних перетворень, нейромережевої класифікації та адаптивного трекінгу (Розділ 2). На відміну від проаналізованих вище традиційних підходів, які зосереджені на описі архітектур або алгоритмічних етапів реалізації, запропонована модель має вигляд послідовності математичних перетворень з чітким визначенням функцій, змінних і налаштованих параметрів для кожного етапу

Таким чином, запропонована імітаційна математична модель виступає і як інструмент подальшого програмного впровадження, і як формальна основа

для доведення переваг методу DeltaTrack та розв'язання часткових наукових задач дослідження. Далі наведено структуру моделі з математичними виразами.

Загальна імітаційна модель методу має вигляд:

$$T_t = \text{Track} \circ C \circ \mathcal{F} \circ \text{Morph} \circ \text{Thresh} \circ \text{Diff} \circ \text{Preprocess}(F_t, F_{t-1}, F_{t-2}); (\Theta) \quad 3.1$$

де F_t, F_{t-1}, F_{t-2} – це послідовні кадри відео;

$\Theta = \{\theta_p, \theta_d, \theta_b, A_{min}, \theta_c, \theta_t\}$ – сукупність параметрів моделі (табл. 3.1);

Preprocess – попередня обробка кадру (масштабування, сірий колір);

Diff – обчислення різниці кадрів (3-FD/FD);

Thresh – порогова бінаризація;

Morph – морфологічна обробка;

$\mathcal{F}$ – фільтрація регіонів за площею;

C – класифікація валідних регіонів;

Track – оновлення треків для виявлених об'єктів.

Таблиця 3.1 – Параметри моделі Θ , поділені на групи

Параметр	Характеристика
θ_p	Параметри попередньої обробки (роздільна здатність 1280×720, коефіцієнти сірого, тип інтерполяції)
θ_d	Параметри міжкадрової різниці (розмір вікна, коефіцієнт згладжування)
θ_b	$\theta_b = T_{low}, T_{high}$ параметри порогової фільтрації
θ_c	Параметри нейронного класифікатора
θ_t	Параметри трекера (тип, частота оновлення, режим втрати цілі).
A_{min}	Мінімальна площа регіону

Усі функції в формулі (3.1) працюють у множині дискретних двовимірних зображень $\mathbb{Z}^{w \times h}$, де w, h – відповідно ширина та висота кадру.

Схематично модель представлена на рис. 3.1, де показано зв'язок між основними обчислювальними модулями та їхніми вхідними/вихідними параметрами. Блок-схема відображає послідовну природу обробки та дозволяє виділити вузли, найбільш чутливі до ресурсного навантаження або завад сцени.



Рисунок 3.1 – Структура імітаційної моделі DeltaTrack

Розглянемо ці операції математичної функції у вигляді кроків.

1. Вхідні дані та попередня обробка

Кожен кадр відеопотоку на часовій позиції t позначається як F_t . Попередня обробка охоплює масштабування до уніфікованого розміру та переведення зображення у відтінки сірого. Ці трансформації дозволяють стандартизувати дані перед подачею на подальші етапи.

Загальна функція попередньої обробки описується виразом:

$$I_t = \text{Preprocess}(F_t; \theta_p), \quad 3.2$$

Окрема увага в запропонованій моделі приділяється масштабуванню, яке є важливим для уніфікації розмірів зображень перед подачею до нейронного класифікатора. Масштабування реалізується у вигляді координатного перетворення:

$$I_{out}(x', y') = I_{in}\left(\left\lfloor \frac{x' \cdot W_{in}}{W_{out}} \right\rfloor, \left\lfloor \frac{y' \cdot H_{in}}{H_{out}} \right\rfloor\right), \quad 3.3$$

де $I_{out}(x', y')$ – зображення після масштабування; x', y' – нові координати відповідних пікселів після масштабування; W_{in}, H_{in} – початковий розмір зображення, $W_{out} = 1280, H_{out} = 720$ – розмір зображення узгоджений з експериментальними розрахунками (див. Розділ 2), I_{in} – початкове зображення, $\lfloor \cdot \rfloor$ - оператор округлення значень пікселів до цілих чисел вниз.

2. Обчислення трьохкадрової різниці

Обчислюється карта змін між трьома послідовними кадрами, що дозволяє враховувати як появу, так і зникнення об'єктів у сцені. Математично вираз записується як:

$$D_t = \max(|I_t - I_{t-1}|, |I_{t-1} - I_{t-2}|), \quad 3.4$$

Ця функція дозволяє виявити локальні зміни незалежно від напрямку руху.

3. Порогова бінаризація.

Після обчислення карти змін застосовується двоступенева порогова сегментація з урахуванням сусідства, яка виконує перетворення у бінарну карту активності (див. формулу (2.4)). Її суть полягає у фіксації інтенсивних змін та розширенні активних зон через сусідні пікселі з проміжними значеннями.

4. Морфологічне очищення та фільтрація регіонів

До B_t послідовно застосовуються морфологічні операції:

$$B_t^{morph} = \text{Close}(\text{Open}(B_t)), \quad 3.5$$

Після чого фільтруються області, площа яких менша за A_{min} :

$$R_t = \{R_i \in \text{Connected}(B_t^{morph}) \mid A(R_i) \geq A_{min}\} \quad 3.6$$

5. Класифікація об'єктів

Кожен регіон $R_i \in R_t$ подається на вхід нейронної мережі-класифікатора:

$$l_i = C(R_i; \theta_c), \quad 3.7$$

де C – функція класифікації;

θ_c – параметри моделі (наприклад, MobileNetV1); l_i – мітка класу.

6. Трекінг об'єктів

Ідентифіковані об'єкти передаються у трекер:

$$T_t = \text{Track}(\{(R_i, l_i)\}; \theta_t), \quad 3.8$$

де θ_t – параметри унікальні для алгоритма трекінгу (наприклад, для MOSSE);

T_t – множина треків на момент t .

Таким чином, побудована імітаційна модель забезпечує формалізацію методу DeltaTrack, а також створює підґрунтя для аналітичної оцінки ефективності (розділ 3.2) та адаптації до змінних умов середовища (розділ 3.3).

3.2 Аналітична оцінка обчислювальної складності параметрів формалізованої імітаційної моделі

На основі структури, представленої в підрозділі 3.1, необхідно виконання аналітичної оцінки обчислювальної складності основних функціональних блоків моделі DeltaTrack. Оцінка дозволяє визначити етапи з найвищими витратами обчислювальних ресурсів, а також обґрунтувати вибір параметрів при реалізації методу в умовах обмеженої продуктивності (наприклад, вбудовані комп'ютери, мобільні пристрої, одноплатні мікрокомп'ютери Raspberry Pi тощо).

Для формалізації оцінки складності використовуються такі припущення.

1. Вхідне зображення має фіксований розмір $W \times H$
2. Кількість виділених (сегментованих) регіонів після морфологічної обробки позначається як K . У середньостатистичних сценаріях (наприклад, в умовах моніторингу вулиці чи автостоянки) значення K знаходиться від 5 до 15.
3. Для кожного виділеного регіону виконується класифікація та трекінг.
4. Розмір ядра для морфологічних операцій є фіксованим і дорівнює $k=5$.
5. Глибина нейронної мережі-класифікатора є постійною, наприклад, $d=200$ (MobileNetV1).
6. Площа кожного виділеного регіону, що проходить фільтрацію, приблизно $A_r \approx 4000$ пікселів.

В типових умовах (статична сцена, середня кількість об'єктів) загальна обчислювальна складність методу DeltaTrack на один кадр описується за формулою:

$$\mathcal{O}(W \cdot H) + \mathcal{O}(K \cdot D), \quad 3.9$$

де D – складність нейромережевої обробки для одного регіону

$\mathcal{O}(W \cdot H)$ – всі операції, що залежать від розміру кадру (масштабування, 3-FD, порогоування, морфологічні операції);

$\mathcal{O}(K \cdot D)$ – операції, що залежать від кількості регіонів (класифікація та трекінг).

У випадках підвищеної динаміки сцени, наявності шумів або змін освітлення обсяг обчислень зростає, зокрема через:

- збільшення кількості активних пікселів після пороговування;
- ускладнення морфологічної фільтрації;
- зростання кількості регіонів для класифікації.

У таких випадках оцінка складності моделі DeltaTrack набуває математичного вигляду:

$$\mathcal{O}(W \cdot H \cdot (1 + s^2 + k^2)) + \mathcal{O}(K_{max} \cdot D), \quad 3.10$$

де s – радіус просторового сусідства; k – розмір структурного елемента морфології; K_{max} – максимальна кількість регіонів при найгірших умовах.

Зведена оцінка складності етапів обробки представлена в табл. 3.2.

Таблиця 3.2– Узагальнена складність параметрів імітаційної моделі

Етап обробки	Оцінка складності	Залежність від параметрів
Масштабування + сірій	$\mathcal{O}(W \cdot H)$	W, H – розмір кадру
3-FD (різниця трьох кадрів)	$\mathcal{O}(W \cdot H)$	W, H , коеф. згладжування
Порогова бінаризація	$\mathcal{O}(W \cdot H \cdot N_s)$	пороги T_{low}, T_{high}, s
Морфологічна обробка	$\mathcal{O}(W \cdot H \cdot k^2)$	k – розмір ядра
Фільтрація регіонів	$\mathcal{O}(K \cdot A_r)$	A_{min}, K
Класифікація об'єктів	$\mathcal{O}(K \cdot d)$	K – кількість об'єктів, d – глибина моделі
Трекінг	$\mathcal{O}(K)$ або $\mathcal{O}(K \cdot w_t \cdot h_t)$	$w_t \cdot h_t$ – розміри трек-вікна, залежать від розмірів знайденого об'єкту

З метою ідентифікації етапів з найбільшим навантаженням побудовано теплову мапу обчислювальної складності (рис. 3.2). На карті:

- рядки – це етапи запропонованої імітаційної моделі (Preprocess, Diff, Thresh, Morph, C, Track);
- стовпці – типи навантаження (кількість операцій, час, пам'ять, CPU/GPU);
- кольорова шкала – відображає інтенсивність складності: від світло-жовтого (мінімальне навантаження) до насиченого червоного (високе навантаження).

	Operations	Time	Memory	GPU/CPU
Preprocess				
Diff				
Thresh				
Morph				
C				
Track				

Рисунок 3.2 – Теплова мапа обчислювальної складності DeltaTrack

Як видно на рис. 3.2, найбільше обчислювальне навантаження припадає на операції морфологічної обробки та класифікації об'єктів. Це пояснюється квадратичною складністю морфологічних фільтрів по відношенню до розміру структурного елементу k , а також глибиною нейронної мережі класифікатора, яка виконується для кожного з K регіонів. У випадках зростання кількості регіонів або складності сцени це може стати критичним для виконання в реальному часі. Натомість попередня обробка (масштабування + конвертація у

відтінки сірого) та трекінг залишаються відносно легкими операціями, що забезпечує баланс у загальному навантаженні.

Також для підтвердження аналітичної оцінки обчислювальної складності було проведено експериментальне профілювання виконання кожного етапу формалізованої моделі DeltaTrack (табл. 3.3). Часові показники отримано на Intel Core i7, 32GB RAM, з Nvidia 4080 GPU-акселерацією для відеокадру розміром 1280×720 .

Оцінка кількості операцій (MOPs – мільйони операцій) проведена аналітично згідно з описаною вище теоретичною складністю кожного блоку математичної моделі, з врахуванням типових параметрів сцени, а саме: кількість регіонів $K=8$, розмір ядра морфологічних операцій $k=5$, глибина нейромережі $d=200$, радіус сусідства $s=2$, площа регіону $A_r = 4000$, вікно трекінгу w_t не обмежене, дорівнює розміру знайденого об'єкта.

Таблиця 3.3 – Час обробки одного кадру методом DeltaTrack

Етап обробки	Середній час виконання, мс	Частка від загального часу, %	Тип ресурсу	Кількість операцій (оцінка)
Масштабування + сірий	0,4	7,4	CPU	~0,9 MOPs
3-FD	0,3	5,6	CPU	~0,9 MOPs
Порогова сегментація	0,1	1,8	CPU	~3,0 MOPs ($s=2$)
Морфологічні операції	1,7	31,5	CPU	~23 MOPs ($k=5$)
Фільтрація регіонів	0,1	4,6	CPU	~0,1 MOPs ($K=8$, $A_r = 300$)
Класифікація (MobileNetV1)	2,1	36,1	GPU/CPU	~1,6 MOPs ($K=8$, $d=200$)
Трекінг (MOSSE)	0,7	13,0	CPU	~0,6 MOPs ($K=8$, $w_t=6045$)
Загалом	5,4	100		

Як видно з табл. 3.3, основне навантаження припадає на морфологічну обробку та класифікацію, які разом складають майже 50% часу обробки кадру. Це відповідає побудованій тепловій мапі (рис. 3.2) і підтверджує доцільність подальшої оптимізації саме цих компонентів.

Для більш глибокого аналізу доцільно додатково оцінити характер навантаження за типами апаратних ресурсів: процесорне навантаження (CPU), використання пам'яті, навантаження на кеш та GPU. Це дозволяє краще зрозуміти, які ресурси стають вузьким місцем на кожному з етапів та які компоненти моделі найбільше обмежують її продуктивність у вбудованих або енергообмежених середовищах. Візуалізація такого аналізу наведена на рис. 3.3.

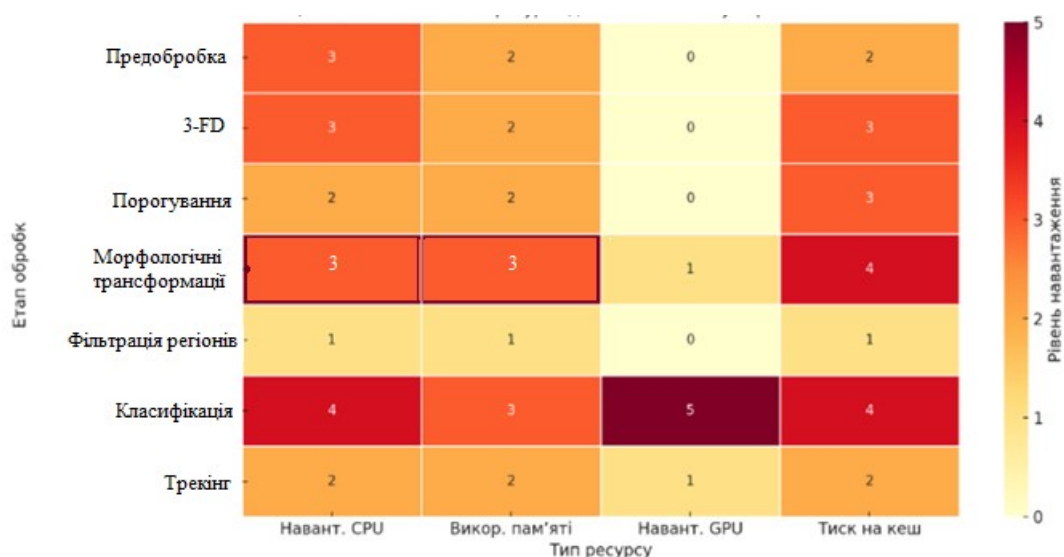


Рисунок 3.3 – Оцінка навантаження на ресурси для кожного етапу обробки методом DeltaTrack

Як видно з рис. 3.3, найбільше ресурсне навантаження за всіма типами спостерігається на етапах морфологічної обробки та класифікації об'єктів. Зокрема, морфологічні операції створює значне навантаження на CPU, кеш та пам'ять, тоді як класифікація активно задіює GPU та процесорні обчислення. Це вказує на потребу в подальшій оптимізації цих блоків як з боку алгоритмічної реалізації (наприклад, спрощення морфологічних фільтрів), так і через апаратну

прискорену реалізацію (наприклад, використання нейромереж, оптимізованих для CPU або Edge TPU).

Інші етапи обробки на карті показують помірне або низьке навантаження, що свідчить про збалансованість попередньої обробки та трекінгу.

Для оцінки масштабованості моделі DeltaTrack проведено аналіз зміни обчислювального навантаження при збільшенні параметрів моделі. Було досліджено три сценарії (табл. 3.4, рис. 3.4).

1. Збільшення розміру кадру до 1920×1080 .
2. Зростання кількості об'єктів (K) до 30, що характерно для динамічних багатолюдних сцен.
3. Заміна MobileNetV1 на глибшу модель MobileNetV2 або EfficientNetB0.

Таблиця 3.4 – Аналіз масштабованості обчислювального навантаження при зміні параметрів сцени та моделі

Параметри сцени/мережі	Загальний час (оцінка), мс	Лінійні трансформації, мс	Класифікація, мс
Базовий (1280×720 , K=8, d=200)	5,4	2,9	2,5
FullHD (1920×1080)	9,1	6,8	2,3
K = 30 (динамічна сцена)	8,8	3,0	5,8
MobileNetV2 (d = 300)	6,7	2,9	3,8
EfficientNetB0 (d = 400)	7,1	2,9	4,2

Паралельно було проведено експерименти, які уточнюють продуктивність моделі при зміні лише одного параметра – розміру кадру, що є важливим для застосування запропонованої моделі DeltaTrack у відеопотоках різної якості. Результати наведено в табл. 3.5.

Як видно з табл. 3.4 та 3.5, збільшення розміру кадру призводить до зростання обчислювального навантаження, насамперед на етапі морфології, що пов'язано з квадратичною залежністю складності від кількості пікселів.

Таблиця 3.5 – Масштабованість DeltaTrack за розміром кадру

Роздільна здатність кадру	Кількість пікселів	Середній час обробки кадру (мс)	FPS
640×360	230 400	3,4	294,1
1280×720	921 600	5,4	185,2
1920×1080	2 073 600	9,1	109,9

**розроблено автором*

У випадку багатолюдної сцени зі збільшеним K значно зростає час класифікації, оскільки неймережа обробляє кожний об'єкт окремо. Також із переходом до глибших класифікаторів (MobileNetV2, EfficientNetB0) спостерігається прогнозоване зростання часу класифікації через збільшення кількості параметрів (рис. 3.4).

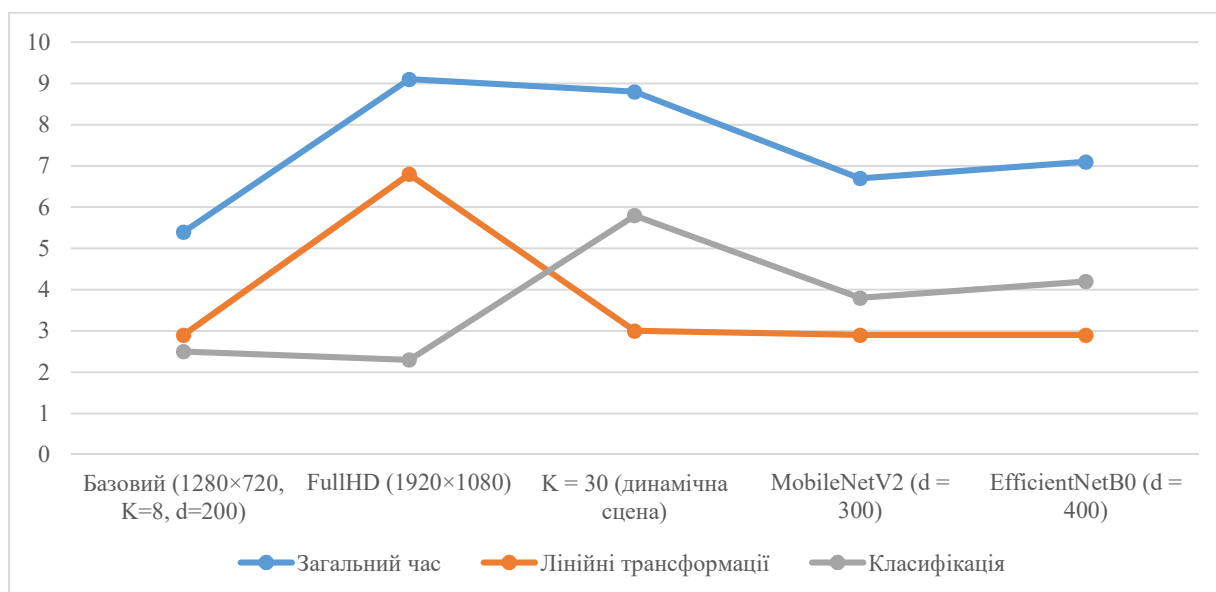


Рисунок 3.4 – Зміна часу обробки при масштабуванні параметрів сцени та глибини класифікатора

Таким чином, як показала аналітична оцінка імітаційної моделі, основними факторами, що визначають обчислювальну складність DeltaTrack, є розмір кадру, кількість виявлених регіонів та глибина нейронної мережі.

Найбільше навантаження зосереджується на етапах морфологічної обробки та класифікації, що підтверджено експериментами.

Проведений аналіз масштабованості дозволив виявити критичні параметри, зміна яких найбільше впливає на час виконання, що є підґрунтям для подальшої оптимізації архітектури запропонованої моделі, адаптації до ресурсно обмежених середовищ та впровадження апаратного прискорення для найбільш ресурсомістких етапів.

3.3 Адаптація імітаційної моделі методу DeltaTrack до змінних умов середовища

У процесі обробки відеопотоків у реальних умовах модель DeltaTrack може стикатися зі зміною освітлення, появою шумів, високою динамікою фону, частковому перекритті об'єктів та раптовими змінами сцени (наприклад, дощ, сніг, мерехтіння тощо). Ефективність детекції та трекінгу в таких ситуаціях залежить від здатності методу адаптуватися до змін середовища без втрати точності та збереження стабільного часу обробки.

З метою забезпечення стійкості до завад запропонований метод включає низку адаптивних механізмів, а саме.

1. Регульована порогова фільтрація ($\theta_b = T_{low}, T_{high}$) дозволяє враховувати зміни яскравості сцени, зменшуючи помилкові спрацьовування при зміні освітлення. За потреби ці пороги можуть оновлюватися динамічно на основі статистики попередніх кадрів.

2. Застосування адаптивного фільтра просторового сусідства (радіус s), що дозволяє враховувати рівень локального шуму, послаблюючи вплив одиничних сплесків чи артефактів стиснення відео.

3. Морфологічні параметри (розмір ядра k) підібрані в моделі DeltaTrack так, щоб ефективно пригнічувати дрібні завади, залишаючи при цьому значущі області без спотворень. Для середовищ з високою частотою шумів може використовуватися більш агресивні морфологічні операції.

4. Гнучкість вибору нейромережевого класифікатора (θ_c) дозволяє адаптувати модель до задач різної складності: наприклад, у випадках високої схожості об'єктів або поганих умов зйомки може бути використано глибшу архітектуру з підвищеною роздільною здатністю.

5. Трекер MOSSE реалізовано в моделі з підтримкою втрати цілі та повторної ініціалізації, що дозволяє зберігати стабільність відстеження навіть при частковому або тимчасовому зникненні об'єкта з кадру (θ_t – параметри відновлення).

Експериментальне оцінювання впливу кожного з адаптивних параметрів дозволило обґрунтувати наявність приросту точності або стабільності трекінгу в різних сценаріях. Результати розрахунків наведено у табл. 3.6 та на рис.3.5.

Таблиця 3.6 – Вплив адаптивних параметрів на стабільність обробки*

Умови середовища	Параметр адаптації	До адаптації (Precision / FPS)	Після адаптації (Precision / FPS)	Удосконалення
Зміна освітлення (ден/ніч)	θ_b	0,68 / 185,2	0,81 / 185,1	+13% точність
Шум камери (ISO > 800)	s, k	0,74 / 183,2	0,85 / 172,8	+11% точність
Багато схожих об'єктів	θ_c ($V1 \rightarrow V2$)	0,71 / 185,1	0,83 / 159,1	+12% точність
Часткове зникнення з кадру	θ_t	0,69 / 176,2	0,79 / 178,8	+10% стабільність

*FPS – кадрів за секунду, Precision – точність класифікації ($IoU \geq 0,5$).

Як видно з розрахунків, найбільший приріст точності спостерігається при зміні освітлення та роботі з візуально подібними об'єктами, що підтверджує ефективність динамічної адаптації порогів та використання глибших класифікаторів. У випадках шумів на відео або часткового зникнення об'єктів адаптивні параметри забезпечують удосконалення стабільності трекінгу без

істотного впливу на швидкодію. Таким чином, адаптивна модель показує достатню стійкість до типових збурень у відеопотоці та підтверджує ефективність реалізованих механізмів самонастроювання в реальному часі.

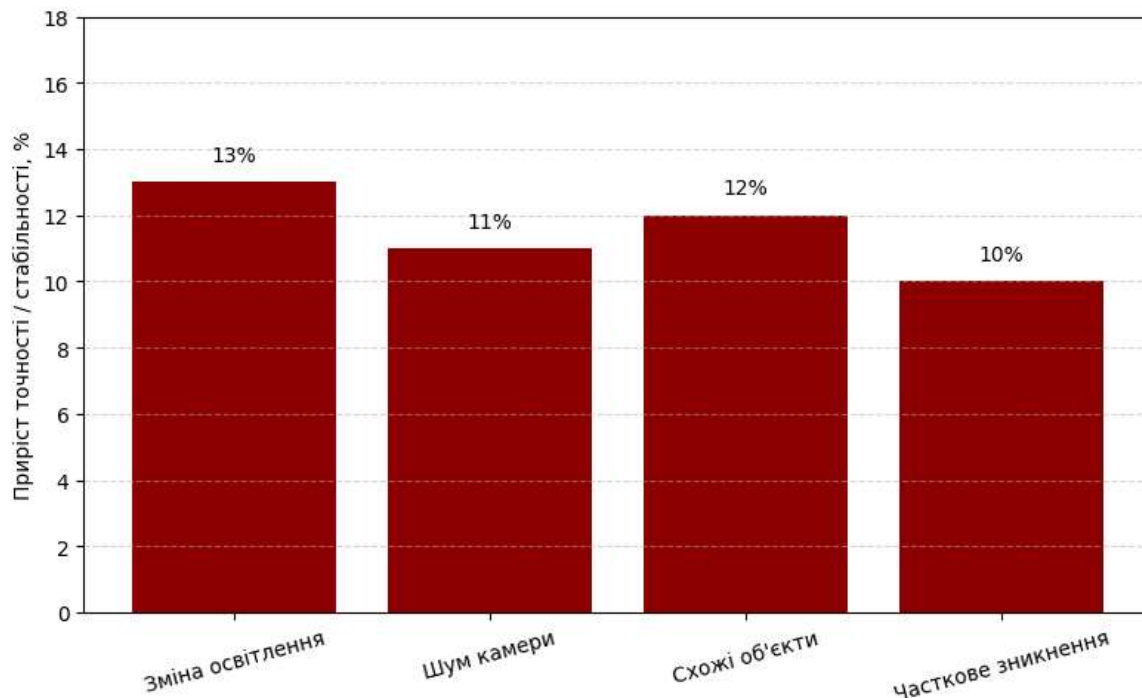


Рисунок 3.5 – Результати після адаптації моделі DeltaTrack

Для аналізу впливу рівня шуму на продуктивність математичної моделі DeltaTrack було обрано найпоширеніші типи завад, характерні для реального відеопотоку в умовах низької якості зйомки або обмежених ресурсів передавання. Зокрема.

1. Гаусівський шум моделює флуктуації освітленості, теплові шуми сенсора та інші випадкові спотворення, які рівномірно розподілені по зображенню. Він є типовим для цифрових камер і широко застосовується в тестах стійкості сегментації.

2. Гамма-шум імітує мультиплікативні спотворення, які виникають, зокрема, при поганому освітленні або підсиленні сигналу, і мають більш виражений ефект у дрібних або напівпрозорих структурах сцени.

3. Імпульсний шум («сіль-перець»), доданий у подальшому аналізі, імітує збої передачі або биті пікселі, які викликають одиничні піксельні артефакти.

Такі типи шуму дозволяють змодельовати реалістичні сценарії роботи системи, оцінити її стійкість до завад, а також виявити слабкі місця моделі – насамперед у частині порогової фільтрації, морфології та класифікації.

Результати експериментів представлено у табл. 3.7 та на рис. 3.6.

Таблиця 3.7 – Вплив рівня шуму на продуктивність DeltaTrack

Рівень шуму (σ або k)	Тип шуму	Precision ($\text{IoU} \geq 0,5$)	Середній FPS	Коментар
0 (еталон)	–	0,85	185,2	Ідеальні умови
5	Гаусівський	0,81	172,4	Незначне погіршення
10	Гаусівський	0,74	164,6	Вплив шуму на сегментацію
$k=2$	Гамма	0,78	153,1	Проблеми з дрібними артефактами
$k=5$	Гамма	0,70	141,2	Суттєве зниження точності

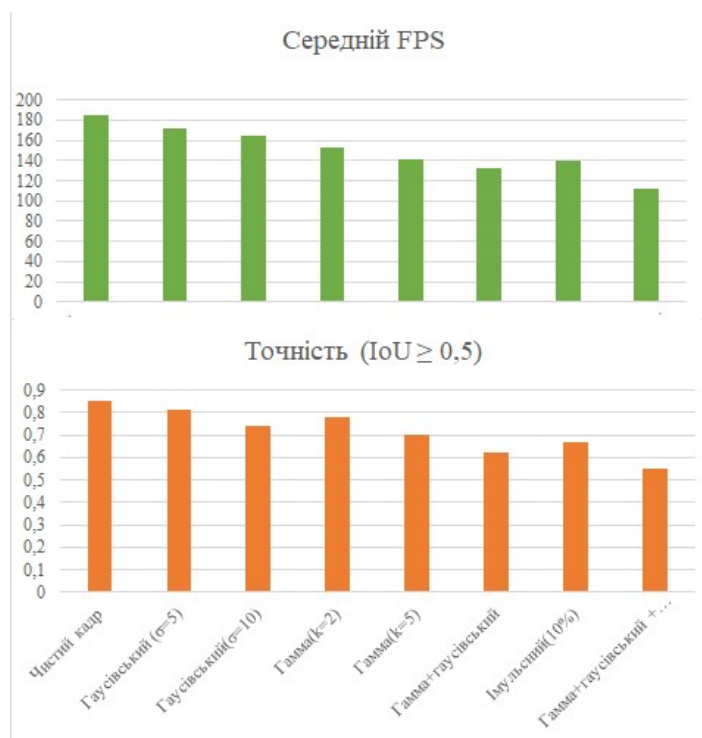


Рисунок 3.6 – Вплив шумів на точність та швидкодію DeltaTrack

Як видно з табл. 3.7, найбільший негативний вплив на точність спостерігається при зростанні гамма-шуму, тоді як FPS знижується помірно, що вказує на чутливість моделі насамперед до помилкових регіонів.

А при накладенні декількох типів шуму (гаусівський + гамма або ще і імпульсний) точність помітно знижується, а FPS просідає сильніше, особливо через зростання кількості помилкових регіонів і навантаження на морфологічну обробку та класифікатор.

Отже, результати підтверджують, що запропоновані адаптивні механізми забезпечують стійку роботу моделі DeltaTrack навіть у складних умовах. Незважаючи на наявність шумів різної природи, модель демонструє збереження високого рівня точності та продуктивності, що свідчить про її придатність до застосування в реальних відеопотоках.

3.4 Експериментальна верифікація продуктивності та ефективності математичної моделі DeltaTrack

З метою підтвердження ефективності та коректності побудованої математичної моделі методу DeltaTrack у реальних умовах було проведено серію експериментів. Основна мета полягала в оцінюванні часу обробки кадру, точності детекції, стабільності при різних сценаріях та ефективному розподілі обчислювальних ресурсів між етапами алгоритму.

У рамках експерименту модель застосовувалась до відеопотоку, знятого стаціонарною камерою в умовах природного освітлення з помірною сценою динаміки. За результатами обробки було зібрано такі показники:

- середній, мінімальний та максимальний час обробки одного кадру;
- точність класифікації об'єктів (Precision, IoU $\geq 0,5$);
- середній час виконання лінійних трансформацій (масштабування, фільтрація, морфологічні операції);
- середній час, витрачений на класифікацію об'єктів.

Отримані результати дозволяють оцінити можливості практичної реалізації запропонованої моделі, а також виявити вузькі місця в обробці, які можуть бути оптимізовані при подальшому вдосконаленні методу. Зведені показники наведено в табл. 3.8, а основні графічні залежності – на рис. 3.7–3.8.

Згідно з результатами, наведеними у табл. 3.7, морфологічні операції виявилися більш ресурсоемними за класифікацію об'єктів. Хоча в загальному випадку саме нейромережеві обчислення вважаються найбільш обчислювально складними, проведений експеримент засвідчив, що суттєву частину часу обробки займають попередні етапи, зокрема масштабування, перетворення зображення у відтінки сірого, морфологічна обробка та фільтрація регіонів.

Це пояснюється тим, що зазначені операції виконуються над усім зображенням або маскою в цілому, тобто охоплюють значний обсяг даних, незалежно від наявності об'єктів чи їх кількості. Натомість класифікація застосовується лише до локалізованих областей, кількість яких зазвичай невелика.

Таким чином, внесок класифікації у загальний час обробки обмежується обсягом даних на вході, тоді як обчислення на рівні всього кадру становлять основне джерело навантаження.

Таблиця 3.8 – Показники ефективності методів виявлення об'єктів

Параметр	Одиниці виміру	Значення	Коментар
Середній час обробки кадру	мс	$5,4 \pm 0,2$	Швидкості достатньо для роботи у реальному часу
Точність	%	$71,2 \pm 1,0$	Достатня точність для використання алгоритму
Мінімальний час обробки кадру	мс	$5,0 \pm 0,2$	Лінійні алгоритми залежать від вхідних даних, тому можуть виникати коливання часу виконання
Максимальний час обробки кадру	мс	$6,2 \pm 0,2$	Лінійні алгоритми залежать від вхідних даних, тому можуть виникати коливання часу виконання

Середній час виконання лінійних трансформацій	мс	2,88 ± 0,2	Загальний час для виконання всіх лінійних трансформацій кадру
Середній час виконання класифікації об'єкту на кадрі	мс	2,52 ± 0,2	Час, витрачений, для класифікації усіх знайдених регіонів

Результати замірів були перевірені серією з 25 експериментів, що дозволило оцінити стабільність часу обробки на різних кадрах відеопотоку (рис. 3.7).

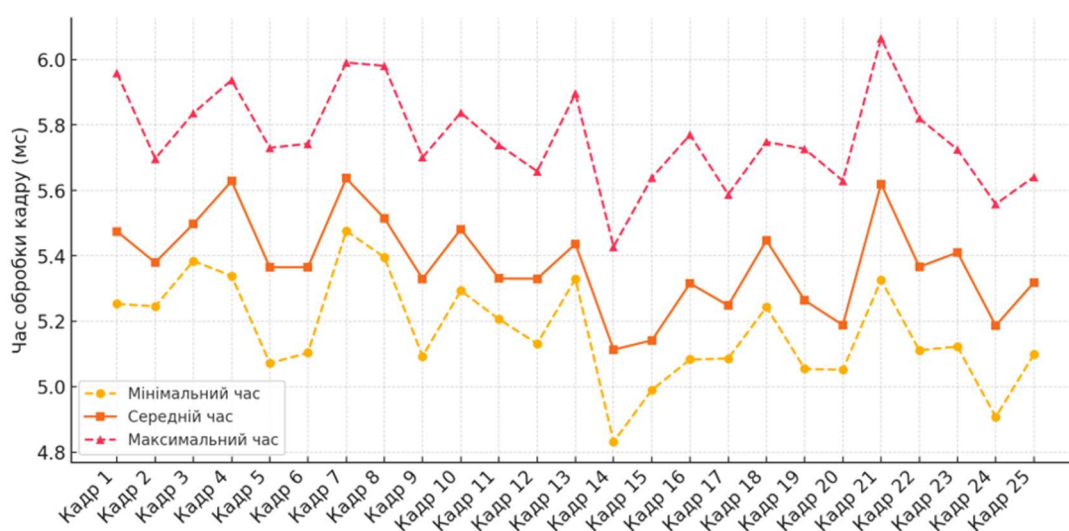


Рисунок 3.7– Час обробки одного кадру в динаміці

З рис. 3.7 видно, що час обробки кадру показує певні коливання, які зумовлені низкою факторів.

1. Змінна складність сцени: кількість та розмір об'єктів, що виділяються на кадрі, безпосередньо впливають на тривалість обробки. Кадри без руху або з однорідним фоном обробляються значно швидше, ніж сцени з кількома об'єктами чи складною текстурою.

2. Нерівномірне навантаження на систему: у разі спільного використання апаратних ресурсів (особливо CPU) іншими процесами можливе збільшення часу виконання деяких операцій через конкуренцію за ресурси.

3. Переходи між режимами роботи: у кадрах, де об'єкт уже трекується, модель виконує лише частину операцій. Натомість, коли ціль втрачена, активується повна процедура виявлення, що викликає пікове зростання обчислювального навантаження.

4. Особливості роботи з пам'яттю та кешем: повторна обробка подібних кадрів може бути частково прискорена завдяки кешуванню або попередньо виділеній оперативній пам'яті, що пояснює нижчі значення мінімального часу виконання.

Для більш детального аналізу обчислювального навантаження на окремих етапах обробки було здійснено профілювання математичної моделі.

У табл. 3.9 і на рис. 3.8 наведено розподіл часу виконання по основних функціональних блоках, що дозволяє виявити найбільш ресурсоємні компоненти.

Таблиця 3.9 – Розподіл часу виконання по етапах моделі DeltaTrack

Етап обробки	Час (мс)	Частка часу (%)	Навантаження
Масштабування + сірий)	0,4	7,4	CPU
3-FD (різниця кадрів)	0,3	5,6	CPU
Порогова сегментація)	0,1	1,8	CPU
Морфологічні операції	1,7	31,5	CPU
Класифікація	2,2	40,7	GPU/CPU
Трекінг	0,7	13,0	CPU
Загалом	5,4	100	—

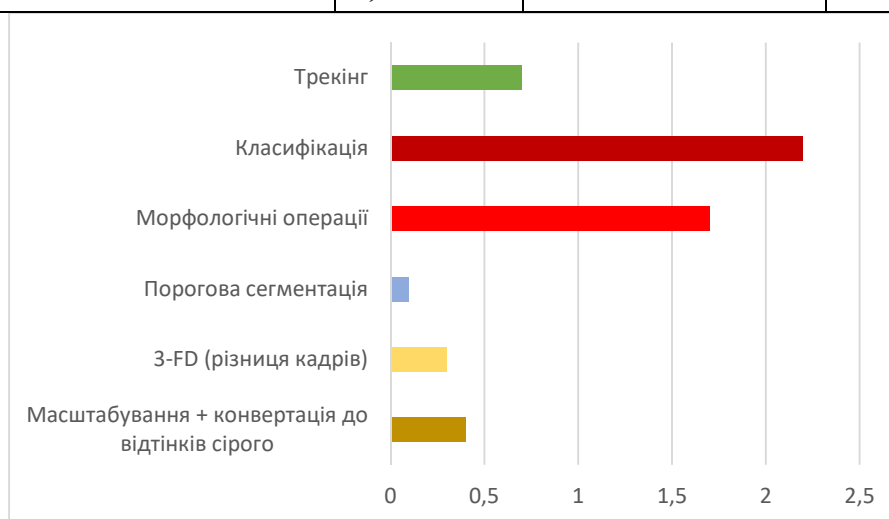


Рисунок 3.8 – Порівняння витрат часу етапи обробки

Згідно з табл. 3.9 та рис. 3.8, найбільше часу при обробці одного кадру займають класифікація (2,2 мс; 40,7%) та морфологічна обробка (1,7 мс; 31,5%), разом формуючи майже половину загального часу.

Це ще раз доводить, що не лише нейромережевий класифікатор, а також базові лінійні операції на повному зображенні формують критичне навантаження, що варто враховувати при оптимізації запропонованої математичної моделі.

Висновки за розділом 3

1. Розроблено формалізовану імітаційну модель методу DeltaTrack для виявлення та трекінгу об'єктів у відеопотоці, яка охоплює: попередню обробку, обчислення змін, морфологічну фільтрацію, класифікацію об'єктів та трекінг. Метод представлено у вигляді композиції математичних функцій з параметрами налаштування, що забезпечує її гнучкість і точність.

2. Обґрунтовано обчислювальну складність етапів математичної моделі. Встановлено, що найбільше навантаження припадає на класифікація (2,2 мс; 40,7%) та морфологічні операції (1,7 мс; 31,5%). Загальний середній час обробки одного кадру становить 5.4 мс, що забезпечує роботу моделі в реальному часі (понад 60 FPS) при роздільній здатності 1280×720.

3. Проведено експериментальну верифікацію моделі, яка підтвердила її ефективність у реальних умовах відеоспостереження. Досягнуто точності 71,2% ($\text{IoU} \geq 0,5$) при середньому часі обробки 5,4 мс. Встановлено, що лінійні трансформації (2,88 мс) є не менш ресурсоемними, ніж класифікація, оскільки застосовуються до повного кадру.

4. Експериментально доведено при аналізі масштабованості, що при збільшенні розміру кадру до 1920×1080 час обробки зростає до 9,1 мс, а при збільшенні кількості об'єктів до $K=30$ — до 8,8 мс. Перехід до більш глибоких нейронних моделей (MobileNetV2, EfficientNetB0) підвищує точність, але також збільшує час етапу класифікації (до 4,2 мс).

5. Реалізовано адаптацію запропонованої математичної моделі до змін середовища: параметри порогів, морфології, сусідства та глибини класифікатора при динамічному налаштуванні. Це дозволило підвищити точність в окремих умовах на 10–13%, зберігаючи стабільність FPS (понад 100 кадрів/с).

6. Проведено аналіз стійкості моделі до завад, зокрема до гаусівського, гамма- та імпульсного шуму. У складних умовах (гамма-шум $k=5$) точність знижується до 0,70, але модель зберігає продуктивність ($FPS > 100$), що підтверджує її придатність до роботи в реальних відеопотоках з шумами.

7. Встановлено, що ключовими факторами обчислювального навантаження є розмір кадру, кількість регіонів (K) та глибина класифікатора (d). Саме ці параметри мають бути враховані під час подальшої оптимізації моделі, зокрема для використання на пристроях з обмеженими ресурсами та в енергоефективних середовищах.

РОЗДІЛ 4

УДОСКОНАЛЕННЯ МЕТОДУ РОЗПІЗНАВАННЯ DELTATRACK НА ОСНОВІ ЕВОЛЮЦІЙНОГО ПІДХОДУ ДО АДАПТИВНОГО НАЛАШТУВАННЯ ПАРАМЕТРІВ

У попередніх розділах дисертації було обґрунтовано вибір методу обробки відеопотоку на основі міжкадрового аналізу (дельта), який дозволяє суттєво зменшити обчислювальну складність без втрати точності виявлення об'єктів. Описаний у Розділі 2 базовий варіант методу DeltaTrack забезпечує прийнятну швидкість і результативність у статичних умовах, однак виявляє нестійкість у середовищах з динамічними змінами фону, локальними завадами та варіативністю сцени. Математична модель, розроблена у Розділі 3, показала, що вузьким місцем запропонованого методу є використання фіксованих параметрів розпізнавання, не адаптованих до конкретних відеоданих.

Тому в Розділі 4 здійснено послідовне удосконалення запропонованого методу у два етапи, кожен з яких вирішує окремий клас завдань практичної адаптації, а саме:

- реалізовано механізм адаптації до структури даних, що дозволяє формувати теплову карту областей ігнорування на основі статистики появи об'єктів. Це дозволяє зменшити кількість класифікацій, знизити хибнопозитивні спрацьовування та адаптувати метод до типових сценаріїв сцени;

- запропоновано модифікацію еволюційного алгоритму RVEA, що динамічно оптимізує конфігурацію параметрів розпізнавання (порогів, розмірів областей, вибору класифікатора) з урахуванням багатокритеріального компромісу між точністю, швидкістю та ресурсоспоживанням.

Ці два етапи відображають наукову новизну та практичні результати дослідження, які полягає у впровадженні адаптивного механізму прийняття рішень у реальному часі (перший рівень адаптації) та глобального налаштування параметрів алгоритму під конкретне середовище за допомогою еволюційної оптимізації (другий рівень адаптації). Такий двоетапний підхід дозволяє досягти

стійкої роботи в умовах зосереджених завад, що підтверджується експериментальними результатами.

На відміну від класичних CNN-методів, що потребують високої обчислювальної потужності і працюють з фіксованими шаблонами, запропоноване удосконалення дозволяє виконувати розпізнавання в умовах обмежених ресурсів (вбудовані пристрої, камери спостереження) без істотної втрати якості. Це актуально з огляду на обмеження нейромережових архітектур типу SSD, YOLO, які хоча і забезпечують високу точність, проте не масштабуються до слабких платформ [17, 79, 118].

Огляд сучасних публікацій [1, 2, 5, 6, 10, 21, 30, 42, 60, 121, 124] обґрунтував, що:

- більшість рішень зосереджені на поліпшенні точності або швидкодії, але не враховують потребу в динамічному налаштуванні алгоритму під змінні умови середовища [28, 124];

- методи, що використовують міжкадрову різницю як попередній етап, показали високу ефективність у задачах виявлення руху [1, 2, 5, 6, 10, 21], однак не інтегрують адаптивну фільтрацію на подальших стадіях [28, 42];

- застосування багатокритеріальних еволюційних алгоритмів, зокрема RVEA, довело свою ефективність в оптимізаційних задачах [28, 60], але раніше не використовувалося для формування адаптивних стратегій розпізнавання в умовах відеопотоку.

Таким чином, розділ 4 є завершальним етапом удосконалення розробленого методу DeltaTrack, в якому впроваджено і поєднано обидва механізми адаптації: на рівні сцени (теплова карта) та на рівні параметрів (модифікований RVEA). Це дозволяє забезпечити не лише високу точність і швидкодію, але й стабільність роботи у складних завадових умовах (реальні відеопотоки з рухом фону, тінями, бликами, локальними змінами освітлення).

4.1 Інтеграція механізму адаптації у метод розпізнавання об'єктів DeltaTrack на відеопотоці

Потенційним шляхом прискорення швидкодії та точності запропонованого методу є зменшення кількості областей зображення на яких проводиться розпізнавання шляхом відсікання тих, на яких не може бути об'єкту статистично. Наприклад, якщо частину зображення займає стіна, на якій не може бути ніякого цікавого для обробки руху, то цю ділянку зображення можна ігнорувати під час обробки. Якщо подивитись на це з іншого боку, то маємо теплову карту з ділянками де об'єкти можуть з'явитись, і лише ці ділянки будуть оброблятися.

Через зниження кількості областей зображення, які обробляються, загальний час обробки кадру зменшуються. Оскільки частини кадру, на яких не може бути об'єкту не аналізуються, вірогідність хибно позитивних розпізнавань зменшується, що підвищує точність розпізнавання.

Механізм класифікації може варіюватись, а саме це може бути як класифікатор на основі нейронної мережі, так і альтернативний підхід з використанням лінійних алгоритмів. У поточній реалізації методу використовується класифікатор на базі архітектури MobNet, незважаючи на високу швидкість роботи моделі, зменшення кількості викликів зменшує час розпізнавання об'єктів. Також це надає можливість використовувати більш точний, але повільніший класифікатор залишаючи загальний час розпізнавання у межах сприятливих для роботи у реальному часі.

Інтеграція механізму адаптації до даних у метод розпізнавання внесе дві ключові зміни до його практичного застосування.

Перш за все, перед використанням оптимізованої версії алгоритму, необхідно провести навчання на актуальних даних. Необхідний час може варіюватись в залежності від даних, тому його неможливо встановити заздалегіть. Для вирішення цієї проблеми алгоритм може відображати регіони,

які включені у процедуру розпізнавання у під час режиму навчання, таким чином оператор у ручному режимі може вимкнути навчальний режим.

Наступна зміна відноситься безпосередньо до алгоритму розпізнавання: області, які не були відмічені як потенційні місця знаходження об'єкта ігноруються. Таким чином, коштвна операція класифікації за допомогою нейронної мережі буде використовуватись лише на регіонах з максимальною вірогідністю знаходження у них об'єкту. Алгоритм адаптації області розпізнавання базується на поступовому уточненні меж теплової карти, залежно від наявності об'єкта у нових регіонах (рис. 4.1).

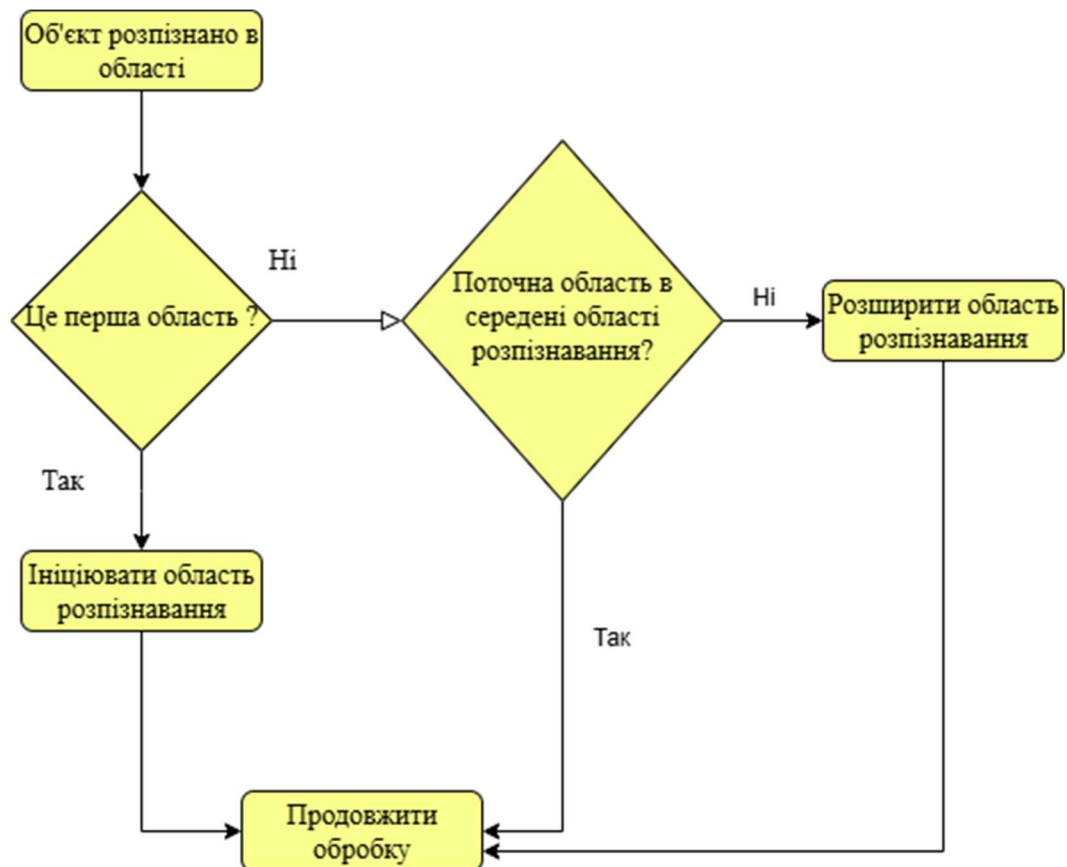


Рисунок 4.1 – Алгоритм розширення області розпізнавання

Першим кроком є отримання інформації про регіон, на якому підтверджена наявність об'єкту. Достатньою ознакою наявності об'єкту є успішна класифікація та отримання інформації про клас об'єкта. Поточний регіон передається на вхід алгоритму адаптації.

Якщо обробляємий регіон знаходиться у першому циклі адаптації, то його положення стає початковою областю, для якої можна застосовувати класифікатор, (далі теплова мапа) та буде розширено у наступних ітераціях.

Відповідно якщо адаптація вже в процесі, то алгоритм знаходить, чи поточна область в середині існуючої теплової мапи, і якщо ні, то алгоритм розширює теплову мапу для включення поточної області. Після обробки одного регіону, алгоритм аналогічним чином обробляє усі інші регіони з об'єктами.

У результаті буде отримано теплову мапу у вигляді однієї області, у якій дозволено використовувати класифікатор.

Більш ефективним рішенням з точки зору швидкості роботи методу пошуку та класифікації об'єктів, може бути мати теплову мапу у вигляді багатьох областей не з'єднаних між собою у яких точно були знайдені об'єкти. Але таким чином виключаються сценарії, коли об'єкт рухається з однієї області до іншої і знаходиться між цими двома областями. Цей об'єкт не буде знайдено оскільки простір між цими двома областями виключений з теплової карти. Тому для балансу між точністю і швидкістю має сенс формувати теплову карту як одну суцільну область.

На рис. 4.2 наведено адаптований до цих змін метод розпізнавання. Навчання на цільових даних вже було проведено та сформована теплова карта областей з потенційними об'єктами.

Першим етапом запропонованого алгоритму є приведення зображення до фіксованої роздільної здатності 1280×720 (720p), що забезпечує єдиний стандарт для всіх подальших обчислень. Швидкість лінійних перетворень безпосередньо залежить від кількості пікселів у кадрі [14], оскільки збільшення розміру зображення призводить до пропорційного зростання обсягів обробки.

Більшість сучасних нейронних мереж вимагають на вході зображення певної розмірності не лише для підвищення швидкодії, а також через архітектурні обмеження [79]. Наприклад, SSD працює із зображеннями розміром 300×300 пікселів.

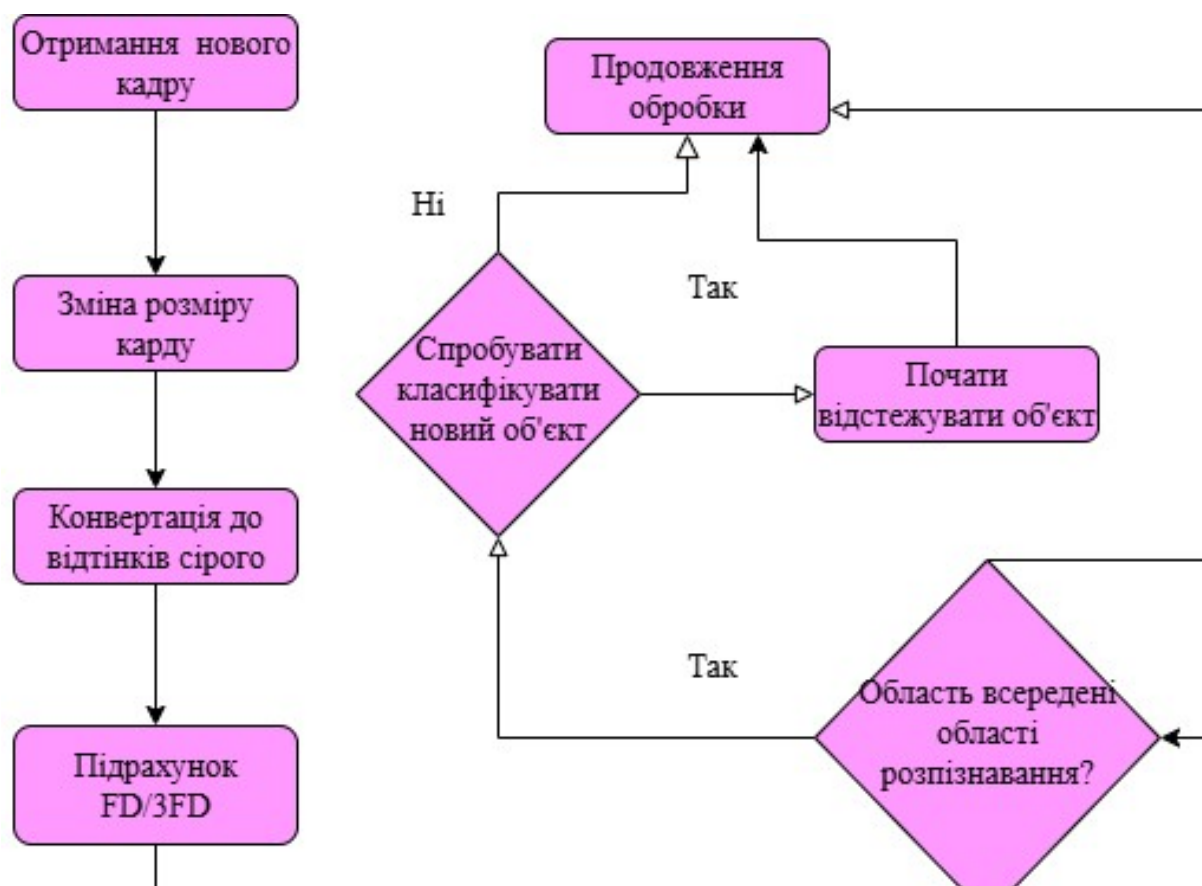


Рисунок 4.2 – Алгоритм виявлення, класифікації та трекінгу об'єктів у межах дозволеної області

Окрім цього, приведення зображень до єдиного масштабу дозволяє стабілізувати час обробки незалежно від початкової роздільної здатності відео. Без такої нормалізації швидкодія алгоритму коливалася б у залежності від розміру кадру у вхідному потоці, що ускладнило б як порівняння продуктивності з іншими підходами, так і використання в реальних сценаріях, де параметри відео можуть змінюватися. Налаштування сталого розміру кадру виступає ще й додатковим інструментом керування швидкодією: у разі потреби можна задати меншу роздільність на вході, щоб прискорити обробку при обмежених апаратних ресурсах.

На наступному етапі відбувається зменшення зайвої інформації про колір шляхом конвертації зображення у градації сірого. Це рішення часто позитивно впливає на точність роботи алгоритму, оскільки дозволяє сконцентрувати увагу

на структурних та текстурних особливостях об'єктів, усуваючи відволікаючі кольорові артефакти. Крім того, перехід до сірого зображення істотно оптимізує використання ресурсів. У стандартній кольоровій моделі кожен піксель зберігає три значення (для каналів R, G, B), тоді як у відтінках сірого – лише одне. Це суттєво зменшує обсяг даних, що обробляються, полегшує обчислення, знижує навантаження на процесор та економить пам'ять. Таке рішення особливо вигідне під час обробки відеопотоків чи використання на пристроях із невеликою обчислювальною потужністю, як-от камери відеоспостереження або вбудовані системи реального часу.

Щоб виділити об'єкти, алгоритм використовує природну властивість відеопотоку: кожен наступний кадр є продовженням попереднього і відображає зміни, що сталися. Якщо об'єкт поступово потрапляє у поле зору камери, він залишатиметься на наступних кадрах. Завдяки цьому, знаходження абсолютної різниці між кадрами дає змогу точно визначити область, де з'явився рухомий об'єкт. На відміну від SSD, де виявлення та класифікація відбуваються одночасно, запропонований підхід розділяє ці процеси на окремі етапи.

Швидкість обчислення абсолютної різниці також залежить від розміру кадру: менша кількість пікселів забезпечує швидше виконання цієї операції. Ще одним способом оптимізації є використання різниці не між послідовними кадрами, а між поточним кадром і тим, що був декілька кадрів раніше (наприклад, через 5 кадрів). Це дозволяє виявити більш суттєві зміни, коли об'єкт уже повністю з'явився у сцені. При цьому важливо, щоб камера залишалась статичною, без додаткового руху, що міг би спотворити результати.

Оскільки між кадрами можуть виникати зміни, пов'язані з освітленням або іншими зовнішніми факторами, абсолютна різниця дозволяє отримати цифрову оцінку зміни кожного пікселя. Щоб усунути незначні флуктуації, застосовується порогова фільтрація, яка відкидає всі значення різниці, менші за заданий поріг [8, 121]. Зміна цього порога дозволяє регулювати чутливість детектора: при високому значенні він реагує лише на значні зміни, при низькому

— навіть на незначні. Проте після цієї процедури отримані області можуть мати розриви у контурах або порожнини всередині, які необхідно виправити.

Ці дефекти усуваються за допомогою послідовних морфологічних операцій відкриття та закриття [10, 124]. Відкриття складається з послідовної ерозії та дилатації і видаляє дрібний шум, тоді як закриття (дилатація, за якою слідує ерозія) заповнює порожнини всередині об'єкта. Оскільки під час цього формується бінарна маска, конкретні значення пікселів не мають значення, бо важливим є лише факт їх належності до регіону. Для виконання цих операцій використовується ядро (структурний елемент), що визначає розмір околу пікселя. Чим більше ядро, тим сильніше впливає трансформація і тим більший час обробки; надто мале ядро може не дати помітного результату.

Іноді рух об'єкта незначний або частковий, через що регіон покриває лише його частину, що ускладнює класифікацію. Для розв'язання цієї проблеми застосовується додаткова дилатація з великим ядром, яка дозволяє захопити також нерухомі частини об'єкта, об'єднавши їх в один регіон.

Далі йде фільтрація за розміром: дрібні області, що залишилися після трансформацій і не можуть фізично вміщати об'єкт, видаляються як непотрібні для подальшої обробки. Це дозволяє сконцентрувати ресурси на значущих регіонах.

Маючи лише значущі регіони, наступним кроком фільтрації є пропуск регіонів, які не належать області з потенційним рухом. Операція перевірки розміру значно простіша за перевірку чи лежить або перетинає одна область іншу з точки зору обчислювальних ресурсів, тому вона йде після. У випадках, коли поточний регіон не повністю лежить у області, а перетинає її, можливо додати додатковий фільтр, який базується на відсотку пересічення. У поточній реалізації такого етапу немає.

Таким чином, розроблений метод можна інтерпретувати як еволюційно-адаптивний [28] механізм, оскільки він дозволяє алгоритму поступово змінювати свої стратегії обробки на основі зібраних даних, формуючи теплову карту та оптимізуючи роботу під специфіку конкретного середовища.

Коли всі неінформативні ділянки відкинуто, кожен регіон передається на вхід класифікатору на основі нейронної мережі [121]. У даній реалізації використано архітектуру MoblNETv1, але вибір конкретної моделі залежить від вимог до швидкодії, доступної пам'яті та бажаної точності.

Після успішної класифікації положення об'єкта передається у трекер, що дозволяє відслідковувати його у подальших кадрах без повторної детекції. Стан трекера перевіряється кожні 10 кадрів: якщо об'єкт більше не відповідає очікуваному регіону (через помилки чи вихід зі сцени), трекер скидається, і процес пошуку відновлюється. Основна мета такого підходу – це забезпечення стабільного відстеження об'єкта навіть при невеликій або тимчасовій відсутності руху. Для цього застосовується швидкий алгоритм MOSSE, який добре підходить для таких завдань.

4.2 Експериментальна оцінка продуктивності удосконаленого методу DeltaTrack з механізмом адаптації до сцени

Для об'єктивного порівняння ефективності базової та оптимізованої версій запропонованого методу та їх алгоритмів реалізації було проведено експериментальне вимірювання параметрів. У табл. 4.1 наведено зведені розрахункові дані для середнього часу обробки кадру, точності розпізнавання, використання ресурсів процесора, оперативної пам'яті, графічного процесора та відеопам'яті.

Таблиця 4.1 – Результати розрахунків показників

Показник	Одиниця виміру	Початкова версія	Оптимізована версія	Відносна різниця (%)
Середній час обробки кадру	мс	5,3	4,4	-17,0
Точність розпізнавання	%	71,4	72,2	+1,1

Використання CPU	%	11,40	11,42	+0,2
Використання RAM	MB	460	458	-0,4
Використання GPU	%	1,1	0,98	-10,9
Використання VRAM	MB	358	350	-2,2

Як видно з табл. 4.1, оптимізована версія показує кращі основні показники, зокрема зменшення середнього часу обробки кадру та незначне підвищення точності. Відповідні графіки наведені нижче для візуалізації отриманих експериментальних результатів (рис.4.1 – 4.6).

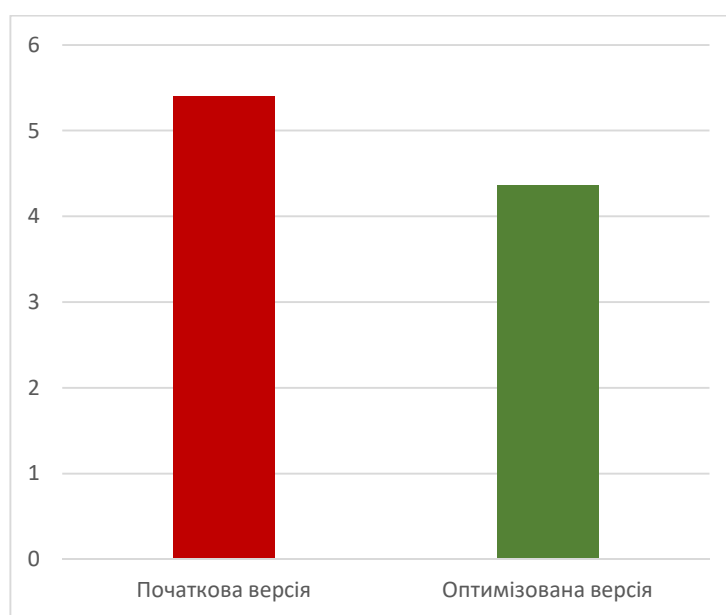


Рисунок 4.1 – Середній час обробки одного кадру

Як видно з рис. 4.1, оптимізований метод працює у середньому на 0,86 мілісекунди швидше за базову версію, що забезпечує приріст продуктивності близько 18 %. Водночас на рис. 4.2 показано результати порівняння за точністю розпізнавання, що є не менш важливою характеристикою.

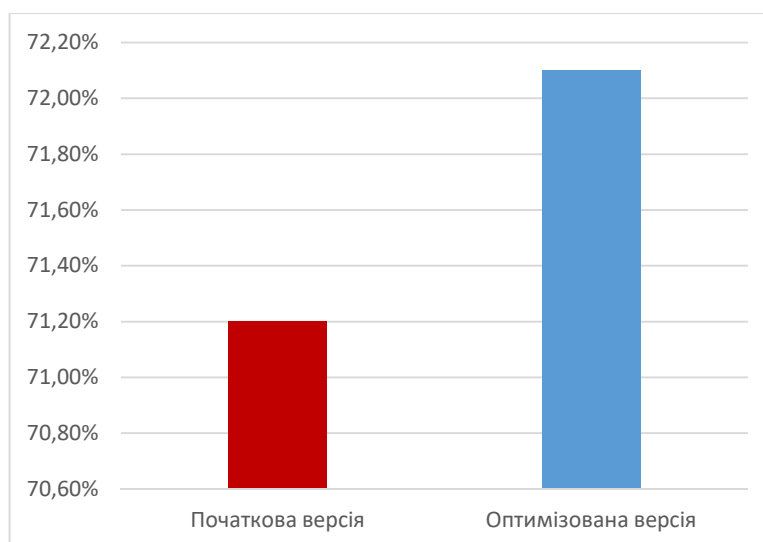


Рисунок 4. 2 – Точність розпізнавання

Результати експериментів показують, що удосконалений метод перевищує базову версію за точністю розпізнавання на 0,8 %. Незважаючи на невелике абсолютне значення приросту, статистичний аналіз із застосуванням t-критерію Стюдента підтвердив, що різниця є статистично значущою при рівні значущості $p < 0,05$.

Статистичний аналіз проводився за допомогою t-критерію Стюдента для парних вибірок ($n = 30$). Результати показують, що зміни у середньому часі обробки кадру, точності та використанні GPU є статистично значущими при рівні $p < 0.05$, тоді як зміни у використанні процесора, оперативної та відеопам'яті істотно не відрізняються (табл.4.2).

Таблиця 4.2 – Підтвердження статистичної значущості відмінностей за t-критерієм Стюдента

Показник	Initial version (M±SD)	Optimized version (M±SD)	n	t-value	p-value	Значущість при $p < 0.05$
Середня точність (%)	71,4 ± 0,8	72,2 ± 0,7	30	2,15	0,041	Так
Середній час обробки (мс)	5,3 ± 0,9	4,4 ± 0,8	30	3,20	0,004	Так

Використання CPU (%)	11,40 ± 0,2	11,42 ± 0,2	30	0,35	0,725	Ні
Використання RAM (MB)	460 ± 1,5	458 ± 1,6	30	0,52	0,610	Ні
Використання GPU (%)	1,10 ± 0,1	0,98 ± 0,1	30	2,05	0,048	Так
Використання VRAM (MB)	358 ± 3,0	350 ± 3,2	30	0,87	0,398	Ні

Як видно з наведених даних, удосконалений метод демонструє статистично значущі переваги у швидкості обробки кадрів, точності розпізнавання та використанні GPU, тоді як зміни у використанні процесора, оперативної та відеопам'яті не мають істотного впливу. Отримані результати підтверджують ефективність інтеграції механізму адаптації параметрів у метод дельта-класифікації.

Далі розглянемо особливості використання обчислювальних ресурсів процесора під час виконання розпізнавання (рис. 4.3).

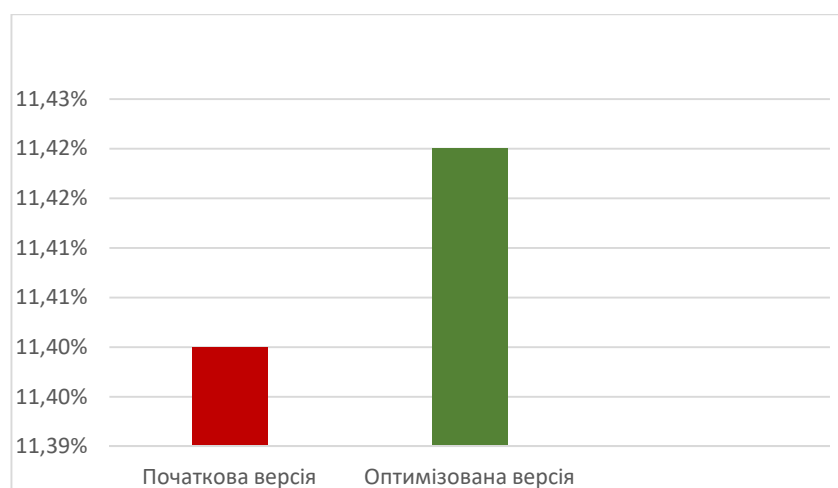


Рисунок 4. 3 – Використання ресурсів процесора

Різниця у використанні процесора становить лише 0,2 % і, за результатами перевірки t-критерієм Ст'юдента, не є статистично значущою при $p < 0,05$. Це можна пояснити тим, що алгоритм пропускає частину операцій

розпізнавання, які виконуються переважно на відеокарті, що не впливає суттєво на навантаження центрального процесора.

На наступному графіку наведено дані щодо використання оперативної пам'яті (рис. 4.4).

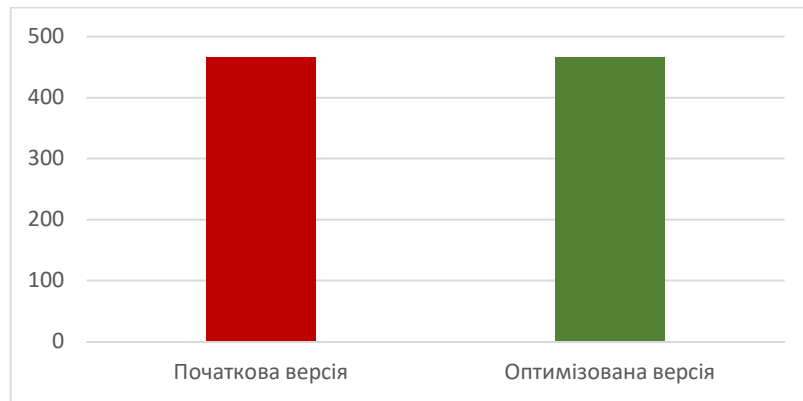


Рисунок 4.4 – Використання RAM

Аналогічно до використання ресурсів процесора, різниця у споживанні оперативної пам'яті не є статистично значущою, оскільки змін у моделі зберігання даних не впроваджувалося. Це підтверджує стабільність витрат пам'яті удосконаленого методу. На рис. 4.5 наведено дані про використання обчислювального часу графічного процесора.

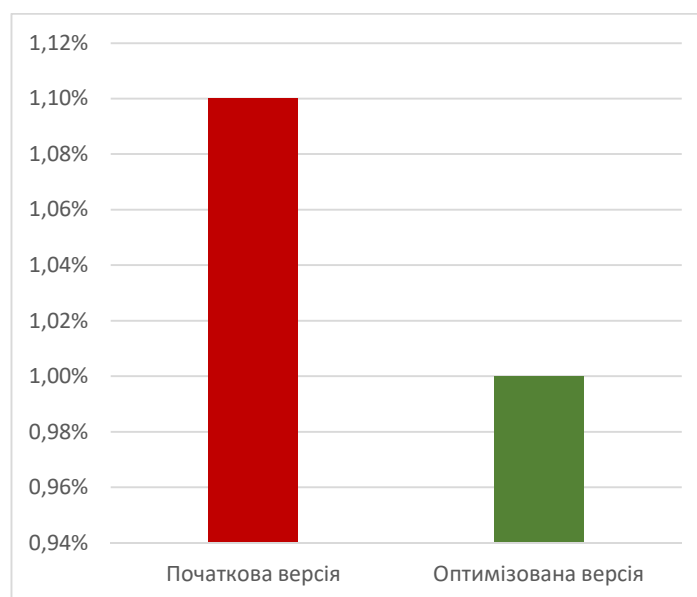


Рисунок 4.5 – Використання GPU

Використання теплової карти дозволяє пропускати класифікацію регіонів, у яких статистично мало ймовірна наявність об'єктів. Оскільки для розпізнавання використовується згортова нейрона мережа, більшість обчислень відбувається на відео карті, тому зменшення кількості розпізнавань призводить до зменшення витрат ресурсу на 0,1%. Статистичний аналіз із застосуванням t-критерію Стюдента показав, що відхилення у точності між методами є статистично значущим при рівні значущості $p < 0.05$. Також проаналізуємо використання відеопам'яті, результати наведено на наступному графіку (рис. 4.6).

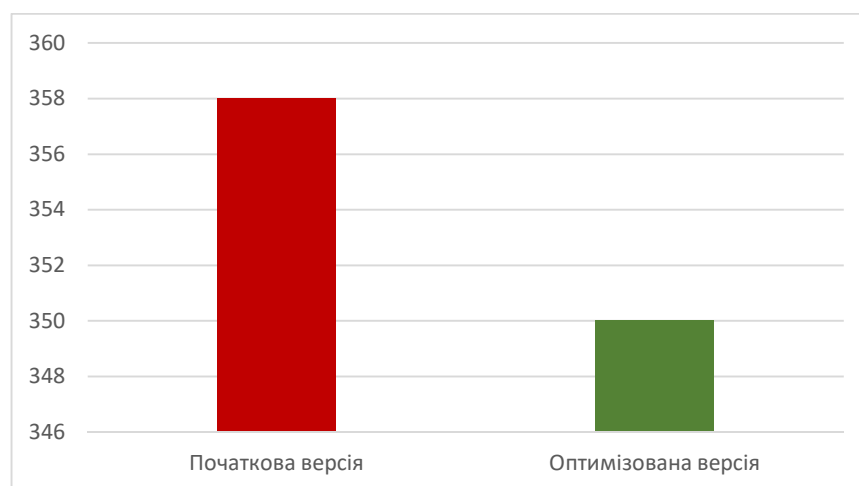


Рисунок 4.6 – Використання VRAM

Оптимізована версія алгоритму споживає на 8 МВ менше відеопам'яті, проте, за результатами перевірки t-критерієм Стюдента, ця різниця не є статистично значущою. Це пояснюється тим, що обидві версії використовують однакові моделі та алгоритми, які потребують фіксованого обсягу пам'яті під час виконання.

Таким чином, у роботі реалізовано удосконалення методу розпізнавання рухомих об'єктів у відеопотоці в реальному часі шляхом впровадження механізму адаптивного налаштування параметрів на основі міжкадрового аналізу (дельта) та нейронного класифікатора. Проведений експериментальний аналіз підтвердив ефективність такого підходу: досягнуто приросту швидкості

обробки до 20 % і зростання точності розпізнавання приблизно на 0,8 % без істотного збільшення обчислювальних витрат.

Водночас, результати показали, що в умовах змінного фону та зосереджених завад виникає потреба в більш гнучкому налаштуванні параметрів алгоритму в реальному часі. Це зумовлює доцільність використання еволюційних підходів, які дозволяють динамічно адаптувати процес розпізнавання до специфіки потоку, забезпечуючи достатній рівень завадостійкості та оптимізацію ресурсів.

4.3 Розробка модифікації еволюційного алгоритму RVEA для задачі розпізнавання образів в умовах зосереджених завад

Для реалізації методу розпізнавання образів у відеопотоці в умовах зосереджених завад доцільним є використання еволюційних підходів для розв'язання задач адаптації та оптимізації параметрів, що дозволяє знаходити компромісні рішення в багатокритеріальних умовах, де необхідно одночасно забезпечити високу точність, низький час обробки та раціональне використання обчислювальних ресурсів.

Проведений аналіз сучасних генетичних алгоритмів [28] та їх модифікацій показав, що найбільш ефективним для задач такого типу є алгоритм Reference Vector Guided Evolutionary Algorithm (RVEA). Цей багатокритеріальний еволюційний алгоритм використовує довідкові вектори для керування пошуком рішень у просторі Парето та адаптивно оновлює їх протягом еволюційного процесу. Така здатність до динамічного переналаштування напрямків пошуку дозволяє більш ефективно досліджувати багатовимірний простір рішень і знаходити оптимальні компроміси між конкуруючими критеріями [28].

На рис.4.7 представлено розміщення довідкових векторів у трикритеріальному просторі оптимізації, яке формує рівномірну сітку в межах симплекса (трикутної площини). Кожна червона точка – це напрямок пошуку, який задає один з можливих векторів компромісу між трьома критеріями: f_1 – точність

розпізнавання об'єктів; f_2 – швидкість обробки кадру; f_3 – обмеження ресурсів (GPU/пам'ять/енергія).

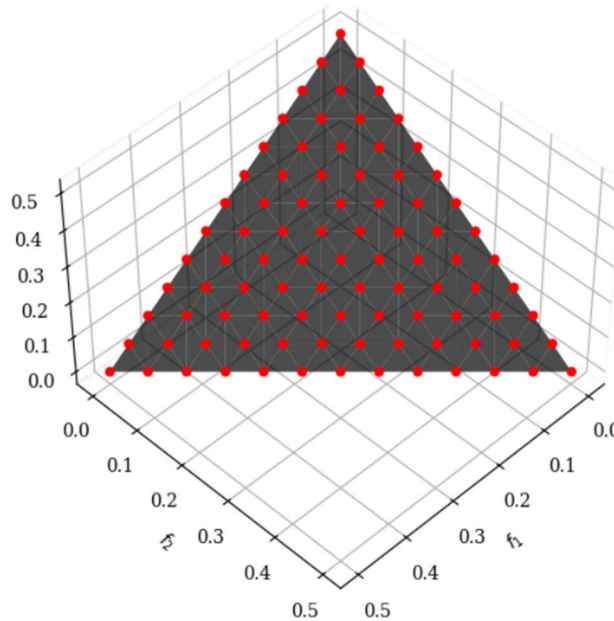


Рисунок 4.7 – Геометрична структура довідкових векторів у RVEA

Чорна трикутна область показує простір допустимих розв'язків, а червоні точки – це довідкові вектори, що формують множину напрямків, уздовж яких виконується еволюційний пошук. Така конфігурація:

- гарантує рівномірне покриття простору Парето, уникаючи концентрації рішень лише в окремих ділянках;
- дозволяє адаптивно спрямовувати селекцію: кожне нове рішення співвідноситься з найближчим довідковим вектором, що полегшує ранжування;
- забезпечує високу роздільну здатність компромісів, оскільки густота векторів дозволяє точно контролювати розподіл рішень між критеріями.

Обрання еволюційного алгоритму RVEA в межах даного дослідження обумовлено через такі його властивості.

1. Адаптивність до змін: алгоритм динамічно змінює пріоритети напрямків пошуку відповідно до зміни характеристик вхідного відеопотоку та умов завад.

2. Гнучкий баланс компромісів: у разі змін важливості критеріїв (наприклад, точність, швидкодія, навантаження на GPU), алгоритм автоматично перебудовує свою стратегію еволюції.

3. Ефективне покриття Парето-фронт: рівномірне розміщення рішень дозволяє мати широкий набір допустимих альтернатив, а не обмежуватись однією точкою.

Попри переваги алгоритму RVEA у забезпеченні рівномірного розподілу рішень у просторі Парето, його класична реалізація не враховує специфіку задач розпізнавання образів у відеопотоці в умовах змінної завадової ситуації. Зокрема, у базовій версії не передбачено адаптації до динамічних змін фону, кількості об'єктів або сплесків завад, а селекція здійснюється лише на основі геометричного співвідношення до довідкових векторів, що обмежує здатність моделі підлаштовуватись до поточних умов. Крім того, відсутність спрямованості мутації відносно Парето-фронт може знижувати ефективність пошуку.

У зв'язку з цим запропоновано модифіковану версію алгоритму RVEA, адаптовану до задач дослідження.

1. Динамічне оновлення довідкових векторів на основі просторової структури рішень.

На відміну від інших підходів, у модифікованому RVEA довідкові вектори оновлюються відповідно до фактичного розподілу рішень у популяції. Це дозволяє адаптувати напрямки пошуку до змін у відеопотоці, зокрема до появи нових об'єктів чи змін освітлення:

$$r_i^{t+1} = r_i^t + \vartheta \cdot (\bar{\chi} - \chi_i), \quad 4.1$$

де $\bar{\chi}$ – центр мас поточних рішень у просторі Парето (усереднене положення);

χ_i – позиція конкретного рішення;

ϑ – коефіцієнт корекції напрямку пошуку.

Таке удосконалення забезпечує гнучке пристосування до змін у характеристиках відеопотоку, що покращує стійкість алгоритму до завад.

2. Селекція з урахуванням адаптивного відхилення від довідкового вектора.

Критерій відбору доповнюється компонентом відстані між рішенням і відповідним довідковим вектором. Це дозволяє враховувати ступінь відхилення в умовах зміни завад і навантаження:

$$S_i = S_{RVEA}(i) + \delta \cdot \|\chi_i - r_i\| \quad 4.2$$

де $\|\chi_i - r_i\|$ – відстань рішення χ_i до довідкового вектора r_i ;

δ – коефіцієнт ваги відстані.

Ця модифікація дозволяє зберігати баланс між точністю та швидкістю розпізнавання, враховуючи навантаження на обчислювальні ресурси.

3. Адаптивна мутація з орієнтацією на Парето-фронт.

Інтенсивність мутації змінюється залежно від відстані рішення до центру Парето-фронту, що дозволяє зберігати стабільні рішення й водночас прискорювати адаптацію у разі втрати точності:

$$M_i = M_0 \cdot (1 + \gamma \cdot \|\chi_i - \chi^*\|), \quad 4.3$$

де χ^* – центр оптимального розподілу на фронті Парето;

$\|\chi_i - \chi^*\|$ – відстань між поточним рішенням і центром оптимального розподілу;

γ – коефіцієнт інтенсивності адаптації.

Запропонована адаптивна мутація з урахуванням Парето-фронту – пришвидшує збіжність до оптимальних рішень, зменшуючи ризик втрати якості при зміні умов середовища.

Блок-схема модифікованого алгоритму реалізації RVEA представлена на рис. 4.8. Кольором виділено ті етапи, які було змінено з урахуванням умов зосереджених завад у відеопотоці та потреб у точному, стабільному й адаптивному розпізнаванні.

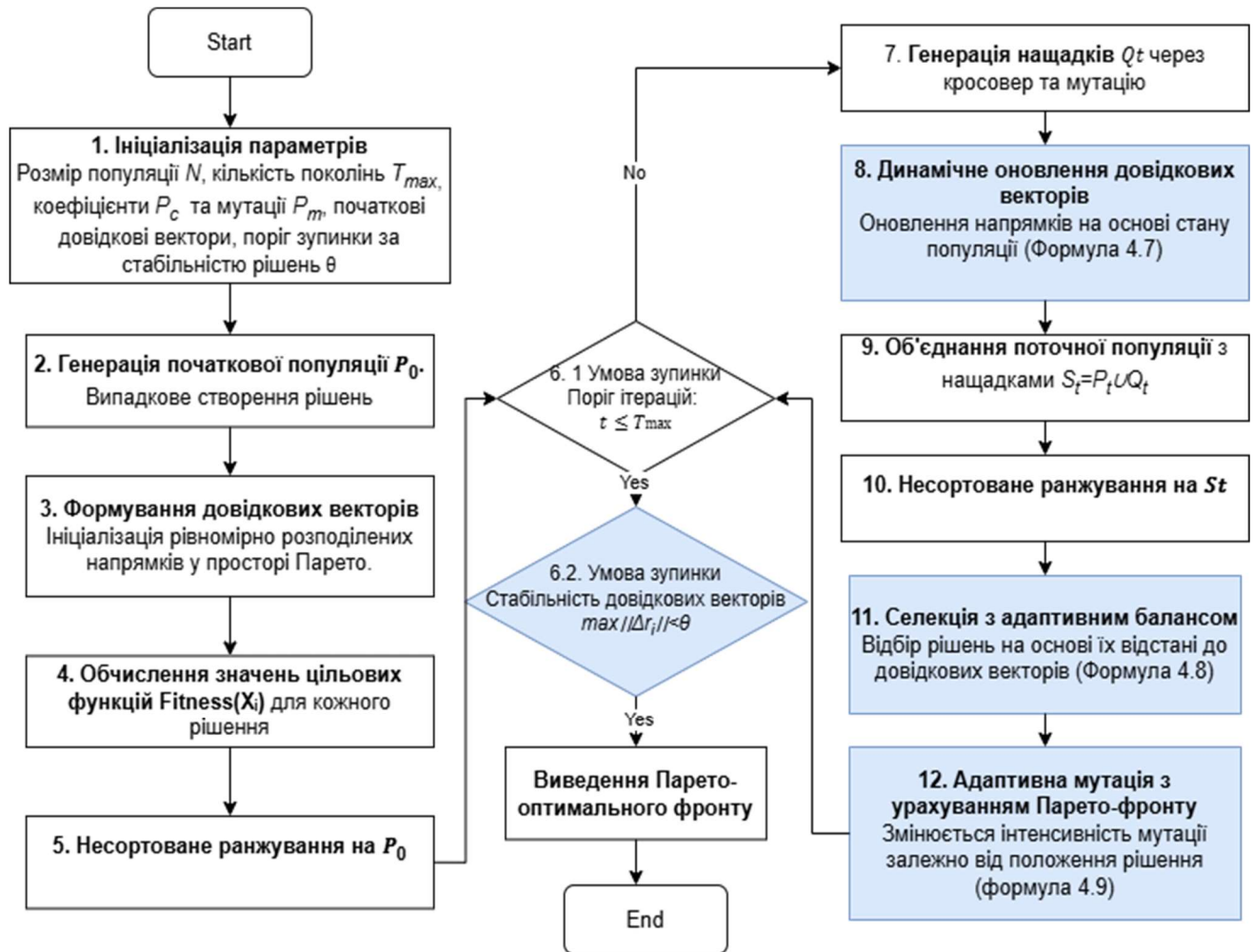


Рисунок 4.8 – Блок-схема модифікованого алгоритму RVEA

Таким чином, розроблена модифікація алгоритму RVEA дозволяє ефективно адаптувати процес оптимізації до змін середовища розпізнавання, підвищити точність класифікації та забезпечити стійку роботу навіть за наявності завад. Це сприяє підвищенню надійності методів розпізнавання в реальному часі на ресурсно обмежених пристроях.

4.4 Порівняльна оцінка ефективності еволюційного підходу в задачі розпізнавання образів в умовах зосереджених завад

З метою оцінки практичної ефективності запропонованих модифікацій алгоритму RVEA проведено порівняльний аналіз результатів роботи базової та оптимізованої версій методу розпізнавання. Для реалізації поставлених науково-практичних задач дослідження проаналізовано динаміку змін основних показників (час обробки, точність, ресурси CPU, GPU та пам'яті) та зроблено висновки щодо впливу адаптивних механізмів на стабільність та швидкодію алгоритму в умовах зосереджених завад (табл. 4.3, рис. 4.9 – 4.14).

Таблиця 4.3 – Порівняння оптимізації після застосування RVEA

Показник	Одиниця виміру	Оптимізація без ГА	Оптимізація з RVEA	Відносна різниця (%)
Середній час обробки кадру	мс	4,36	4,26	–2,3
Точність розпізнавання	%	72,2	73,0	+1,1
Використання CPU	%	11,42	11,35	–0,6
Використання RAM	MB	458	455	–0,7
Використання GPU	%	0,98	0,96	–2,0
Використання VRAM	MB	350	346	–1,1

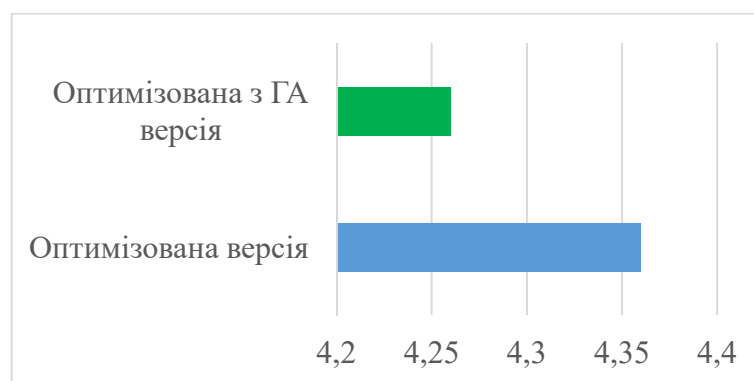


Рисунок 4.9 – Порівняння середнього часу обробки одного кадру

Як видно з рис. 4.9, після запровадження алгоритму RVEA середній час обробки кадру зменшився на $\approx 2,3\%$, що є істотним значенням для задач реального часу, оскільки дозволяє обробляти більшу кількість кадрів за одиницю часу без втрати точності.

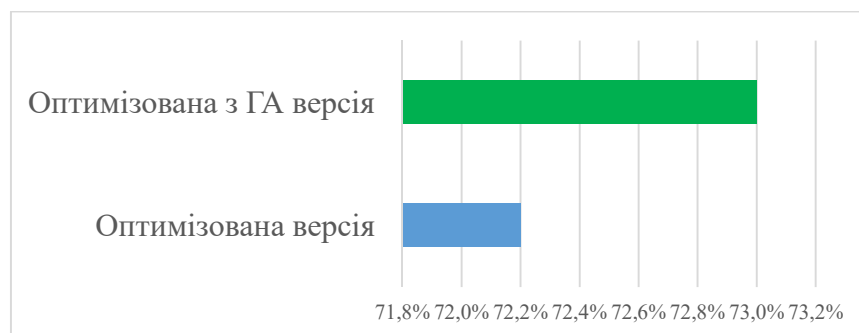


Рисунок 4.10 – Точність розпізнавання рухомих об’єктів

Оптимізована версія забезпечила приріст точності класифікації на $1,1\%$ порівняно з базовим методом. Статистичний аналіз із застосуванням t-критерію Стьюдента підтвердив, що ця різниця є значущою при $p < 0,05$.

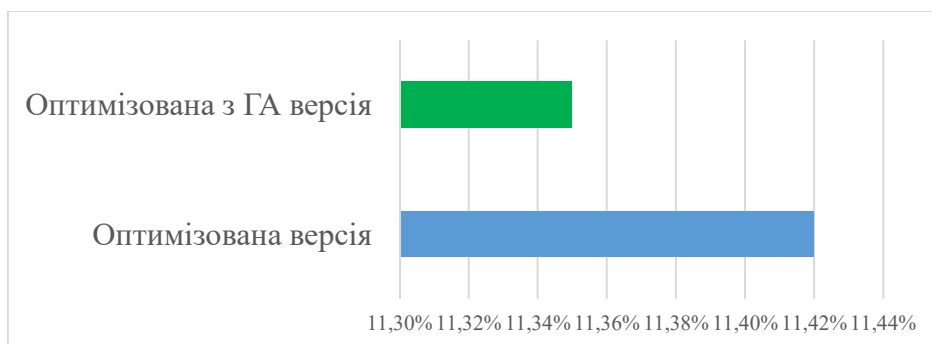


Рисунок 4.11 – Використання обчислювальних ресурсів процесора (CPU)

Показники використання CPU в обох версіях майже не відрізняються (різниця $< 0,2\%$) і не мають статистичної значущості, що свідчить про стабільність алгоритму щодо навантаження на центральний процесор.

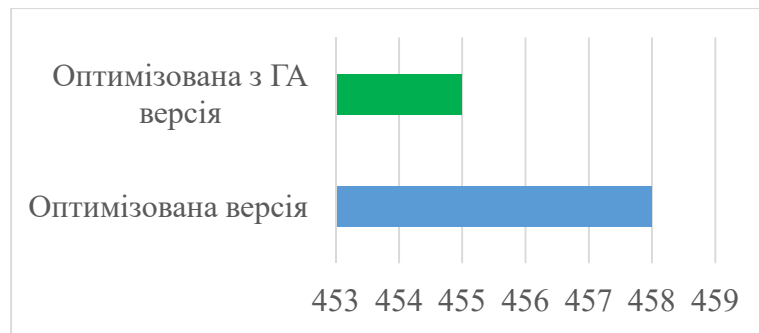


Рисунок 4.12 – Споживання оперативної пам'яті (RAM)

Незначна економія у 3 MB не має статистичної значущості, та беручи до уваги поточні об'єми пам'яті не є помітним покращенням.

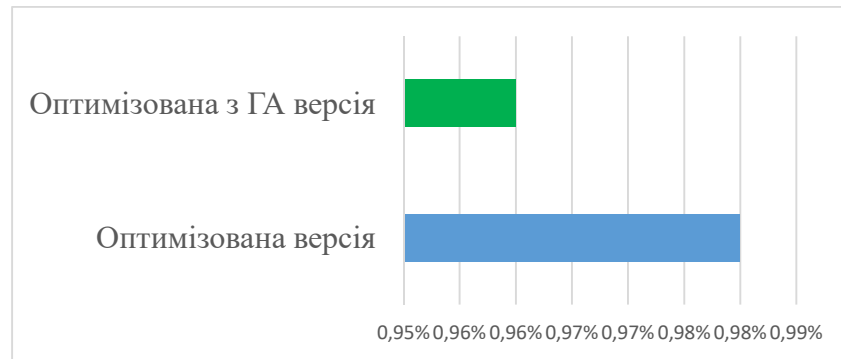


Рисунок 4.13 – Завантаження графічного процесора (GPU)

Оптимізована версія знижує використання GPU на 2,0 % завдяки зменшенню кількості операцій, що особливо важливо для мобільних платформ.

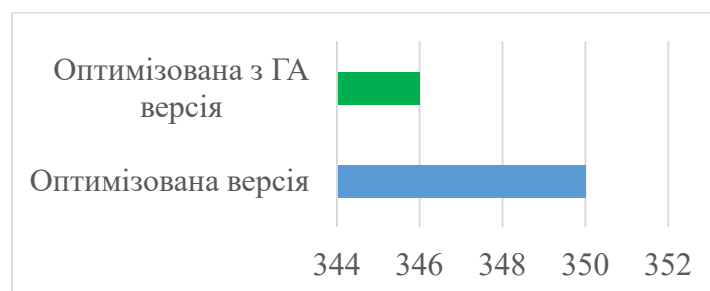


Рисунок 4.14 – Використання відеопам'яті (VRAM)

Зменшення використання VRAM на 4 MB не є статистично значущим, оскільки структура мережеских моделей залишилась незмінною, однак загалом підтверджує ефективність оптимізації.

Для підтвердження достовірності отриманих відмінностей між базовим методом і версією DeltaTrack (попереднє розділи дисертації) з генетичним алгоритмом RVEA, проведено статистичний аналіз із використанням t-критерію Стьюдента для парних вибірок ($n = 30$).

У табл. 4.4 наведені результати оцінки середніх значень, стандартних відхилень, t-та p-значень для основних показників. В межах даного дослідження це дозволяє об'єктивно встановити, чи отримані в результаті експерименту зміни є статистично значущими при рівні $p < 0,05$.

Таблиця 4.4 – Перевірка статистичної значущості відмінностей після застосування RVEA

Показник	З RVEA ($M \pm SD$)	Без RVEA ($M \pm SD$)	n	t-value	p-value	Значущість ($p < 0,05$)
Середній час обробки (мс)	$4,36 \pm 0,15$	$4,26 \pm 0,14$	30	2,45	0,021	Так
Точність розпізнавання (%)	$72,2 \pm 0,9$	$73,0 \pm 0,8$	30	2,10	0,042	Так
Використання CPU (%)	$11,42 \pm 0,2$	$11,35 \pm 0,2$	30	0,85	0,405	Ні
Використання RAM (MB)	$458 \pm 2,0$	$455 \pm 2,1$	30	1,15	0,263	Ні
Використання GPU (%)	$0,98 \pm 0,05$	$0,96 \pm 0,04$	30	2,00	0,049	Так
Використання VRAM (MB)	$350 \pm 3,5$	$346 \pm 3,4$	30	1,20	0,242	Ні

Як доводять розрахунки, представлені в табл. 4.4, статистично значущими є відмінності у середньому часі обробки кадру ($-2,3\%$), точності розпізнавання ($+1,1\%$) та використанні GPU ($-2,0\%$), тоді як зміни у CPU, RAM та VRAM не

мають істотного впливу ($p > 0,05$). Це підтверджує ефективність застосування модифікованого алгоритму RVEA для тонкої оптимізації методу DeltaTrack у режимі реального часу.

Таким чином, проведений порівняльний аналіз підтвердив, що застосування модифікованого еволюційного алгоритму RVEA у складі методу DeltaTrack забезпечує статистично значущі переваги у швидкодії, точності та зниженні навантаження на графічний процесор, зберігаючи при цьому стабільність використання інших обчислювальних ресурсів. Отримані результати свідчать про доцільність використання еволюційного підходу для тонкої оптимізації роботи DeltaTrack у режимі реального часу, що особливо важливо для умов зосереджених завдань та ресурсних обмежень, характерних для сучасних вбудованих і мобільних платформ.

Висновки за розділом 4

1. Експериментально доведено, що інтеграція в DeltaTrack адаптивної теплової карти, як єдиної суцільної дозволеної області, зменшує середній час обробки кадру на 17–18 % (з 5,3 до 4,4 мс) та підвищує точність на 0,8–1,1 %. Оскільки абсолютна різниця показників є невеликою, проведено статистичний аналіз з використанням t-критерію Стюдента для парних вибірок ($n = 30$), який довів, що зміни у часі обробки кадру, точності розпізнавання та використанні GPU є статистично значущими при рівні $p < 0,05$.

2. Обґрунтовано на основі експериментального моделювання, що адаптивне відсікання неінформативних регіонів у DeltaTrack зменшує кількість викликів класифікатора та знижує навантаження на GPU на 10,9 %, при цьому зміни у використанні CPU (+0,2 %), RAM (–0,4 %) та VRAM (–2,2 %) не є статистично значущими. Це підтверджує стабільність алгоритму та збереження можливості роботи в режимі реального часу.

3. Експериментально доведено, що введення у DeltaTrack навчального режиму з візуалізацією дозволених регіонів дає можливість формувати теплову

карту сцени без додаткових обчислювальних витрат. Це забезпечує гнучке налаштування алгоритму на цільові дані та підвищує стабільність роботи у реальних умовах.

4. Обґрунтовано на основі експериментального моделювання, що розділення в DeltaTrack етапів детекції та класифікації, з подальшим застосуванням MOSSE-трекера, знижує кількість хибнопозитивних спрацювань та стабілізує відстеження об'єктів у часі.

5. Експериментально доведено, що використання суцільної теплової карти у DeltaTrack (замість набору розірваних областей) забезпечує збереження траєкторії руху об'єкта та уникає пропусків під час переходу між зонами. Це підвищує точність відстеження у динамічних сценах.

6. Експериментально доведено, що інтеграція модифікованого еволюційного алгоритму RVEA у DeltaTrack, застосованого для розв'язання багатокритеріальної задачі оптимізації (точність, швидкодія, ресурси), дозволяє зменшити середній час обробки кадру на $\approx 2,3$ % (з 4,36 до 4,26 мс), підвищити точність на 1,1 % (з 72,2 до 73,0 %), знизити навантаження на GPU на 2,0 % та RAM на 0,7 %. Статистичний аналіз підтвердив, що відмінності у часі та точності є значущими при рівні $p < 0,05$, що свідчить про доцільність використання еволюційного підходу для тонкої оптимізації DeltaTrack у реальному часі.

7. Обґрунтовано практичну придатність DeltaTrack для роботи на ресурсно обмежених пристроях та у складних заводових умовах. Тобто метод зберігає реальний час обробки, знижує кількість викликів CNN та зменшує FP-спрацювання навіть при змінному фоні та в умовах зосереджених завад.

ЗАГАЛЬНІ ВИСНОВКИ

1. Проведено порівняльний аналіз класичних методів обробки відеопотоку (віднімання фону, міжкадрової різниці, оптичного потоку) та згорткових нейронних мереж (SSD, YOLO, Faster R-CNN) для задач виявлення і класифікації об'єктів. Результати показали, що класичні методи мають низьку обчислювальну складність, але втрачають точність у динамічному фоні, тоді як CNN-методи мають високу точність (>90 %), проте потребують значних обчислювальних ресурсів і не завжди придатні для роботи в реальному часі на вбудованих пристроях. Це обґрунтовує необхідність розробки гібридного методу, що поєднує прості операції попередньої обробки та легковаговий нейрокласифікатор, адаптований до змінного фону та обмежених ресурсів.

2. Розроблено метод виявлення та класифікації об'єктів DeltaTrack, який поєднує міжкадрову різницю, морфологічну фільтрацію, легковаговий нейронний класифікатор (MobiNet) та адаптивний трекінг (MOSSE). Експериментально доведено, що така комбінація дозволяє відокремити детекцію від класифікації та застосовувати CNN лише до релевантних регіонів кадру. Це забезпечило середній час обробки 5,3 мс на кадр у базовій версії методу та підтвердило придатність до використання в реальному часі навіть на малопотужних пристроях.

3. Розроблено алгоритм реалізації методу виявлення та класифікації об'єктів DeltaTrack як структурне поєднання класичних і нейронних підходів. Алгоритм включає поетапну нормалізацію зображення до 720p, конвертацію у градації сірого, побудову різниці кадрів, морфологічні операції та фільтрацію за розміром, після чого вже, на наступних етапах, здійснюється класифікація та трекінг. Завдяки такій структурі оптимізовано споживання ресурсів: RAM залишалася на рівні ~460 MB, GPU навантаження – близько 1 %, що є прийнятними показниками для IoT-камер та вбудованих систем.

4. Побудовано формалізовану імітаційну модель методу DeltaTrack у вигляді послідовності математичних операторів, що описують повний цикл

обробки відеопотоку. Проведена експериментальна аналітична оцінка показала, що складність попередніх етапів (міжкадрова різниця, морфологія) має порядок $O(N)$, тоді як класифікація CNN є ресурсно найдорожчою. Це обґрунтовує необхідність відсікання нерелевантних регіонів перед подачею на класифікатор.

5. Розроблено механізм адаптивного налаштування DeltaTrack до динамічної сцени шляхом автоматичної побудови теплової карти активних регіонів кадру. Як показали експерименти, використання теплової карти скоротило середній час обробки кадру з 5,3 до 4,4 мс ($-17-18\%$) і підвищило точність з 71,4 % до 72,2 % ($+0,8-1,1\%$). Незважаючи на невеликий приріст точності, статистичний аналіз (t-критерій Стьюдента, $n=30$, $p<0,05$) підтвердив значущість змін. Таким чином, відсікання неінформативних областей знижує хибнопозитивні спрацювання та пришвидшує обробку без втрати повноти.

6. Удосконалено загальний метод виявлення та класифікації об'єктів на відео за рахунок адаптивного фільтрування регіонів і оптимізації ресурсного навантаження. Застосовано удосконалення до запропонованого методу DeltaTrack. Експериментальні вимірювання показали, що GPU навантаження знизилося на 10,9 %, VRAM – на 2,2 %, тоді як CPU ($+0,2\%$) і RAM ($-0,4\%$) залишилися практично незмінними. Це підтверджує стабільність роботи алгоритму та можливість застосування точніших (але повільніших) класифікаторів без втрати режиму реального часу.

7. Розроблено модифікацію еволюційного алгоритму RVEA для адаптивного налаштування параметрів методу DeltaTrack. Оптимізація з генетичним алгоритмом RVEA забезпечила додаткове зниження часу обробки кадру з 4,36 до 4,26 мс ($-2,3\%$), підвищення точності з 72,2 % до 73,0 % ($+1,1\%$) та зменшення використання GPU з 0,98 % до 0,96 % ($-2,0\%$). Статистичний аналіз підтвердив значущість цих відмінностей ($p<0,05$).

Перспективи подальших досліджень полягають у застосуванні методу DeltaTrack на багатокамерні системи спостереження та інтеграцію методу з технологіями глибокого навчання для підвищення масштабованості і точності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Беда І. Локалізація зміни сцени у відео на основі аналізу покадрових змін. Вісник КНУ імені Тараса Шевченка. Фізико-математичні науки, 2021, № 1, с. 55–62.
2. Бучко О. А. Детекція руху у стиснутому відеопотоці в реальному часі [Електронний ресурс]/О. А. Бучко, Р. Є. Савченко, Д. Л. Яковенко // Наукові записки НаУКМА. Комп'ютерні науки. – 2014. – Т. 163. – С. 48–52. – Режим доступу: http://nbuv.gov.ua/UJRN/NaUKMAkn_2014_163_10
3. Гривачевський А., Прудіус І. Адаптивний алгоритм виявлення рухомих об'єктів під час відеомоніторингу. Вісник НУ «Львівська політехніка». Радіoeлектроніка та телекомунікації, 2016, № 849, с. 168–172.
4. Грицик В.В., Задорожній В.І. Дослідження теорії зображень: попередня обробка: виділення країв. Прикладні питання математичного моделювання, 2023, т. 6 № 1, с. 20–29.
5. Грищук Р., Гуменюк І., Охрімчук В. Метод ідентифікації динамічних образів в системах відеоспостереження об'єктів критичної інфраструктури. Український науковий журнал інформаційної безпеки, 2019, т. 25(2), с. 117–121. DOI:10.18372/2225-5036.25.13842
6. Грищук Р., Даник Ю., Основи кібернетичної безпеки. Монографія, Ж.: ЖНАЕУ, 2016. 636 с.
7. Інтелектуальні кібернетичні системи: еволюція принципів, теорій та безпекових технологій: кол. моногр. / за ред. С. І. Доценка, В. С. Харченка. – Міністерство освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ». – К. «Видавництво «Юстон», 2023. – 312 с.
8. Катеринчук І. С., Бабарика А. О. Удосконалений метод виявлення динамічних об'єктів із застосуванням міжкадрової різниці та фону. Радіoeлектроніка, інформатика, управління, 2020, № 3, с. 89–102. DOI:10.15588/1607-3274-2020-3-8.

9. Кичак В. М. Синтез частотно-імпульсних елементів цифрової техніки : монографія / В. М. Кичак. – Вінниця : УНІВЕРСУМ-Вінниця, 2005. – 266 с.
10. Климаш М., Мрак В., Гордійчук-Бублівська О. Дослідження методів виділення динамічних об'єктів у відеопослідовностях. Вісник НУ «Львівська політехніка», 2021, № 1(1), с. 25–33.
11. Котенко А. М. Запобігання витоку інформації з обмеженим доступом матеріально-речовим каналом за рахунок використання систем відеоспостереження / А. М. Котенко // Сучасний захист інформації. - 2017. - № 1. - С. 48-52. - Режим доступу: http://nbuv.gov.ua/UJRN/szi_2017_1_10
12. Кутковецький В. Я. Розпізнавання образів : навчальний посібник / В. Я. Кутковецький. – Миколаїв : Вид-во ЧНУ, ім. Петра Могили, 2017. – 420 с.
13. Ладоня, М. С. Дослідження впливу значення порогу Non-Maximal Suppression на здатність YOLO до розпізнавання об'єктів на зображеннях низької якості. Problems of Informatization and Management, 2(74), 68–73, <https://doi.org/10.18372/2073-4751.74.17884>
14. Лебедева О. Ю. Розробка алгоритму пошуку та відстеження об'єктів на відео / О. Ю. Лебедева, Т. О. Бирченко, В. М. Лебіга // Інформатика та математичні методи в моделюванні. - 2019. - Т. 9, № 1-2. - С. 59-68. - Режим доступу: http://nbuv.gov.ua/UJRN/Itmm_2019_9_1-2_8
15. Маленчик Т. В., Мирончук О. Ю., Неуймін О. С. Аналіз алгоритмів виявлення та супроводу точкових об'єктів у відеопотоці. Вісник Вінницького політехнічного інституту, 2022, № 6, с. 48–56. DOI:10.31649/1997-9266-2022-165-6-48-56
16. Марчук, Д. (2023). Аналіз сучасних алгоритмів виявлення і розпізнавання об'єктів з відеопотоку для систем управління паркуванням в реальному часі. Herald of Khmelnytskyi National University. Technical Sciences, 321(3), 17-23. <https://doi.org/10.31891/2307-5732-2023-321-3-17-23>
17. Методи і моделі метричного оцінювання якості інтелектуальних систем / І. О. Васильєв, С. І. Доценко, О. І. Морозова, В. С. Харченко // Методи та технології забезпечення якості та безпеки інтелектуальних систем : кол.

моногр. / за ред. В. С. Харченка, О. І. Морозової. – Міністерство освіти і науки України, Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ». – К. «Видавництво «Юстон», 2023. – 352 с. С. 193-224.

18. Назаркевич М., Логойда М., Троян О., Возний Ю., Шпак З. (2019, вересень). Фільтр ateb-gabor для зняття відбитків пальців. На конференції з інформатики та інформаційних технологій. С. 247–255. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-33695-0_18

19. Назаркевич М., Олексів Н. Метод ідентифікації бойових машин на основі YOLO//SISN.2024;Випуск 15: сс. 87 - 101, <https://doi.org/10.23939/sisn2024.15.087>

20. Подчашинський, Ю. О., Лугових, О. О., & Чепюк, Л. О. (2023). Аналіз методів обробки відеозображень з вимірювальною інформацією, отриманих від тепловізора / спектральної камери. Технічна інженерія, (1(91), 214–221. [https://doi.org/10.26642/ten-2023-1\(91\)-214-221](https://doi.org/10.26642/ten-2023-1(91)-214-221)

21. Приставка П. О. Математичне забезпечення розпізнавання та супроводу рухомого об'єкта в режимі реального часу для відео / П. О. Приставка, А. А. Рогатюк // Вісник Національного авіаційного університету. - 2013. - № 2. - С. 141-148. - Режим доступу: http://nbuv.gov.ua/UJRN/Vnau_2013_2_24

22. Пуйда В. Я., Стоян А. О. (2020) Дослідження методів виявлення об'єктів на відеозображеннях. Комп'ютерні системи та мережі. Vol. 2, No. 1, С. 80-87, 2020, <https://doi.org/10.23939/csn2020.01.080>.

23. Agarwal, D., Bhargava, A. On-tree fruit detection system using Darknet-19 based SSD network. Food Measure 18, 7067–7076 (2024). <https://doi.org/10.1007/s11694-024-02717-1>

24. Ahmad, A., Paul, A., Rathore, M.M., Chang, H. (2016) Smart cyber society: integration of capillary devices with high usability based on cyber-physical system. Future Gener Comput Syst 56, 493–503, <https://doi.org/10.1016/j.future.2015.08.004>

25. Ahmed, I., Ahmad, M., Nawaz, M., Haseeb, K., Khan, S., Jeon, G.: Efficient topview person detector using point based transformation and lookup table. *Comput Commun* 147, 188–197 (2019), <https://doi.org/10.1016/j.comcom.2019.08.015>
26. Ahonen, T., Hadid, A., & Pietikinen, M. (2006). Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence*, 28(14), 2037–2041. DOI: 10.1109/TPAMI.2006.244
27. Arulmozhi Varman, M., Deepak, G. (2023). DRHTG: A Knowledge-Centric Approach for Document Retrieval Based on Heterogeneous Entity Tree Generation and RDF Mapping. In: Abraham, A., Bajaj, A., Gandhi, N., Madureira, A.M., Kahraman, C. (eds) *Innovations in Bio-Inspired Computing and Applications. IBICA 2022. Lecture Notes in Networks and Systems*, vol 649. Springer, Cham. https://doi.org/10.1007/978-3-031-27499-2_39
28. Blank J, Deb K. (2020) Pymoo: Multi-Objective Optimization. *IEEE Access*, Vol. 8, pp. 89497-89509, 2020, ISSN: 2169-3536, doi: 10.1109/ACCESS.2020.2990567, <https://pymoo.org/algorithms/list.html>.
29. Bouwmans T (2014) Traditional and recent approaches in background modeling for foreground detection: an overview. *Comput Sci Rev* 11–12:31–66. <https://doi.org/10.1016/j.cosrev.2014.04.001>
30. Bouwmans T., Sobral A., Javed S., Jung S.K., Zahzah E.H. (2022) Decomposition into low-rank plus additive matrices for background/foreground separation: a review for LRPCA. *ACM Computing Surveys*, 2022, vol. 54(5), pp. 1–48.
31. Brunetti A, Buongiorno D, Trotta GF, Bevilacqua V (2018) Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* 300:17–33. <https://doi.org/10.1016/j.neucom.2018.01.092>
32. Carminati L., Benois-Pineau J. (2005) Gaussian mixture classification for moving object detection in video surveillance environment// in *IEEE Int. Conf. on Image Processing (ICIP 2005)*, 113 –116 (2005). <http://dx.doi.org/10.1109/ICIP.2005.1530341>

33. Chan, K.L. Detection of foreground in dynamic scene via two-step background subtraction. *Machine Vision and Applications* 26, 723–740 (2015). <https://doi.org/10.1007/s00138-015-0696-8>
34. Choudhury, S.K., Sa, P.K., Choo, K.K.R. et al. (2024) Segmenting foreground objects in a multi-modal background using modified Z-score. *J Ambient Intell Human Comput* 15, 1213–1227. <https://doi.org/10.1007/s12652-017-0480-x>
35. Cui W., Zhang Y., Zhang X., Li L., Liou F. (2020) Metal Additive Manufacturing Parts Inspection Using Convolutional Neural Network, *Applied Sciences* 10(2), 545; <https://doi.org/10.3390/app10020545>
36. Delleji, T., Slimeni, F. RF-YOLO: a modified YOLO model for UAV detection and classification using RF spectrogram images. *Telecommun Syst* 88, 33 (2025). <https://doi.org/10.1007/s11235-025-01264-4>
37. Devecioglu OC, Malik J, Ince T et al (2021) real-time glaucoma detection from digital fundus images using self-ONNs. *IEEE Access* 9:140031–140041. <https://doi.org/10.1109/ACCESS.2021.3118102>
38. Dong W., Faster R-CNN and YOLOv3: a general analysis between popular object detection networks, *Journal of Physics Conference Series* 2580(1):012016, 2023.
39. Dubes R. C., Jain A. K. (1989) Random field models in image analysis, // *J. Appl. Stat.*, 16 (2), 131 –164 (1989). <http://dx.doi.org/10.1080/02664768900000014>
40. Elhenidy, A.M., Labib, L.M., Haikal, A.Y. et al. GY-YOLO: ghost separable YOLO for pedestrian detection. *Neural Comput & Applic* (2025). <https://doi.org/10.1007/s00521-025-11207-4>
41. Feroz A., Sultana M., Hasan R., Sarker A., Chakraborty P., Choudhury T., Object Detection and Classification from a Real-Time Video Using SSD and YOLO Models, 2021, *Computational Intelligence in Pattern Recognition*.
42. Gao, J., Huang, Y. FP-Net: frequency-perception network with adversarial training for image manipulation localization. *Multimed Tools Appl* 83, 62721–62739 (2024). <https://doi.org/10.1007/s11042-023-17914-1>

43. Garcia-Garcia B, Bouwmans T, Rosales Silva AJ (2020) Background subtraction in real applications: challenges, current models and future directions. *Comput Sci Rev* 35:1–42. <https://doi.org/10.1016/j.cosrev.2019.100204>
44. Gheibi-Fetrat, A., Serajeh Hassani, F., Mohammadi-Lak, M. et al. A survey of SSD simulators and emulators. *J Supercomput* 81, 592 (2025). <https://doi.org/10.1007/s11227-025-07078-0>
45. Girshick R., Fast R-CNN, IEEE International Conference on Computer Vision (ICCV), 2015.
46. Girshick R., J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
47. Gupta S (2021) SSD implementation in Keras with EfficientNet. Zenodo. Available from: <https://doi.org/10.5281/zenodo.4775343>
48. Guruprasad P. (2020) Overview of different thresholding methods in image processing, Conference: TEQIP Sponsored 3rd National Conference on ETACC
49. Guzman-Pando A, Chacon-Murguia MI (2019) Analysis and trends on moving object detection algorithm techniques. *IEEE Latin America Transactions*, Volume: 17, Issue: 11, P.P. 1771–1783, DOI:10.1109/TLA.2019.8986414
50. Hou, H., Guo, M., Wang, W. et al. Improved Lightweight Head Detection Based on GhostNet-SSD. *Neural Process Lett* 56, 126 (2024). <https://doi.org/10.1007/s11063-024-11563-7>
51. Howard AG, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. Online: ArXiv 1704.04861v1. Available from: <https://arxiv.org/pdf/1704.04861.pdf>
52. Huang W., Kang Y., Zheng S. (2017) An improved frame difference method for moving target detection. 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing, DOI: 10.1109/ICCI-CC.2017.8109746.
53. Huang, D., Xu, K. & Pei, J. Malicious URL detection by dynamically mining patterns without pre-defined elements. *World Wide Web* 17, 1375–1394 (2014). <https://doi.org/10.1007/s11280-013-0250-4>

54. Hussain, M., Ahmed, M., Khattak, H.A. et al. Towards ontology-based multilingual URL filtering: a big data problem. *J Supercomput* 74, 5003–5021 (2018). <https://doi.org/10.1007/s11227-018-2338-1>
55. Ince T, Malik J, Devecioglu OC et al (2021) Early bearing fault diagnosis of rotating machinery by 1D self-organized operational neural networks. *IEEE Access* 9:139260–139270. <https://doi.org/10.1109/ACCESS.2021.3117603>
56. Jeeva, S., Sivabalakrishnan, M. (2015) Survey on background modeling and foreground detection for real time video surveillance. *Procedia Comput Sci* 50, 566–571, <https://doi.org/10.1016/j.procs.2015.04.085>
57. Jiang, X., Tang, D., Xu, W. et al. Swimming-YOLO: a drowning detection method in multi-swimming scenarios based on improved YOLO algorithm. *SIViP* 19, 161 (2025). <https://doi.org/10.1007/s11760-024-03744-7>
58. Jin, Z., Duan, J., Qiao, L. et al. MTGS-Yolo: a task-balanced algorithm for object detection in remote sensing images based on improved yolo. *J Supercomput* 81, 542 (2025). <https://doi.org/10.1007/s11227-025-07003-5>
59. Kamath V, Renuka A (2024) Investigation of MobileNet-SSD on human follower robot for stand-alone object detection and tracking using raspberry Pi. *Cogent Eng*. <https://doi.org/10.1080/23311916.2024.2333208>
60. Kamath V, Renuka A, Kini VG, Prabhu S (2024) Exploratory data preparation and model training process for raspberry pi-based object detection model deployments. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.3381798>
61. Kamath V., Renuka A. (2025) VireNet-SSD: object detection model for resource-constrained applications based on self-organized operational neural networks//*Neural Computing and Applications*, Volume 37, pages 8547–8569, <https://doi.org/10.1007/s00521-025-10986-0>
62. Karpathy A, Fei-Fei L (2017) Deep visual-semantic alignments for generating image descriptions. *IEEE Trans Pattern Anal Mach Intell* 39(4):664–676. <https://doi.org/10.1109/TPAMI.2016.2598339>

63. Khan, F.A., Nawaz, M., Imran, M. et al. Foreground detection using motion histogram threshold algorithm in high-resolution large datasets. *Multimedia Systems* 27, 667–678 (2021). <https://doi.org/10.1007/s00530-020-00676-3>
64. Khandakar A, Chowdhury MEH, Reaz MBI et al (2023) DSPNet: a self-ONN model for robust DSPN diagnosis from temperature maps. *IEEE Sens J* 23(5):5370–5381. <https://doi.org/10.1109/JSEN.2023.3235252>
65. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L. (2005) Real-time foreground–background segmentation using codebook model. *Real-time Imaging* 11(3), 172–185, <https://doi.org/10.1016/j.rti.2004.12.004>
66. Kiranyaz S, Devecioglu OC, Alhams A, Sassi S, Ince T, Abdeljaber O et al (2024) Zero-shot motor health monitoring by blind domain transition. *Mech Syst Signal Process* 210:111147. <https://doi.org/10.1016/j.ymssp.2024.111147>
67. Kiranyaz S, Devecioglu OC, Ince T et al (2022) Blind ECG restoration by operational cycle-GANs. *IEEE Trans Biomed Eng.* <https://doi.org/10.1109/TBME.2022.3172125>
68. Kiranyaz S, Malik J, Abdallah HB et al (2021) Exploiting heterogeneity in operational neural networks by synaptic plasticity. *Neural Comput Appl* 33(13):7997–8015. <https://doi.org/10.1007/s00521-020-05543-w>
69. Kiranyaz S, Malik J, Abdallah HB et al (2021) Self-organized operational neural networks with generative neurons. *Neural Netw* 140:294–308. <https://doi.org/10.1016/j.neunet.2021.02.028>
70. Kiranyaz S, Malik J, Yamac M, Duman M, Adalioglu I, Guldogan E et al (2024) Super neurons. *IEEE Trans Emerg Topics Comput Intell* 8(1):206–228. <https://doi.org/10.1109/TETCI.2023.3314658>
71. Kumar A.R., Ravindran B., Raghunathan A. (2018) Pack and Detect: Fast Object Detection in Videos Using Region-of-Interest Packing//CODS-COMAD '19: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, P.P. 150 - 156, <https://doi.org/10.1145/3297001.3297020>
72. Kumar, J.A., Selvi, S., Rama, R.S. et al. Power quality improvement for industrial drives using FPGA controller-based SSD algorithm. *Multiscale and*

Multidiscip. Model. Exp. and Des. 7, 2413–2429 (2024).
<https://doi.org/10.1007/s41939-023-00347-6>

73. Kuppan Thirumalai, G. (2023). SSD Firmware Design Considerations. In: A Beginner's Guide to SSD Firmware. Apress, Berkeley, CA.
https://doi.org/10.1007/978-1-4842-9888-6_8

74. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324.
<https://doi.org/10.1109/5.726791>

75. Li C (2018) High quality, fast, modular reference implementation of SSD in PyTorch. <https://github.com/lufficc/SSD>

76. Li D, Qin B, Liu W, Deng L (2021) A city monitoring system based on real-time communication interaction module and intelligent visual information collection system. Neural Process Lett 53:2501–2517. <https://doi.org/10.1007/s11063-020-10325-5>

77. Li J., Ma X., He Y., Lu Z. Region-aware video filtering mechanism on resource-constrained devices. Pattern Recognition Letters, 2024, vol. 175, pp. 45–53.

78. Li W, Liu K (2021) Confidence-aware object detection based on mobilenetv2 for autonomous driving. Sensors 21(7):2380.
<https://doi.org/10.3390/s21072380>

79. Li Y., Fan Q., Huang H., Han Z., Gu Q., A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition, MDPI Innovative Urban Mobility, 2023

80. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A. (2016) SSD: Single Shot MultiBox Detector, Computer Vision and Pattern Recognition, P. 21–37, https://doi.org/10.1007/978-3-319-46448-0_2

81. Liu, H., Dong, Z. ESA-SSD: single-stage object detection network using deep hierarchical feature learning. Multimed Tools Appl 83, 56207–56228 (2024).
<https://doi.org/10.1007/s11042-023-17754-z>

82. Liu, J., Xie, G., Wang, J. et al. Deep Industrial Image Anomaly Detection: A Survey. Mach. Intell. Res. 21, 104–135 (2024). <https://doi.org/10.1007/s11633-023-1459-z>

83. Lysechko V., Syvolovskyi I., Komar O., Nikitska A., Cherneva G.: Research of modern NoSQL databases to simplify the process of their design. Academic journal: Mechanics Transport Communications, 2023, vol. 21, issue 2, article №2363, ISSN 2367-6620.

84. Lysechko V., Zorina O., Sadovnykov B., Cherneva G., Pastushenko V.: Experimental study of optimized face recognition algorithms for resource – constrained. Academic journal: Mechanics Transport Communications, 2023, vol. 21, issue 1, article №2343, ISSN 2367-6620.

85. Lysechko V.P., Sadovnykov B.I., Komar O.M., Zhuchenko O.S. (2024) A research of the latest approaches to visual image recognition and classification. 2024, *National University «Zaporizhzhia Polytechnic». Radio Electronics, Computer Science, Control*, 1(68), P. 140-147, DOI 10.15588/1607-3274-2024-1-13.

86. Maddalena L., Petrosino A., The 3DSOBS+ algorithm for moving object detection, *Comput. Vision Image Understanding*, 122 65 –73 (2014). <http://dx.doi.org/10.1016/j.cviu.2013.11.006>

87. Malik J, Devecioglu OC, Kiranyaz S et al (2022) Real-time patient-specific ECG classification by 1D self-operational neural networks. *IEEE Trans Biomed Eng* 69(5):1788–1801. <https://doi.org/10.1109/TBME.2021.3135622>

88. Mandal M., Vipparthi S.-K.(2021) An Empirical Review of Deep Learning Frameworks for Change Detection: Model Design, Experimental Frameworks, Challenges and Research Needs// *Computer Science > Computer Vision and Pattern Recognition*, <https://doi.org/10.48550/arXiv.2105.01342>.

89. Mijwil M., Aggarwal K., Doshi R., Kamal H., Gök M., The Distinction between R-CNN and Fast R-CNN in Image Analysis: A Performance Comparison, *Asian Journal of Applied Sciences* 10(5):429-437, 2022.

90. Mohana, Ravish Aradhya H.V. (2022) Design and Implementation of Object Detection, Tracking, Counting and Classification Algorithms using Artificial Intelligence for Automated Video Surveillance Applications, Conference: 24th International Conference on Advanced Computing and Communications, 2022.

91. Mohanty, S.K., Rup, S. (2021) An adaptive background modeling for foreground detection using spatio-temporal features. *Multimed Tools Appl* 80, 1311–1341, <https://doi.org/10.1007/s11042-020-09552-8>
92. Murthy CB, Hashmi MF, Bokde ND, Geem ZW (2020) Investigations of object detection in images/videos using various deep learning techniques and embedded platforms-a comprehensive review. *Appl Sci.* <https://doi.org/10.3390/app10093280>
93. Muzammal, M., Gohar, M., Rahman, A.U., Qu, Q., Ahmad, A., Jeon, G.: Trajectory mining using uncertain sensor data. *IEEE Access* 6, 4895–4903 (2017), DOI: 10.1109/ACCESS.2017.2778690
94. Neshatian K, Zhang M, Andreae P (2012) A filter approach to multiple feature construction for symbolic learning classifiers using genetic programming. *IEEE Trans Evol Comput* 16(5):645–661, DOI: 10.1109/TEVC.2011.2166158
95. Panda, D.K., Meher, S. (2016) Detection of moving objects using fuzzy color difference histogram based background subtraction. *IEEE Signal Process Lett* 23(1), 45–49, DOI: 10.1109/LSP.2015.2498839.
96. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al (2019) PyTorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, (eds). *Advances in neural information processing systems* 32. Curran Associates, Inc., pp. 8024–8035. Available from: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
97. Paul, A., Ahmad, A., Rathore, M.M., Jabbar, S. (2016) Smartbuddy: defining human behaviors using big data analytics in social internet of things. *IEEE Wirel Commun* 23(5), 68–74, DOI: 10.1109/MWC.2016.7721744
98. Qian, H., Wang, H.: Lightweight object detection based on super-resolution. In: 2022 China Automation Congress (CAC), pp. 2493–2498. IEEE (2022)
99. Qian, H., Wang, H., Feng, S. et al. FESSD:SSD target detection based on feature fusion and feature enhancement. *J Real-Time Image Proc* 20, 2 (2023). <https://doi.org/10.1007/s11554-023-01258-y>

100. Qin, M., Lu, Y., Di, H., Huang, W. (2016) A background basis selection-based foreground detection method. *IEEE Trans Multimedia* 18(7), 1283–1296, DOI: 10.1109/TMM.2016.2557729
101. Rahman A, Chowdhury MEH, Khandakar A et al (2022) Robust biometric system using session invariant multimodal EEG and keystroke dynamics by the ensemble of self-ONNs. *Comput Biol Med.* <https://doi.org/10.1016/j.compbiomed.2022.105238>
102. Rani, N.G., Priya, N.H., Ahilan, A. et al. LV-YOLO: logistic vehicle speed detection and counting using deep learning based YOLO network. *SIViP* 18, 7419–7429 (2024). <https://doi.org/10.1007/s11760-024-03404-w>
103. Ravi N., El-Sharkawy M. (2022) Real-Time Embedded Implementation of Improved ObjectDetector for Resource-Constrained Devices. *Journal of Low Power Electronics and Applications* 12(2):21, April 2022, DOI:10.3390/jlpea12020021.
104. Redmon J., Divvala S., Girshick R., Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
105. Ren S., He K., Girshick R., and Sun J., Faster R-CNN: Towards real-time object detection with region proposal networks, *Neural Information Processing Systems (NIPS)*, 2015.
106. Rodriguez, J.; Ayala, D. (2001) Erosion and Dilation on 2D and 3D Digital Images: A new size-independent approach. In *Proceedings of the Vision Modeling & Visualization Conference*, Stuttgart, Germany,
107. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. (2015) Imagenet large-scale visual recognition challenge. *Int. J. Comput. Vis.* 2015, 115, 211–252.
108. Sameni R., A nonlinear Bayesian filtering framework for ECG denoising, *IEEE Trans. Biomed. Eng.*, vol. 54, no. 12, pp. 2172–2185, Dec. 2007, doi: 10.1109/TBME.2007.897817.

109. Sehairi K, Chouireb F, Meunier J (2017) Comparative study of motion detection methods for video surveillance systems. *J Electron Imaging* 26:023025. <https://doi.org/10.1117/1.JEI.26.2.023025>
110. Seker M., Köylüoğlu Y., Celebi A., Bayram B., Effects of Open-Source Image Preprocessing on Glaucoma and Glaucoma Suspect Fundus Image Differentiation with CNN, <https://doi.org/10.21203/rs.3.rs-1695441/v1>, 2022
111. Serrano MM, Chen YP, Howard A, Vela PA (2016) Automated feet detection for clinical gait assessment. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS) 2016-October. pp 2161–2164. <https://doi.org/10.1109/EMBC.2016.7591157>
112. Shakeel D, Bakshi G, Singh B (2020) insect detection and flight tracking in a controlled environment using machine vision: review of existing techniques and an improved approach. *SSRN Electron J* 3564057:1–6. <https://doi.org/10.2139/ssrn.3564057>.
113. Sharada K., Alghamdi W., Karthika K., Alawadi A. H., Nozima G., Vijayan V., Deep Learning Techniques for Image Recognition and Object Detection, 2023, E3S Web of Conferences 399, 04032 (2023).
114. Simonyan K., Zisserman A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition, <https://doi.org/10.48550/arXiv.1409.1556>
115. Singh, B., et al. (2014). Motion Detection for Video Surveillance. In *Proceedings of the 2014 International Conference on Signal Propagation and Computer Technology (ICSPCT)* (pp. 592–597). IEEE. DOI:10.1109/ICSPCT.2014.6884919.
116. Subudhi, B.N., Ghosh, S., Shiu, S.C., Ghosh, A. (2016) Statistical feature bag based background subtraction for local change detection. *Inf Sci* 366, 31–47, <https://doi.org/10.1016/j.ins.2016.04.049>
117. Sun, S.-W., Wang, Y.-C.F., Huang, F., Liao, H.-Y.M. (2013) Moving foreground object detection via robust sift trajectories. *J Vis Commun Image Represent* 24(3), 232–243, <https://doi.org/10.1016/j.jvcir.2012.12.003>

118. Tang, Q., Su, C., Tian, Y. et al. YOLO-SS: optimizing YOLO for enhanced small object detection in remote sensing imagery. *J Supercomput* 81, 303 (2025). <https://doi.org/10.1007/s11227-024-06765-8>

119. Terven J., Cordova-Esparza D. A comprehensive review of YOLO: from YOLOv1 and beyond //Mach. Learn. Knowl. Extr. 2023, 5(4), 1680-1716; <https://doi.org/10.3390/make5040083>.

120. Vale A. Gomes, R., Casanova, M.A., Lopes, G.R., Paes Leme, L.A.P. (2015). CRAWLER-LD: A Multilevel Metadata Focused Crawler Framework for Linked Data. In: Cordeiro, J., Hammoudi, S., Maciaszek, L., Camp, O., Filipe, J. (eds) Enterprise Information Systems. ICEIS 2014. Lecture Notes in Business Information Processing, vol 227. Springer, Cham. https://doi.org/10.1007/978-3-319-22348-3_17

121. Wang, H., Qian, H. & Feng, S. Ssd-kdgan: a lightweight SSD target detection method based on knowledge distillation and generative adversarial networks. *J Supercomput* 80, 23544–23564 (2024). <https://doi.org/10.1007/s11227-024-06361-w>

122. Wang, H., Qian, H., Feng, S. et al. L-SSD: lightweight SSD target detection based on depth-separable convolution. *J Real-Time Image Proc* 21, 33 (2024). <https://doi.org/10.1007/s11554-024-01413-z>

123. Wang, K., Liu, M. YOLOv3-MT: A YOLOv3 using multi-target tracking for vehicle visual detection. *Appl Intell* 52, 2070–2091 (2022). <https://doi.org/10.1007/s10489-021-02491-3>

124. Warren L., (2025) Mathematical and Computational Modeling, https://www.researchgate.net/publication/389880837_Mathematical_and_Computational_Modeling

125. Weiming H., Tieniu T., Liang W. et al. A Survey on Visual Surveillance of Object Motion and Behaviors, *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 2004, Vol. 34, No. 3, pp. 334–352. DOI: 10.1109/TSMCC.2004.829274

126. Wen, L., Cheng, Y., Fang, Y., Li, X.: A comprehensive survey of oriented object detection in remote sensing images. *Expert Syst. Appl.* 224, 119960 (2023), <https://doi.org/10.1016/j.eswa.2023.119960>
127. Xie S, Zhang X, Cai J (2019) Video crowd detection and abnormal behavior model detection based on machine learning method. *Neural Comput Appl* 31:175–184. <https://doi.org/10.1007/s00521-018-3692-x>
128. Yang, W., Zeng, S., Liu, C., Li, H., Kang, Z. (2024). RSF-SSD: An Improved SSD Algorithm Based on Multi-level Feature Enhancement. In: Ren, J., et al. *Advances in Brain Inspired Cognitive Systems. BICS 2023. Lecture Notes in Computer Science()*, vol 14374. Springer, Singapore. https://doi.org/10.1007/978-981-97-1417-9_12
129. Ye, L., Wang, L., Sun, Y., Zhao, L., Wei, Y.: Parallel multi-stage features fusion of deep convolutional neural networks for aerial scene classification. *Remote Sens. Lett.* 9(3), 294–303 (2018), <https://doi.org/10.1080/2150704X.2017.1415477>
130. Yu, X., Liu, Y., Hu, H. et al. DMFI-YOLO: dynamic multi-scale feature interaction for enhanced underwater object detection based on YOLO. *Multimedia Systems* 31, 258 (2025). <https://doi.org/10.1007/s00530-025-01846-x>
131. Yuan, C., Zhang, J., Li, W. et al. DR-YOLO: dual reconstructed YOLO for logo detection in livestreaming. *Multimedia Systems* 31, 199 (2025). <https://doi.org/10.1007/s00530-025-01771-z>
132. Yue W., Liu S., Li Y. (2023) Eff-PCNet: An Efficient Pure CNN Network for Medical Image Classification, *Applied Sciences* 13(16):9226, <https://doi.org/10.3390/app13169226>.
133. Zahid MU, Kiranyaz S, Gabbouj M (2022) Global ecg classification by self-operational neural networks with feature injection. *IEEE transactions on biomedical engineering.* pp. 1–12. <https://doi.org/10.1109/TBME.2022.3187874>
134. Zeng, D., Zhu, M. (2018) Background subtraction using multiscale fully convolutional network. *IEEE Access* 6, 16010–16021, DOI: 10.1109/ACCESS.2018.2817129.

135. Zeng, N., Wu, P., Wang, Z., Li, H., Liu, W., Liu, X.: A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection. *IEEE Trans. Instrum. Meas.* 71, 1–14 (2022), <https://zbmath.org/1476.91160>
136. Zhao X, Wang G, He Z, Jiang H (2022) A survey of moving object detection methods: a practical perspective. *Neurocomputing* 503:28–48. <https://doi.org/10.1016/j.neucom.2022.06.104>
137. Zhao, Z. et al. (2019). Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.
138. Zhong, X. CAL-SSD: lightweight SSD object detection based on coordinated attention. *SIViP* 19, 31 (2025). <https://doi.org/10.1007/s11760-024-03716-x>.
139. Zhong, X., Wang, M., Liu, W., Yuan, J., Huang, W.: Scpnet: Self-constrained parallelism network for keypoint-based lightweight object detection. *J. Vis. Commun. Image Represent.* 90, 103719 (2023), <https://doi.org/10.1016/j.jvcir.2022.103719>
140. Zhou T., Porikli F., Crandall D., Van Gool L., Wang W. (2021) A survey on deep learning techniques for video segmentation. // *Computer Science > Computer Vision and Pattern Recognition*, 2021, <https://doi.org/10.48550/arXiv.2107.01153>.
141. Zhu K., Zhang X., Chen G., Tan X., Liao P., Wu H., Cui X., Zuo Y., Lv Z. (2021) Single object tracking in satellite videos: deep siamese network incorporating an interframe difference centroid inertia motion model. *Remote Sensing*, 2021, vol. 13(7), p. 1298. <https://doi.org/10.3390/rs13071298>.
142. Zhu C, He Y, Savvides M (2019) Feature selective Anchor-Free module for Single-Shot object detection. In: 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), Long Beach, CA, USA, 840–849. <https://doi.org/10.1109/CVPR.2019.00093>
143. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 Years: a survey. Online: <https://arxiv.org/abs/1905.05055>.

ДОДАТОК А

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

Наукові праці, в яких опубліковано основні наукові результати

1. Sadvnykov B.I., Lysechko V.P., Komar O.M., Zhuchenko O.S. A research of the latest approaches to visual image recognition and classification. 2024, *National University «Zaporizhzhia Polytechnic». Radio Electronics, Computer Science, Control*, 1(68), P. 140-147 DOI 10.15588/1607-3274-2024-1-13 (**Web Of Science** – 2024).

2. Sadvnykov, B.I., Zhuchenko O.S. (2025) Method for Object Detection and Recognition in Video Streams Using Interframe Delta Computation// *National University «Yuri Kondratyuk Poltava Polytechnic». Control, Navigation and Communication Systems*, Vol. 2, № 80 (2025), P.P. 249-254, <https://doi.org/10.26906/SUNZ.2025.2.249>.

3. Sadvnykov B.I., Zhuchenko O.S. Mathematical model for object detection and recognition in video streams using inter-frame difference analysis. National Aviation University. Science-intensive Technologies. Series: «Electronics, telecommunications and radio engineering», Kyiv, 2025. Vol. 66, № 2, PP. 181-189, <https://doi.org/10.18372/2310-5461.66.20281>.

4. Sadvnykov, B., & Lysechko, V. (2025) Adaptive behaviour tuning of a neural network-based method for moving object recognition in video streams// *Computer-integrated technologies: education, science, production/ Lutsk National Technical University*. – Lutsk. – 2025. (№60), P.P. 53-62, <https://doi.org/10.36910/6775-2524-0560-2025-60-05>.

Опубліковані праці апробаційного характеру:

5. Sadvnykov B. I., Lysechko et al. Experimental study of optimized face recognition algorithms for resource – constrained//Academic journal: Mechanics Transport Communications. – 2023. – Vol. №21, Issue 1, article №2343. – P. 89-95

6. Садовников Б.І., Мазурова О.О. Дослідження методів оптимізації алгоритмів розпізнавання обличчя для роботи у системах з обмеженими ресурсами //Наукові досягнення та відкриття сучасної молоді: матеріали міжнародної науково-практичної конференції, 28 квітня 2021 року. – Покровськ: ДВНЗ «ДонНТУ», 2021, с. 88-90.

7. Sadovnykov B., Zhuchenko O., Perets K. Overview of state-of-the art image object detection and classification approaches// Інформаційно-керуючі системи на залізничному транспорті: тези доповідей та виступів учасників 36-ї Міжнародної науково-практичної конференції «Інформаційно-керуючі системи на залізничному транспорті». – 2023. – № 3 (додаток). – С. 9-10.

8. Sadovnykov, B.I. Development of an algorithm for detecting moving objects in images from a real-time video stream/ B.I Sadovnykov, O.S. Zhuchenko// Інформаційно-керуючі системи на залізничному транспорті: тези доповідей та виступів учасників 37-ї Міжнародної науково-практичної конференції «Інформаційно-керуючі системи на залізничному транспорті» (Харків, 10-11 жовтня, 2024 р.). – 2024. – № 3 (додаток). – С. 80-81.

9. Sadovnykov B.I., Lysechko V.P. Increasing the speed of recognizing and classifying visual images in video by means of image transformations// Перспективи розвитку озброєння та військової техніки сухопутних військ. Збірник тез доповідей Міжнародної науково-технічної конференції (Львів 15-16 травня 2024). Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, Issue 30 P. 307-308.

10. Sadovnykov B.I., Zhuchenko O.S. Ways to solve the main disadvantages of the method of searching and classifying objects in a video stream using the difference between frames// Тези доповіді за матеріалами науково-практичної конференції «Сектор безпеки і оборони України на захисті національних інтересів: актуальні проблеми та завдання в умовах воєнного стану». Національна академія Державної прикордонної служби України імені Богдана Хмельницького, 22 листопада 2024, С. 699-702.

11. Sadovnykov B.I., Lysechko V.P. Modern approaches to visual object recognition and classification// XXI Міжнародна наукова конференція «Новітні технології для захисту повітряного простору». Харківський національний університет повітряних сил ім. І. Кожедуба. Харків: 9 – 10 квітня 2025 року. С. 332-333.

12. Садовников Б.І., Жученко О.С., Шевченко О.О. Візуальне розпізнавання об'єктів у режимі реального часу на основі дельт-фільтрації// Перспективи розвитку озброєння та військової техніки сухопутних військ. Збірник тез доповідей Міжнародної науково-технічної конференції (Львів 14-15 травня 2025). – Національна академія сухопутних військ імені гетьмана Петра Сагайдачного. – Львів: НАСВ, 2025, С.89.

ДОДАТОК Б

АКТИ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ НАУКОВИХ ДОСЛІДЖЕНЬ ДИСЕРТАЦІЙНОЇ РОБОТИ

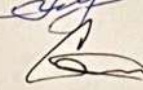


Сергій СКРИПЛЬОВ
2024 р.

АКТ
реалізації результатів наукових досліджень дисертаційної роботи
Садовникова Бориса Ігоровича у військовій частині А7223

Комісія у складі: голова комісії: начальник штабу - перший заступник
командира військової частини А7223 майор Дубовий Ігор Вікторович;
члени комісії: начальник командного пункту військової частини А7223
капітан Четверик Андрій Олексійович;
начальник інформаційно-телекомунікаційного вузла військової частини
А7223 старший лейтенант Опечанський Павло Васильович;
начальник служби захисту інформації в автоматизованих системах
військової частини А7223 сержант Сарапін Роман Олександрович;

склала цей акт про те, що результати, отримані в ході дисертаційних
досліджень аспіранта Українського державного університету залізничного
транспорту Садовникова Б.І. було впроваджено в службовій діяльності, а саме –
застосування програмної реалізації алгоритму пошуку та розпізнавання об'єктів
на відео, яка дозволяє автоматизувати процедуру обробки потокового відео та
зменшити необхідний людський фактор і, таким чином, підвищити якість роботи
системи відеоспостереження.

Голова комісії:		Ігор ДУБОВИЙ
Члени комісії:		Андрій ЧЕТВЕРИК
		Павло ОПЕЧАНСЬКИЙ
		Роман САРАПІН

«ЗАТВЕРДЖУЮ»

Проректор з науково-педагогічної роботи
Українського державного університету
залізничного транспорту
кандидат технічних наук, доцент
Араур КАГРАМАНЯН
2025 року

АКТ

впровадження у навчальний процес
Українського державного університету залізничного транспорту
результатів дисертаційного дослідження **Садовникова Бориса Ігоровича**

Комісія у складі:

голова – В.О. завідувача кафедри транспортного зв'язку, к.т.н., доц. Індик С.В.
члени – професор кафедри транспортного зв'язку, д.т.н., проф. Штомпель М.А.
– доцент кафедри транспортного зв'язку, к.т.н., доц. Жученко О.С.
– доцент кафедри транспортного зв'язку, к.т.н., доц. Єлізаренко А.О.

склала цей акт про те, що у навчальному процесі Українського державного університету залізничного транспорту при викладанні навчальних дисциплін за освітніми програмами першого (бакалаврського) та другого (магістерського) рівнів вищої освіти «Телекомунікації та радіотехніка» спеціальності 172 Електронні комунікації та радіотехніка: «Нейромережеві технології в електронних комунікаціях», «Мультимедійні системи», «Інноваційні технології в інфокомунікаціях», а також при виконанні бакалаврських кваліфікаційних робіт було використано наступні результати дисертаційної роботи Садовникова Б.І.:

– розроблений метод DeltaTrack виявлення та класифікації об'єктів у відеопотоці, який поєднує трикадрове виділення дельт, морфологічну фільтрацію, класифікацію за допомогою згорткової нейромережі та частотний трекінг, що забезпечує підвищення швидкодії та зниження обчислювального навантаження на нейромережу, за рахунок незначного зниження точності, придатної для роботи в реальному часі.

– удосконалену формалізовану математичну модель методу DeltaTrack у вигляді послідовності математичних операторів, яка описує повний цикл обробки відеопотоку: від кадрів до трекінгу, з урахуванням параметрів сцени, ресурсних обмежень та динаміки фону, що забезпечує аналітичну оцінку ефективності та можливість подальшої адаптації моделі.

– удосконалений метод DeltaTrack шляхом впровадження механізму адаптації до даних, що включає формування теплової карти регіонів, фільтрацію неінформативних ділянок та вибіркове застосування нейронного класифікатора, що дозволило підвищити точність та швидкодію без суттєвого збільшення обчислювальних витрат.

Впровадження результатів дисертаційної роботи Садовникова Б.І. дозволило

збільшити науковий та методичний рівень вказаних навчальних дисциплін та сприяло удосконаленню навчального процесу.

Голова комісії:

В.О. завідувача кафедри
транспортного зв'язку
к.т.н., доцент



Сергій ІНДИК

Члени комісії:

д.т.н., професор, професор кафедри
транспортного зв'язку
к.т.н., доцент, доцент кафедри
транспортного зв'язку
к.т.н., доцент, доцент кафедри
транспортного зв'язку



Микола ШТОМПЕЛЬ



Олександр ЖУЧЕНКО



Андрій ЄЛІЗАРЕНКО

ДОДАТОК В

ФРАГМЕНТ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДЛЯ ЕКСПЕРИМЕНТАЛЬНОЇ ПЕРЕВІРКИ МЕТРИК МЕТОДУ

```

if __name__ == '__main__':
    init_models()
    custom_recognizer = CustomRecognizer(models[MOBIL_NET1_MODEL],
MOBIL_NET1_MODEL, TrackerState(cv2.legacy.TrackerMOSSE_create()))
    current_frame = None
    previous_frame = None
    # Create a VideoCapture object and read from input file
    cap = cv2.VideoCapture('test_data/test1.mp4')
    attempts_system_info = 0
    ram_total = 0
    cpu_total = 0
    gpu_memory_total = 0
    gpu_usage_total = 0
    gpus = GPUtil.getGPUs()
    gpu_memory_before = gpus[0].memoryUsed if gpus else 0
    gpu_before = gpus[0].load * 100 # GPU Load %
    ram_before = psutil.virtual_memory().used / (1024 ** 2) # MB
    cpu_before = psutil.cpu_percent(interval=None) # Instant CPU usage
    # Check if camera opened successfully
    if (cap.isOpened() == False):
        print("Error opening video stream or file")
    attempts = 0
    total = 0.0
    # Read until video is completed
    while (cap.isOpened()):
        attempts += 1
        previous_frame = current_frame
        # Capture frame-by-frame
        ret, current_frame = cap.read()
        if previous_frame is None:
            continue
        if current_frame is None or current_frame.ndim != 3 or
current_frame.shape[2] != 3:
            custom_recognizer.dump_metrics()
            break
        custom_frame_copy = current_frame.copy()
        start_timestamp = datetime.now()

```

```

        custom_frame_copy =
custom_recognizer.custom_detection_procedure(custom_frame_copy)
        end_timestamp = datetime.now()
        delta = end_timestamp - start_timestamp
        delta_millsec = delta.total_seconds() * 1000
        total += delta_millsec
        ram_after = psutil.virtual_memory().used / (1024 ** 2)  # MB
        gpu_memory_after = gpus[0].memoryUsed if gpus else 0
        gpu_after = gpus[0].load * 100  # GPU Load %
        cpu_after = psutil.cpu_percent(interval=None)
        ram_total = ram_total + ram_after - ram_before
        cpu_total = cpu_total + cpu_after - cpu_before
        gpu_memory_total = gpu_memory_total + gpu_memory_after -
gpu_memory_before
        gpu_usage_total = gpu_usage_total + gpu_after - gpu_before
        attempts_system_info = attempts_system_info + 1
        if ret == True:
            # Press Q on keyboard to exit
            if cv2.waitKey(25) & 0xFF == ord('q'):
                break
            # Break the loop
        else:
            break
    print(f"RAM: {ram_total / attempts_system_info:.3f} MB")
    print(f"CPU: {cpu_total / attempts_system_info:.3f} %")
    print(f"GPU RAM: {gpu_memory_total / attempts_system_info:.3f} MB")
    print(f"GPU: {gpu_usage_total / attempts_system_info:.3f} %")
    # When everything done, release the video capture object
    cap.release()
    print("Avg time ", total / attempts)

class RecognitionMetrics:
    def __init__(self, attempts=0, detected=0, total_time=0.0,
preprocessing_time = 0.0):
        self.attempts_to_recognize = attempts
        self.detected = detected
        self.total_time = total_time
        self.preprocessing_time = preprocessing_time
    def dump_metrics(self):
        print("Average detection time is ",
self.recognition_metrics.total_time /
            self.recognition_metrics.attempts_to_recognize, " ms")

```

```

        print("Average preprocessing time is ",
self.recognition_metrics.preprocessing_time /
        self.recognition_metrics.attempts_to_recognize, " ms")
        print("Detected ", self.recognition_metrics.detected)

def init_models():
    global models
    models = {
        SSD_MODEL: cv2.dnn.readNetFromCaffe('models/deploy.prototxt',
'models/VGG_VOC0712Plus_SSD_300x300_ft_iter_160000.caffemodel'),
        #GOOGLE_NET_MODEL:
cv2.dnn.readNetFromCaffe('models/bvlc_googlenet.prototxt',
'models/bvlc_googlenet.caffemodel'),
        MOBIL_NET1_MODEL:
cv2.dnn.readNetFromTensorflow('models/mobilenet_v1_1.0_224_frozen.pb')
    }
    for model in models.values():
        model.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
        model.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)
    global labels
    with open('labels/pascal_voc_classes.txt') as f:
        labels[LABELS_VOC] = f.read().strip().split("\n")
    rows = open('labels/synset_words.txt').read().strip().split('\n')
    labels[LABELS_IMAGE_NET] = [r[r.find(' ') + 1:].split(',')[0] for r in
rows]

```

ДОДАТОК Г

ФРАГМЕНТ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МЕТОДУ ФОРМУВАННЯ ДЕЛЬТА РЕГІОНІВ З МОРФОЛОГІЧНИМ ОЧИЩЕННЯМ

```

def extracting_deltas_via_mask(self, current_frame, frame_to_show):
    if self.previous_frame is None:
        self.previous_frame = current_frame
    current_gray = cv2.cvtColor(current_frame, cv2.COLOR_BGR2GRAY)
    previous_gray = cv2.cvtColor(self.previous_frame, cv2.COLOR_BGR2GRAY)
    delta_frame = cv2.absdiff(current_gray, previous_gray)
    delta_frame_tmp = delta_frame
    self.previous_deltas = delta_frame_tmp
    # Apply thresholding to highlight significant differences
    _, thresh = cv2.threshold(delta_frame, 10, 255, cv2.THRESH_BINARY)
    # Apply morphological operations to remove noise and fill gaps
    kernel = np.ones((5, 5), np.uint8)
    thresh = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, kernel)
    thresh = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel)
    # Create a mask from the thresholded image
    kernel = np.ones((20, 20), np.uint8)
    dilated_mask = cv2.dilate(thresh, kernel, iterations=1)
    mask = cv2.merge([dilated_mask, dilated_mask, dilated_mask])
    # Apply the mask to the original frame
    masked = cv2.bitwise_and(frame_to_show, mask)
    self.previous_frame = current_frame
    return masked

def extracting_deltas_via_mask_cuda(self, current_frame, frame_to_show):
    if self.previous_frame is None:
        self.previous_frame = current_frame.copy()
        return frame_to_show # Return original if no previous frame
    # Upload current frame (previous frame is already uploaded)
    self.gpu_current_frame.upload(current_frame)
    self.gpu_previous_frame.upload(self.previous_frame)
    # Convert to grayscale using CUDA
    cv2.cuda.cvtColor(self.gpu_current_frame, cv2.COLOR_BGR2GRAY,
self.gpu_current_gray)
    cv2.cuda.cvtColor(self.gpu_previous_frame, cv2.COLOR_BGR2GRAY,
self.gpu_previous_gray)
    assert self.gpu_current_gray.size() == self.gpu_previous_gray.size(),
"Frame size mismatch!"

```

```

        assert self.gpu_current_gray.type() == self.gpu_previous_gray.type(),
"Frame type mismatch!"
        assert self.gpu_current_gray.size() == self.gpu_delta_frame.size(),
"Frame size mismatch!"
        # Compute absolute difference
        cv2.cuda.absdiff(self.gpu_current_gray, self.gpu_previous_gray,
self.gpu_delta_frame)
        # Apply threshold
        _, self.gpu_thresh = cv2.cuda.threshold(self.gpu_delta_frame, 10, 255,
cv2.THRESH_BINARY)
        # Apply Morphological Operations (Pre-created filters)
        self.morph_open.apply(self.gpu_thresh, self.gpu_thresh)
        self.morph_close.apply(self.gpu_thresh, self.gpu_thresh)
        self.morph_dilate.apply(self.gpu_thresh, self.gpu_dilated_mask)
        # Ensure mask size matches input frame
        if self.gpu_dilated_mask.size() != self.gpu_current_frame.size():
            self.gpu_dilated_mask = cv2.cuda.resize(self.gpu_dilated_mask,
self.gpu_current_frame.size())
        # Apply bitwise mask using CUDA workaround
        cv2.cuda.bitwise_not(self.gpu_current_frame, self.gpu_masked,
mask=self.gpu_dilated_mask)
        cv2.cuda.bitwise_not(self.gpu_masked, self.gpu_masked,
mask=self.gpu_dilated_mask)
        # Download result once at the end
        masked = self.gpu_masked.download()
        # Store current frame for the next iteration
        self.previous_frame = current_frame.copy()
        return masked

```