

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кваліфікаційна наукова  
праця на правах рукопису

**МРАК ВАСИЛЬ БОГДАНОВИЧ**

УДК 621.396

**ДИСЕРТАЦІЯ**

**МЕТОДИ ТА ЗАСОБИ РОЗПІЗНАВАННЯ РУХОМИХ ОБ’ЄКТІВ В  
ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ В РЕАЛЬНОМУ МАСШТАБІ  
ЧАСУ**

172 – Телекомунікації та радіотехніка  
(шифр і назва спеціальності)

17 «Електроніка та телекомунікації»  
(галузь знань)

**Дисертація на здобуття наукового ступеня  
доктора філософії**

Подається на здобуття наукового ступеня доктора філософії

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

\_\_\_\_\_ / Мрак Василь Богданович /

Науковий керівник

Климаш Михайло Миколайович д.т.н., проф.  
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

ЛЬВІВ – 2024

## АНОТАЦІЯ

*Мрак В.Б.* Методи та засоби розпізнавання рухомих об'єктів в телекомунікаційних системах в реальному масштабі часу. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 172 “Телекомунікації та радіотехніка”. – Національний університет «Львівська політехніка», Львів, 2024.

Зміст анотації. Дисертація присвячена розробленню методів і засобів розпізнавання рухомих об'єктів в телекомунікаційних системах в реальному масштабі часу.

Досягнення у сфері комп'ютерного зору не стоять на місці, так само, як і прогрес у цілому. До існуючих проблем додаються нові, які, у свою чергу, потребують також нових і ефективних рішень. Вони створюються для певної галузі та та вирішення певних проблеми, наприклад, глибоке навчання прийшло на зміну класичним методам комп'ютерного зору, бо була потреба у збільшенні якості задачі класифікації, на той час, класичні методи вирішували її не достатньо якісно. Так само можна сказати і про задачу ідентифікації, а саме про детектування та розпізнавання облич. Проблема захисту майна, об'єкту або іншого ресурсу не нова, але досі є актуальною. Для її вирішення застосовують засоби та системи безпеки. Розвитку останніх в значній мірі сприяв розвиток систем комп'ютерного зору, бо набагато зручніше встановити відео камери, та у автоматичному режимі аналізувати що відбувається. Наприклад, схожий сценарій реалізований у аеропортах, для підтримки безпеки. Він аналізує особи людей і шукає їх по базах даних, аналізує емоції, стан людини і багато інших характеристик на основі яких приймає рішення.

В першому розділі «Проблемні питання та аналіз технологій побудови систем відеоспостереження» описано особливості побудови та функціонування як звичайних систем відеоспостереження так і систем із застосуванням функцій відеоаналітики. Показано, що на сучасному етапі розвитку методів обробки інформації, застосування систем відеоспостереження як засобу фіксації подій є

недоцільним. Моніторинг обстановки в секторах огляду камер відеоспостереження в режимі реального часу є малоефективним, оскільки людина-оператор не має фізичних можливостей протягом тривалого часу сприймати інформацію від кількох (а то і кількох десятків) джерел одночасно. Саме для допомоги людині-оператору і призначені інтелектуальні системи відеоспостереження. Використання систем відеоспостереження з функціями відеоаналітики (інтелектуальних систем відеоспостереження, IVSS) дає можливість автоматизувати такі напрямки діяльності як аналітика контролю периметру, ситуаційний аналіз (автоматичне виявлення кризових ситуацій пов'язаних з накопиченням великої кількості людей), біометричний аналіз (дає змогу проводити автоматичну ідентифікацію осіб за зображеннями обличчя), аналіз номерних знаків транспортних засобів, аналіз з декількох камер (дає змогу здійснювати автоматичне супроводження об'єкта), автоматичне виявлення та класифікація об'єктів, пошук об'єктів в базі даних відеоархіву тощо.

Проведено оглядовий аналіз алгоритмів виявлення об'єктів та алгоритмів супроводження об'єктів, в результаті якого запропоновано спосіб класифікації алгоритмів супроводження об'єктів. Також проведено експериментальне дослідження ряду алгоритмів супроводження, в результаті якого сформовано рекомендації щодо умов застосування вказаних алгоритмів.

В другому розділі «Розроблення методів та моделей розпізнавання облич та стискання трафіку з застосуванням алгоритму віднімання фону для стаціонарних систем відеоспостереження» запропоновано ефективний алгоритм віднімання фону на основі ViBE для складного фону. Він поєднує в собі вдосконалений ViBE, який використовує кумулятивне середнє значення та механізм підрахунку пікселів для швидкого усунення привидів, і адаптивний ViBE, який обчислює адаптивний радіус залежно від зміни фону.

Запропоновано алгоритм для стиснення відео, знятих камерами з обмеженим ресурсом, який використовує аналіз руху для видалення небажаних пікселів і кадрів зображення.

Запропоновано метод розпізнавання обличчя, заснована на виявленні руху. Цей метод зосереджується не лише на етапах виявлення та розпізнавання облич, але включає виявлення рухомих об'єктів, що зменшує обсяг даних, які потрібно обробити.

Розроблено програмну модель обробки даних в системах відеонагляду. Доведено ефективність використання технології Spark. Ця технологія забезпечує швидку роботу алгоритмів навіть при збільшенні обсягів даних завдяки оптимізації обчислень.

В третьому розділі «Дослідження розроблених методів та моделей розпізнавання облич та стискання трафіку з застосуванням алгоритму віднімання фону» проведено дослідження створеного алгоритму віднімання фону на основі ViBE для складного фону.

Здійснено експерименти щодо стійкості моделей до повороту обличчя у різних площинах, зміни розміру обличчя, а також оцінено витрати часу. Найкращими моделями за параметрами обертання стали RetinaFaceResNet125, DSFD, RetinaFaceMobile Net0.25, які точно (точність  $\geq 0,9$ ) визначають обличчя із поворотами в діапазоні  $[-45; 45]$ , що є достатньою вимогою для використання у системах безпеки. Моделі MTCNN, FaceBoxes, CenterFace і SCRFD-500MF також працюють із поворотом обличчя в різних площинах, але у них нижча точність і менший діапазон кута обличчя.

В експериментах з різними розмірами обличчя найкращі результати показали моделі MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25, які виявляють зображення, починаючи з розміру  $75 \times 75$  px, та точністю  $\geq 0,9$ . Моделі CenterFace і SCRFD0.5GF мають значно нижчу точність, але в деяких випадках виявляють дуже маленьких розмірів обличчя.

Вимірювання часу виявлення об'єктів показало, що найшвидшими є RetinaFace-MobileNet0.25 і Face Box, які витрачають менше ніж 47 мс на опрацювання одного зображень VGA і 100 мс для розміру HG. На наступному місці SCRFD-500MF, CenterFace. Для моделей MTCNN і RetinaFaceResNet125

час виявлення перевищує 100 мс навіть для зображень VGA. Модель DSFD не можна застосовувати в реальному часі навіть для зображень VGA.

Досліджено створений метод для стиснення відео, знятих камерами з обмеженим ресурсом, який використовує аналіз руху для видалення небажаних пікселів і кадрів зображення. Аналізуючи відеокадри піксель за пікселем, метод досяг середнього стиснення тестових даних на 30%, зберігаючи при цьому всю інформацію, важливу для аналізу поведінки рухомих об'єктів. Це значне зменшення розміру файлу значно розширює можливості моніторингу за рахунок мінімізації вимог до пропускну здатності та полегшення ефективної передачі даних із пристроїв з обмеженими ресурсами. Розгортання цього методу на камерах має потенціал для значного просування моніторингу та збереження за рахунок оптимізації можливостей існуючих систем.

Досліджено запропонований метод розпізнавання обличчя, заснована на виявленні руху. Цей підхід має ряд переваг, таких як: зменшення обсягу даних, підвищення швидкості, зниження споживання енергії, універсальність, покращення продуктивності. Виявлення рухомих об'єктів дозволяє обробляти лише ті частини зображення, де відбуваються зміни, що значно зменшує обсяг даних, які потребують обробки. Система може обробляти зображення значно швидше, що робить її більш придатною для задач реального часу. Обробка меншого обсягу даних потребує менше енергії, що робить цю модель більш енергоефективною. Модель може бути адаптована до різних середовищ та умов освітлення. Завдяки поєднанню виявлення руху з методами розпізнавання облич, модель може підвищити точність за допомогою використання додаткової інформації про рух може допомогти усунути помилки, пов'язані з шумом та складними фоновими зображеннями. Зменшити кількість помилкових спрацювань за рахунок реагування лише на ті об'єкти, які дійсно рухаються, що мінімізує ймовірність помилкового розпізнавання облич.

В четвертому розділі « Програмна реалізація моделі розпізнавання облич на основі алгоритму віднімання фону» для створення застосунку була використані мови програмування Python разом з бібліотеками OpenCV,

TensorFlow, pickle. З їх допомогою були створені частини системи, які можуть зменшити навантаження на телекомунікаційну мережу та підвищити ефективність розпізнавання та ідентифікації облич.

У висновках дисертаційної роботи викладено основні результати і рекомендації, які випливають з проведених досліджень, запропоновано та охарактеризовано кількісні оцінки показників ефективності в умовах використання запропонованих рішень.

У додатках до дисертації долучено програмні акти впровадження результатів дисертаційної роботи, а також список наукових праць і апробацій автора за темою дисертації.

Ключові слова: інтелектуальна система відеоспостереження, виявлення динамічних об'єктів, ідентифікація облич, розпізнавання облич, удосконалений ViBe, CCTV, VSS.

## СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

*Публікації, у яких опубліковані основні наукові результати дисертації:*

1. Климаш М. М., Гордійчук-бублівська О. В., Мрак В. Б., Браницький А. В. Дослідження ефективності обробки великих даних в системах відеонагляду // Інфокомунікаційні та комп'ютерні технології. 2022. № 1 (01). С. 52–62. (Належить до фахових видань України. технічні науки ISSN 2788-5518)  
<https://doi.org/10.36994/2788-5518-2021-01-01-033>

*Особистий внесок здобувача: опрацювання результатів експериментальних досліджень, проведення аналізу даних в системах відеонагляду, обговорення одержаних результатів, підготовка статті до друку.*

2. Климаш М.М., Мрак В.Б., Гордійчук-Бублівська О.В., Дослідження методів виділення динамічних об'єктів у відеопослідовностях // Вісник «Інфокомунікаційні технології та електронна інженерія», №1, С. 63-75, 2021  
<https://doi.org/10.23939/ictee2021.01.063>

*Особистий внесок здобувача: опрацювання результатів експериментальних досліджень виділення динамічних об'єктів у відеопослідовностях, обговорення одержаних результатів, підготовка статті до друку.*

3. Юнак О. М., Климаш М. М., Шпур О. М., Мрак В. Б. Математична модель розпізнавання фрактальних структур з використанням технології нейронних мереж // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2023. Vol. 3, № 1. Р. 1–9. (Належить до фахових видань України. технічні науки ISSN 2786-4553)  
<https://doi.org/10.23939/ictee2023.01.001>

*Особистий внесок здобувача: участь у дослідженнях розпізнавання фрактальних структур з використанням технології нейронних мереж, опрацювання результатів досліджень, обговорення одержаних результатів, підготовка статті до друку.*

4. Мрак В. Б. Методи розпізнавання обличчя у системах відеоспостереження з використанням машинного навчання // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2023. Vol. 3, № 2. Р. 33–42. (Належить до фахових видань України. технічні науки ISSN 2786-4553)

<https://doi.org/10.23939/ictee2023.02.033>

*Особистий внесок здобувача: дослідження методів розпізнавання обличчя у системах відеоспостереження, інтерпретація результатів, написання статті.*

5. Мрак В. Б., Климаш М. М., Масюк А. Р., Колодій Т. Б. Методи та показники оцінки ефективності при розробці інтелектуальних систем відеоспостереження // Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки. 2024. Т. 35 (74), № 1. С. 252–259. (Належить до фахових видань України. технічні науки ISSN 2663-5941 E-ISSN 2663-595X)

<https://doi.org/10.32782/2663-5941/2024.1.1/38>

*Особистий внесок здобувача: опрацювання результатів оцінки ефективності при розробці інтелектуальних систем відеоспостереження, участь у інтерпретації результатів, написання статті.*

6. Мрак В. Б., Климаш М. М. Модель системи розпізнавання обличчя для нерухомих систем відеоспостереження // Вимірювальна та обчислювальна техніка в технологічних процесах. 2024. № 1. С. 68–73. (Належить до фахових видань України. технічні науки ISSN 2219-9365)

<https://doi.org/10.31891/2219-9365-2024-77-9>

*Особистий внесок здобувача: системи розпізнавання обличчя для нерухомих систем відеоспостереження, реалізація моделювання дослідження, написання статті.*

7. Мрак В. Б., Климаш М. М., Бабинець В. В. Удосконалення методу виявлення динамічних об'єктів у відео послідовностях // Вимірювальна та



обчислювальна техніка в технологічних процесах. 2024. № 2. С. 195–204.  
(Належить до фахових видань України. технічні науки ISSN 2219-9365)

<https://doi.org/10.31891/2219-9365-2024-78-22>

*Особистий внесок здобувача: дослідження удосконаленого методу виявлення динамічних об'єктів у відео послідовностях, обговорення одержаних результатів, підготовка статті до друку.*

8. Климаш М. М., Пиріг Ю. В., Сеник А. Д., Мрак В. Б. Дослідження контекстно-чутливого алгоритму моніторингу кібербезпеки на основі рекурентних нейронних мереж // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2024. Vol. 4, № 1. Р. 1–9. (Належить до фахових видань України. технічні науки ISSN 2786-4553)

<https://doi.org/10.23939/ictee2024.01.001>

*Особистий внесок здобувача: аналіз результатів дослідження моніторингу кібербезпеки, інтерпретація результатів, написання статті.*

9. Zhuravel S., Klymash M., Shpur O., Mrak V. Reducing the impact of unstable connections among nodes of wireless IIoT clusters using machine learning methods // Lecture Notes in Electrical Engineering. 2024. Vol. 1198 : Digital ecosystems: interconnecting advanced networks with AI applications. Р. 144–159. (Наукове серійне видання Netherlands, ISSN: 1876-1100, E-ISSN: 1876-1119)

*Особистий внесок здобувача: дослідження даних в IIoT за допомогою методів машинного навчання, інтерпретація результатів, написання статті.*

### **Публікації, які засвідчують апробацію матеріалів дисертації:**

1. “Алгоритми та методи виявлення динамічних об'єктів інтелектуальних систем відеоспостереження”, Мрак В. Б., V Міжнародна науково-практична конференція “Наукоємні технології в інфокомуніаціях” НІСТ'2023 1-3 червня 2023

<http://repositsc.nuczu.edu.ua/bitstream/123456789/18013/1/%D0%9D%D0%86%D0%A1%D0%A2%202023.pdf>.

*Особистий внесок здобувача: опрацювання результатів досліджень методів виявлення динамічних об'єктів інтелектуальних систем відеоспостереження, аналіз та інтерпретація результатів, обговорення одержаних результатів, апробація матеріалів на конференції*

2. Intelligent IoT control system based on hand gesture recognition / D. Balazh, V. Mrak, A. Sydor, V. Andrushchak, B. Rusyn, T. Maksymyuk // Advanced information and communication technologies : proceedings of the 5th IEEE International conference (Lviv, Ukraine, November 21–25, 2023). 2023. P. 102–104.

<https://doi.org/10.1109/AICT61584.2023.10452708>

*Особистий внесок здобувача: опрацювання результатів досліджень аналізу даних в IoT, обговорення одержаних результатів, апробація матеріалів на конференції.*

## ABSTRACT

Mrak V.B. Methods and means of recognition of moving objects in telecommunication systems in real time. – Qualifying scientific work on manuscript rights. Dissertation for the Doctor of Philosophy degree in specialty 172 "Telecommunications and radio engineering". - Lviv Polytechnic National University, Lviv, 2024.

Abstract content. The dissertation is devoted to the development of methods and means of recognition of moving objects in telecommunication systems in real time.

Advances in computer vision do not stand still, nor does progress in general. New problems are added to existing problems, which, in turn, also require new and effective solutions. They are created for a specific field and problem, for example, deep learning replaced classical methods of computer vision, because there was a need to increase the quality of the classification task, at that time, classical methods did not solve it sufficiently well. The same can be said about the task of identification, namely the detection and recognition of faces. The problem of protecting property, object or other resource is not new, but it is still relevant. Security tools and systems are used to solve it. The development of the latter was greatly facilitated by the development of computer vision systems, because it is much more convenient to install a video camera and analyze what is happening in automatic mode. For example, a similar scenario is implemented in airports to support security. It analyzes people's faces and searches for them in databases, analyzes emotions, the state of a person and many other characteristics on the basis of which it makes decisions.

The first chapter "Problematic issues and analysis of video surveillance system construction technologies" describes the features of construction and operation of both conventional video surveillance systems and systems with the use of video analytics functions. It is shown that at the current stage of development of information processing methods, the use of video surveillance systems as a means of

recording events is impractical. Real-time monitoring of the situation in the inspection sectors of video surveillance cameras is ineffective, since the human operator does not have the physical ability to perceive information from several (or even several dozen) sources simultaneously for a long time. It is to help the human operator that intelligent video surveillance systems are designed. The use of video surveillance systems with video analytics functions (intelligent video surveillance systems, IVSS) makes it possible to automate such activities as perimeter control analytics, situational analysis (automatic detection of crisis situations associated with the accumulation of a large number of people), biometric analysis (enables automatic identification of persons by face images), analysis of vehicle license plates, analysis from several cameras (allows for automatic tracking of the object), automatic detection and classification of objects, search for objects in the video archive database, etc.

An overview analysis of object detection algorithms and object tracking algorithms was carried out, as a result of which a method of classifying object tracking algorithms was proposed. An experimental study of a number of tracking algorithms was also conducted, as a result of which recommendations were formed regarding the conditions of application of the specified algorithms.

In the second chapter "Development of methods and models for face recognition and traffic compression using the background subtraction algorithm for stationary video surveillance systems" an efficient background subtraction algorithm based on ViBE for complex backgrounds is proposed. It combines advanced ViBE, which uses cumulative averaging and a pixel-counting engine for fast de-ghosting, and adaptive ViBE, which calculates an adaptive radius based on background changes.

An algorithm for compressing videos captured by cameras with limited resources is proposed, which uses motion analysis to remove unwanted pixels and image frames.

A face recognition method based on motion detection is proposed. This method not only focuses on the steps of face detection and recognition, but also includes the

detection of moving objects, which reduces the amount of data that needs to be processed.

A software model of data processing in video surveillance systems has been developed. The efficiency of using Spark technology has been proven. This technology ensures fast operation of algorithms even when data volumes increase due to optimization of calculations.

In the third section "Analysis of the developed methods and models of face recognition and traffic compression using the background subtraction algorithm" an analysis of the created background subtraction algorithm based on ViBE for a complex background is performed.

Experiments were carried out on the resistance of the models to the rotation of the face in different planes, changes in the size of the face, and also the time consumption was evaluated. The best models in terms of rotation parameters were RetinaFaceResNet125, DSFD, RetinaFaceMobile Net0.25, which accurately (accuracy  $\geq 0.9$ ) determine faces with rotations in the range  $[-45; 45]$ , which is a sufficient requirement for use in security systems. The MTCNN, FaceBoxes, CenterFace and SCRFD-500MF models also work with multi-plane face rotation, but have lower accuracy and a smaller face angle range.

In experiments with different face sizes, the best results were shown by the MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25 models, which detect images starting at  $75 \times 75$  px with an accuracy of  $\geq 0.9$ . The CenterFace and SCRFD0.5GF models have significantly lower accuracy, but in some cases detect very small face sizes.

Measuring object detection times showed that RetinaFace-MobileNet0.25 and Face Box are the fastest, taking less than 47ms to process a single VGA image and 100ms for HG size. Next up is the SCRFD-500MF, CenterFace. For the MTCNN and RetinaFaceResNet125 models, detection times exceed 100ms even for VGA images. The DSFD model cannot be applied in real time even to VGA images.

An algorithm created for compressing videos captured by cameras with limited resources, which uses motion analysis to remove unwanted pixels and image frames,

is analyzed. Analyzing the video frames pixel by pixel, the algorithm achieved an average compression of 30% of the test data, while preserving all the information important for analyzing the behavior of moving objects. This significant file size reduction significantly expands monitoring capabilities by minimizing bandwidth requirements and facilitating efficient data transfer from resource-constrained devices. Deploying this algorithm to cameras has the potential to significantly advance monitoring and conservation by optimizing the capabilities of existing systems.

The proposed face recognition method based on motion detection is analyzed. This approach has a number of advantages, such as: reducing the amount of data, increasing speed, reducing energy consumption, versatility, and improving performance. Detection of moving objects allows you to process only those parts of the image where changes occur, which significantly reduces the amount of data that needs processing. The system can process images much faster, which makes it more suitable for real-time tasks. Processing less data requires less energy, making this model more energy efficient. The model can be adapted to different environments and lighting conditions. By combining motion detection with face recognition techniques, the model can improve accuracy by using additional motion information that can help eliminate errors associated with noise and complex background images. Reduce the number of false positives by responding only to objects that are actually moving, which minimizes the likelihood of false face recognition.

In the fourth chapter, "Software implementation of the face recognition model based on the background subtraction algorithm", the Python programming language was used together with the OpenCV, TensorFlow, pickle libraries to create the application. With their help, parts of the system were created that can reduce the load on the telecommunications network and increase the effectiveness of face recognition and identification.

In the conclusions of the dissertation, the main results and recommendations resulting from the conducted research are outlined, quantitative evaluations of

performance indicators in the conditions of using the proposed solutions are proposed and characterized.

Appendices to the dissertation include program acts for the implementation of the results of the dissertation work, as well as a list of the author's scientific works and approvals on the topic of the dissertation.

Keywords: intelligent video surveillance system, dynamic object detection, face identification, face recognition, advanced ViBe, CCTV, VSS.

## **LIST OF PUBLICATIONS OF THE APPLICANT BY DISSERTATION TOPIC**

**Publications in which the main scientific results of the dissertation were published:**

1. Klymash M. M., Gordiychuk-bublivska O. V., Mrak V. B., Branytskyi A. V. Research on the effectiveness of big data processing in video surveillance systems // Infocommunication and computer technologies. 2022. No. 1 (01). P. 52–62. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2788-5518)

<https://doi.org/10.36994/2788-5518-2021-01-01-033>

The recipient's personal contribution: processing the results of experimental studies, conducting data analysis in video surveillance systems, discussing the obtained results, preparing the article for publication.

2. Klymash M.M., Mrak V.B., Gordiychuk-Bublivska O.V., Research on the methods of selection of dynamic objects in video sequences // Bulletin "Infocommunication technologies and electronic engineering", No. 1, pp. 63-75 , 2021

<https://doi.org/10.23939/ictee2021.01.063>

Personal contribution of the recipient: elaboration of the results of experimental research on the selection of dynamic objects in video sequences, discussion of the obtained results, preparation of the article for publication.

3. Yunak O. M., Klymash M. M., Shpur O. M., Mrak V. B. Mathematical model of recognition of fractal structures using neural network technology //

Infocommunication Technologies and Electronic Engineering = Infocommunication Technologies and Electronic Engineering. 2023. Vol. 3, No. 1. P. 1–9. (Belongs to professional publications of Ukraine. technical sciences ISSN 2786-4553)

<https://doi.org/10.23939/ictee2023.01.001>

The recipient's personal contribution: participation in research on the recognition of fractal structures using neural network technology, processing of research results, discussion of the obtained results, preparation of the article for publication.

4. Mrak V. B. Face recognition methods in video surveillance systems using machine learning // Infocommunication Technologies and Electronic Engineering = Infocommunication technologies and electronic engineering. 2023. Vol. 3, No. 2. P. 33–42. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2786-4553)

<https://doi.org/10.23939/ictee2023.02.033>

Personal contribution of the winner: research of face recognition methods in video surveillance systems, interpretation of results, writing of an article.

5. Mrak V.B., Klymash M.M., Masyuk A.R., Kolodiy T.B. Methods and indicators of efficiency assessment in the development of intelligent video surveillance systems // Scientific notes of V. I. Vernadskyi Tavri National University. Series: Technical sciences. 2024. Vol. 35 (74), No. 1. P. 252–259. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2663-5941 E-ISSN 2663-595X)

<https://doi.org/10.32782/2663-5941/2024.1.1/38>

The recipient's personal contribution: working out the results of the evaluation of efficiency in the development of intelligent video surveillance systems, participating in the interpretation of the results, writing the article.

6. Mrak V.B., Klymash M.M. Model of face recognition system for stationary video surveillance systems // Measuring and computing equipment in technological processes. 2024. No. 1. P. 68–73. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2219-9365)



<https://doi.org/10.31891/2219-9365-2024-77-9>

Personal contribution of the recipient: face recognition systems for stationary video surveillance systems, implementation of research simulation, writing of the article.

7. Mrak V.B., Klymash M.M., Babynets V.V. Improvement of the method of detection of dynamic objects in video sequences // Measuring and computing equipment in technological processes. 2024. No. 2. P. 195–204. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2219-9365)

<https://doi.org/10.31891/2219-9365-2024-78-22>

Personal contribution of the recipient: research of an improved method of detecting dynamic objects in video sequences, discussion of the obtained results, preparation of the article for publication.

8. Klymash M. M., Pyrig Yu. V., Senyk A. D., Mrak V. B. Research of a context-sensitive cyber security monitoring algorithm based on recurrent neural networks // Infocommunication Technologies and Electronic Engineering . 2024. Vol. 4, No. 1. P. 1–9. (Belongs to the professional publications of Ukraine. technical sciences ISSN 2786-4553)

<https://doi.org/10.23939/ictee2024.01.001>

Personal contribution of the recipient: analysis of the results of the cyber security monitoring study, interpretation of the results, writing of the article.

9. Zhuravel S., Klymash M., Shpur O., Mrak V. Reducing the impact of unstable connections among nodes of wireless IIoT clusters using machine learning methods // Lecture Notes in Electrical Engineering. 2024. Vol. 1198 : Digital ecosystems: interconnecting advanced networks with AI applications. P. 144–159. (Netherlands scientific serial, ISSN: 1876-1100, E-ISSN: 1876-1119)

Personal contribution of the recipient: data research in IIoT using machine learning methods, interpretation of results, writing of the article.

### **Publications certifying the approval of the dissertation materials:**

1. "Algorithms and methods of detecting dynamic objects of intelligent video surveillance systems", V. B. Mrak ., V International Scientific and Practical Conference "Scientific Technologies in Information Communications" NIST'2023 June 1-3, 2023

<http://repositsc.nuczu.edu.ua/bitstream/123456789/18013/1/%D0%9D%D0%86%D0%A1%D0%A2%202023.pdf>.

The recipient's personal contribution: processing the results of research into the methods of detecting dynamic objects of intelligent video surveillance systems, analysis and interpretation of the results, discussion of the obtained results, approbation of materials at the conference

2. Intelligent IoT control system based on hand gesture recognition / D. Balazh, V. Mrak, A. Sydor, V. Andrushchak, B. Rusyn, T. Maksymyuk // Advanced information and communication technologies : proceedings of the 5th IEEE International conference (Lviv, Ukraine, November 21–25, 2023). 2023. P. 102–104.

<https://doi.org/10.1109/AICT61584.2023.10452708>

The recipient's personal contribution: processing the results of data analysis research in IIoT, discussion of the obtained results, approbation of materials at the conference.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ .....	21
ВСТУП.....	23
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМ ТА ТЕХНОЛОГІЙ ВПРОВАДЖЕННЯ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ.....	27
1.1 Аналіз сучасних технологій систем відеоспостереження.....	27
1.2 Класифікація сучасних систем відеоспостереження.....	28
1.3 Проблемні аспекти функціонування сучасних технологій передачі відеоінформації .....	32
1.4 Технології побудови інтелектуальних систем відеоспостереження та методи виділення фону у відеопослідовностях.....	33
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	49
РОЗДІЛ 2. РОЗРОБЛЕННЯ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ТА СТИСКАННЯ ТРАФІКУ З ЗАСТОСУВАННЯМ АЛГОРИТМУ ВІДНІМАННЯ ФОНУ ДЛЯ СТАЦІОНАРНИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ.....	51
2.1 Удосконалення методу виявлення динамічних об'єктів у відео послідовностях .....	51
2.2 Метод стиснення відеопотоку на основі алгоритму віднімання фону для стаціонарних систем відеоспостереження.....	57
2.3 Модель системи розпізнавання обличчя для стаціонарних систем відеоспостереження .....	62
2.4 Дослідження ефективності обробки великих даних в системах відео- нагляду.....	67
ВИСНОВКИ ДО РОЗДІЛУ 2.....	77
РОЗДІЛ 3. ДОСЛІДЖЕННЯ РОЗРОБЛЕНИХ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ТА СТИСКАННЯ ТРАФІКУ ІЗ ЗАСТОСУВАННЯМ МЕТОДУ ВІДНІМАННЯ ФОНУ .....	79

3.1 Дослідження удосконаленого методу виявлення динамічних об'єктів у відео послідовностях .....	79
3.2. Дослідження методів розпізнавання облич з використанням машинного навчання.....	84
3.3 Дослідження методу стиснення відеопотоку на основі алгоритму віднімання фону .....	108
3.4. Дослідження моделі системи розпізнавання обличчя для стаціонарних систем відеоспостереження .....	111
ВИСНОВКИ ДО РОЗДІЛУ 3.....	116
РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛІ РОЗПІЗНАВАННЯ ОБЛИЧ НА ОСНОВІ АЛГОРИТМУ ВІДНІМАННЯ ФОНУ .....	118
4.1. Обґрунтування вибору технології для програмної реалізації моделі .....	118
4.2. Реалізація алгоритму віднімання фону ViVe та його програмна оптимізація.....	121
4.3. Створення клієнт-серверної частини для стискання трафіку .....	126
4.4. Програмна реалізація методів розпізнавання та ідентифікації облич .....	129
ВИСНОВКИ РОЗДІЛУ 4 .....	136
ВИСНОВКИ .....	137
СПИСОК ВИКОРИСАНИХ ДЖЕРЕЛ .....	141
ДОДАТОК А. АКТИ ВИКОРИСТАННЯ ТА ВПРОВАДЖЕННЯ.....	161
ДОДАТОК Б. СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ ТА ВІДОМОСТІ ПРО АПРОБАЦІЮ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЇ.....	166

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ADVISOR – Advanced Digital Video Storage and On-line Retrieval system;

ARGOS – Advanced pRotection of critical buildinGs by Overall anticipating System;

ARGUS-IS – the Autonomous Real-Time Ground Ubiquitous Surveillance Imaging System;

CARETAKER – Content Analysis and REtrieval Technologies to Apply Knowledge Extraction to massive Recording;

CCD – charge coupled device;

CIE – Commission internationale de l'éclairage;

CONNEXIONs – InterCONnected NEXt-Generation Immersive IoT Platform of Crime and Terrorism DetectiON, PredictiON, InvestigatiON, and PreventiON Services;

CTF – Contrast threshold function

CMOS – complementary metal-oxide-semiconductor;

DARPA – The Defense Advanced Research Projects Agency's;

DEPA – Distributed Enhanced Processing Architecture;

DIVA – Deep Intermodal Video Analytics;

DVR – digital video recoder

EWISA – Early Warning for Increased Situational Awareness;

FORENSOR – FOREnsic evidence gathering autonomous sensor;

IARPA – The Intelligence Advanced Research Projects Activity;

IMSK – The Integrated Mobile Security Kit;

IVSS – intelligent video surveillance system;

MOG – mixture of gaussian;

NVR – network video recorder;

ONVIF – Open Network Video Interface Forum;

P5 – Privacy Preserving Perimeter Protection Project;

PRISMATICA – Pro-active Integrated Systems for Security Management by Technological, Institutional and Communication Assistance;

PSNR – peak signal-to-noise ratio;

PTZ – Pan Tilt Zoom;

S3 – Smart Surveillance System;

SAVASA – Standards-based Approach to Video Archive Search and Analysis;

SOBCAH – Surveillance of Borders, Coastlines and Harbors;

SSIM – Structural SIMilarity;

SUBITO – Surveillance of Unattended Baggage and the Identification and Tracking of the Owner;

TTP – Targeting Task Performance;

VANAHEIM – Video/Audio Networked surveillance system enhancement through Human-cEntered adaptive Monitoring;

VCC – Matthews correlation coefficient;

ViBe – Visual Background extractor;

VICTORIA – Video analysis for Investigation of Criminal and TerrORist Activities;

VSS – video surveillance system.

CNN – Convolutional Neural Network

FPN – Feature Pyramid Network

RPN – Regional Proposal Network

NMS – Non-maximum Suppression

LBP – Local Binary Patterns

AP – Average Precision

mAP – Mean Average Precision

PR – Precision Recall

MTCNN – Multi-task Cascaded Convolutional Networks

DSFD – Dual Shot Face Detector

SCRFD – Sample and Computation Redistribution for Efficient Face Detection

ARI – Adjusted Rand Index

AMI – Adjusted Mutual Information

## ВСТУП

**Актуальність теми.** Досягнення у сфері комп'ютерного зору не стоять на місці, так само, як і прогрес у цілому. До існуючих проблем додаються нові, які, у свою чергу, потребують також нових і ефективних рішень. Вони створюються для певної галузі та проблеми, наприклад, глибоке навчання прийшло на зміну класичним методам комп'ютерного зору, бо була потреба у збільшенні точності класифікації, на той час, класичні методи вирішували її не достатньо якісно. Так само можна сказати і про задачу ідентифікації, а саме про детектування та розпізнавання облич. Проблема захисту майна, об'єкту або іншого ресурсу не нова і досі є актуальною. Для її вирішення застосовують різні комплекси та системи безпеки. Розвитку останніх як раз сприяв розвиток систем комп'ютерного зору, оскільки набагато зручніше встановити відеокамери, та у автоматичному режимі аналізувати що відбувається. Наприклад, схожий сценарій реалізований у аеропортах, для підтримки безпеки. Він аналізує потоки людей і ідентифікує їх по базам даних, аналізує емоційний стан людини і багато інших характеристик. Тому розробка методів та засобів розпізнавання рухомих об'єктів в телекомунікаційних системах в реальному масштабі часу є актуальною науково-практичною задачею

**Зв'язок роботи з науковими програмами, планами і темами.** Дисертація виконувалася відповідно до тематичних планів науково-дослідних робіт Національного університету "Львівська політехніка". Зокрема – в межах наукових досліджень, які виконуються відповідно до держбюджетної роботи кафедри телекомунікацій «Розроблення інноваційних методів та засобів розгортання інтелектуальної інформаційної інфраструктури для подвійного використання в умовах цифрової трансформації України, (2023–2025 рр.), № держреєстрації 0123U100232.

**Метою дисертаційної роботи** є розроблення методів та засобів оптимізації трафіку відео потоків і розпізнавання та ідентифікації облич у системах відеоспостереження.

Для досягнення поставленої мети в роботі розв'язано такі задачі:

1. Аналіз методів та технологій побудови сучасних систем відеоспостереження, сучасних підходів щодо побудови інтелектуальних систем відеоспостереження;
2. Дослідження методів виділення фону у відеопослідовностях камер відеоспостереження;
3. Дослідження методів розпізнавання та ідентифікації облич;
4. Розробка методу виділення динамічних об'єктів на відеопослідовностях;
5. Дослідження та удосконалення сучасних методів детектування та розпізнавання на відкритих та власних наборах даних з метою визначення кращи методів в залежності від середовища та умов спостереження;
6. Розроблення методу оптимізації трафіку за рахунок віднімання фону;
7. Розроблення моделі інтелектуальної системи відеоспостереження на основі створених та протестованих методів;

*Об'єкт дослідження* – процеси обробки, зберігання та передачі інформації у системах відеоспостереження.

*Предмет дослідження* – методи та засоби віднімання фону, розпізнавання облич та оптимізації трафіку в системах відеоспостереження.

**Методи дослідження.** Для вирішення поставлених задач застосовувались методи аналізу, синтезу, моделювання, методи теорії ймовірності та математичної статистики, методи теорії експертних систем, методи машинного навчання. Для побудови програмних продуктів застосовано об'єктно-орієнтоване програмування, методи паралельних та розподілених обчислень.

**Наукова новизна одержаних результатів** полягає в тому що:

- *вперше запропоновано* модель розпізнавання обличчя, заснована на виявленні руху у системах відеоспостереження, який зосереджується не лише на етапах виявлення та розпізнавання облич, але включає виявлення рухомих об'єктів, що зменшує обсяг даних, які потрібно обробити. Цей підхід має ряд переваг, таких як: зменшення обсягу даних, підвищення



швидкості прийняття рішення, зниження споживання енергії, універсальність, покращення продуктивності. Виявлення рухомих об'єктів дозволяє обробляти лише ті частини зображення, де відбуваються зміни, що значно зменшує обсяг даних, які потребують обробки;

- *набув подальшого розвитку* метод виявлення динамічних об'єктів на відеопослідовностях, оснований на алгоритмі ViVe, який поєднує в собі вдосконалений ViBE, використовує кумулятивне середнє значення та механізм підрахунку пікселів для швидкого усунення привидів, і адаптивний ViBE, який обчислює адаптивний радіус залежно від зміни фону. Це дозволяє підвищити ефективність виявлення на 6,5% в порівнянні з відомим методами;
- *запропоновано* метод для оптимізації відеопотоків, знятих камерами відеоспостереження з обмеженим ресурсом, який використовує аналіз руху для передачі через телекомунікаційну мережу лише зміни в кадрі і реконструкції повного кадру на приймальній стороні, що дозволяє зменшити навантаження на мережу до 30% в залежності від кількості рухомих об'єктів у кадрі;

**Практичне значення одержаних результатів.** Практична цінність роботи полягає у доведенні отриманих наукових результатів технологій, методик, алгоритмів та програмних продуктів розпізнавання обличчя в системах відеоспостереження. На основі запропонованих методів розроблено модель розпізнавання обличчя, заснована на виявленні руху. Цей метод зосереджується не лише на етапах виявлення та розпізнавання обличчя, але включає виявлення рухомих об'єктів, що зменшує обсяг даних, які потрібно обробити. Розроблено програмну модель, у якій реалізовані усі наукові результати.

**Особистий внесок здобувача.** Основні положення та результати дисертаційної роботи одержані автором самостійно. Особисто здобувачеві належать наступні наукові результати: розроблено методи розпізнавання обличчя у системах відеоспостереження з використанням машинного навчання [6, 8, 10], розроблено показники оцінки ефективності при розробці

інтелектуальних систем відеоспостереження [4-5, 7], розроблено модель системи розпізнавання обличчя для стаціонарних систем відеоспостереження на основі алгоритму віднімання фону [4-7, 11], розроблено метод оптимізації відеопотоку на основі віднімання фону[1-3].

Іншим авторам спільних публікацій належить: постановка експериментальних розробці математичних моделей для аналізу відеопотоків [1-3, 7, 8] (М. Klymash), аналізу великих даних та їх обробки в системах відеонагляду [1, 2] (О. Hordiichuk-Bublivska), дослідження та використання нейронних мереж для розпізнавання об'єктів [3, 11] (О. Yunak, О. Shpur, D. Balazh, А. Sydor, V. Andrushchak, В. Rusyn, Т. Maksymyuk), обговорені результатів дослідження [1, 5, 7, 8] ((А. Branytskyi, А. Masyuk, Т. Kolodiy, V. BABYNETS, А. SENYK, Y. PIRIH), інтеграція методів машинного навчання в дослідженні стабільності з'єднань у кластерних системах [9] (S. Zhuravel).

**Апробація результатів дисертації.** Результати дисертаційної роботи доповідались на конференціях: Наукоємкі технології в інфокомунікаціях : Міжнародна науково-практична конференція НІСТ (2023, Кам'янець-Подільський, Харків), International Conference on Advanced Information and Communication Technologies (AICT) 2023. Також результати доповідались на семінарах кафедри телекомунікацій «Національного університету «Львівська політехніка».

**Публікації.** За результатами виконаних досліджень опубліковано 11 наукових праць, із них 7 статей – у фахових виданнях України, 1 стаття – у науковому періодичному виданні іншої держави, яке включене до науково метричної бази Scopus та Web of Science, 1 стаття у періодичному виданні України, 2 – у збірниках наукових праць конференцій.

**Структура та обсяг дисертації.** Дисертація складається із вступу, чотирьох розділів, висновків, списку використаних джерел із 165 найменувань та додатків. Повний обсяг дисертації складає 169 сторінки, основний зміст викладено на 114 сторінках, де наведено 42 рисунки.

# РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМ ТА ТЕХНОЛОГІЙ ВПРОВАДЖЕННЯ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ

## 1.1 Аналіз сучасних технологій систем відеоспостереження

У наш час системи відеоспостереження широко використовуються у всіх галузях, починаючи від забезпечення безпеки об'єктів і закінчуючи контролем технологічних процесів, біометричною ідентифікацією та інтелектуальним пошуком у відеоархівах. Вони забезпечують передачу інформації з відеокамер на обмежену кількість моніторів або записують цю інформацію [1]. Під інтелектуальними системами відеоспостереження (IVSS) мають на увазі системи з різноманітними функціями відеоаналітики. Основними складовими систем відеоспостереження є:

- пристрої для збору відеосигналу;
- канали передачі відеосигналів від пристроїв для збору відеосигналу до пристроїв для зберігання та обробки відеосигналу;
- пристрої для зберігання та обробки відеосигналу;
- канали передачі даних від пристроїв для зберігання та обробки відеосигналу до пристроїв для відображення відеоінформації;
- пристрої для відображення відеоінформації.

Система відеоспостереження може бути простою, яка складається з однієї камери та монітора, або складною з великою кількістю розподілених камер, телекомунікаційною системою та різними системами зберігання і аналізу зображень. Основними завданнями сучасних систем відеоспостереження є:

- моніторинг та оцінка ситуації на об'єктах;
- виявлення незаконних дій та надзвичайних ситуацій;
- розпізнавання об'єктів та дій;
- ідентифікація об'єктів;
- прогнозування кризових ситуацій;
- зберігання записаної відеоінформації;

- аналіз записаної відеоінформації про події, що відбулися в секторах камер спостереження та інше.

Цей перелік завдань, що ставляться перед системами відеоспостереження, є умовним, оскільки перед кожною системою стоять відповідні завдання, які вона має виконувати ще на етапі проектування. При цьому враховуються такі критерії, як технічні вимоги до системи, опис об'єкта, вимоги безпеки, вплив зовнішніх факторів, умови експлуатації, надійність, технічне обслуговування та можливість модернізації, а також бюджет проекту.

## 1.2 Класифікація сучасних систем відеоспостереження

Сучасні системи відеоспостереження можна класифікувати таким чином:

- Системи відеоспостереження, що базуються на аналоговому обладнанні.
- Системи відеоспостереження, що ґрунтуються на комбінованому обладнанні, яке включає як аналогові, так і цифрові компоненти.
- Системи відеоспостереження, засновані на IP-технологіях для підключення цифрових пристроїв.

Під аналоговою системою відеоспостереження розуміється система, в якій відеосигнал генерується аналоговими камерами та передається за допомогою аналогових каналів зв'язку (див. рис. 1.1).



Рис. 1.1 – Аналогова система відеоспостереження

Основними перевагами використання аналогових систем відеоспостереження є їх невисока вартість при високій якості зображення, простота налаштування, сумісність обладнання різних виробників в межах одного стандарту та відсутність затримок у передачі сигналу. Для створення сучасних аналогових систем відеоспостереження необхідне таке обладнання:

Аналогові камери спостереження (з можливістю цифрової обробки сигналу).

Відеореєстратори:

Комп'ютерні відеореєстратори (PC based DVR) - комп'ютери з встановленою платою відеозахоплення та спеціальним програмним забезпеченням.

Відокремлені відеореєстратори (Stand Alone DVR) - спеціальні пристрої для запису в цифровому форматі аналогового відеопотоку (такі відеореєстратори мають входи лише для підключення аналогових камер).

Блоки живлення для камер.

Передавальне середовище для відеосигналу (зазвичай коаксіальний кабель).

Відеомонітори.

Проте аналогові системи відеоспостереження мають деякі недоліки:

Аналогові системи є локальними, оскільки камери можуть розміщуватися на відстані не більше 500 метрів від відеореєстратора через обмеження характеристик коаксіального кабелю.

Структура аналогової системи обов'язково включає в себе відеореєстратор, жорсткий диск, блоки живлення та кабельні лінії.

Аналогові камери використовують черезрядкову розгортку, що може призводити до розмиття зображення при спостереженні за швидкорухомими об'єктами.

Характеристики аналогових камер відповідають існуючим стандартам, що обмежує їхню роздільну здатність та частоту кадрів.

Цифрові системи відеоспостереження використовують цифрові камери для створення відеосигналу, який передається у цифровій формі (див. рис. 1.2).



Рис. 1.2 – Цифрова система відеоспостереження

Сучасні цифрові системи відеоспостереження включають:

- IP камери відеоспостереження.
- Мережеві відеореєстратори (NVR - network video recorder).
- Мережеве обладнання та середовище передачі даних.
- Спеціалізоване програмне забезпечення.

Переваги використання цифрових систем включають:

- Використання прогресивної розгортки, що дозволяє уникнути спотворень швидкорухомих об'єктів.
- Відсутність обмежень, накладених телевізійними стандартами, що дозволяє отримувати високоякісне зображення з високою роздільною здатністю та частотою кадрів.
- Зручність побудови розподілених систем.
- Можливість будувати системи без використання відеореєстраторів, з записом відео безпосередньо на «хмарний відеореєстратор» чи на вбудовану карту пам'яті.

- Можливість безпроводної передачі даних за допомогою обладнання безпроводних стандартів Wi-Fi, Wi-Max та інших.
- Можливість проведення відеоаналітики за допомогою спеціалізованого програмного забезпечення, яке може обробляти інформацію як на серверному обладнанні, так і безпосередньо на камері.

У випадку використання як цифрових, так і аналогових камер відеоспостереження для формування відеозображення та його передачі через канали зв'язку в аналоговій або цифровій формі, система вважається комбінованою. До складу таких систем можуть входити аналогові та/або цифрові камери, а також PC-based DVR, Stand Alone DVR, NVR (спеціальні мережеві пристрої для запису відеоінформації, що надходить з IP-камер), а також спеціалізовані HDVR (гібридні цифрові відеореєстратори), які підтримують як аналогові, так і IP камери (див. рис. 1.3).

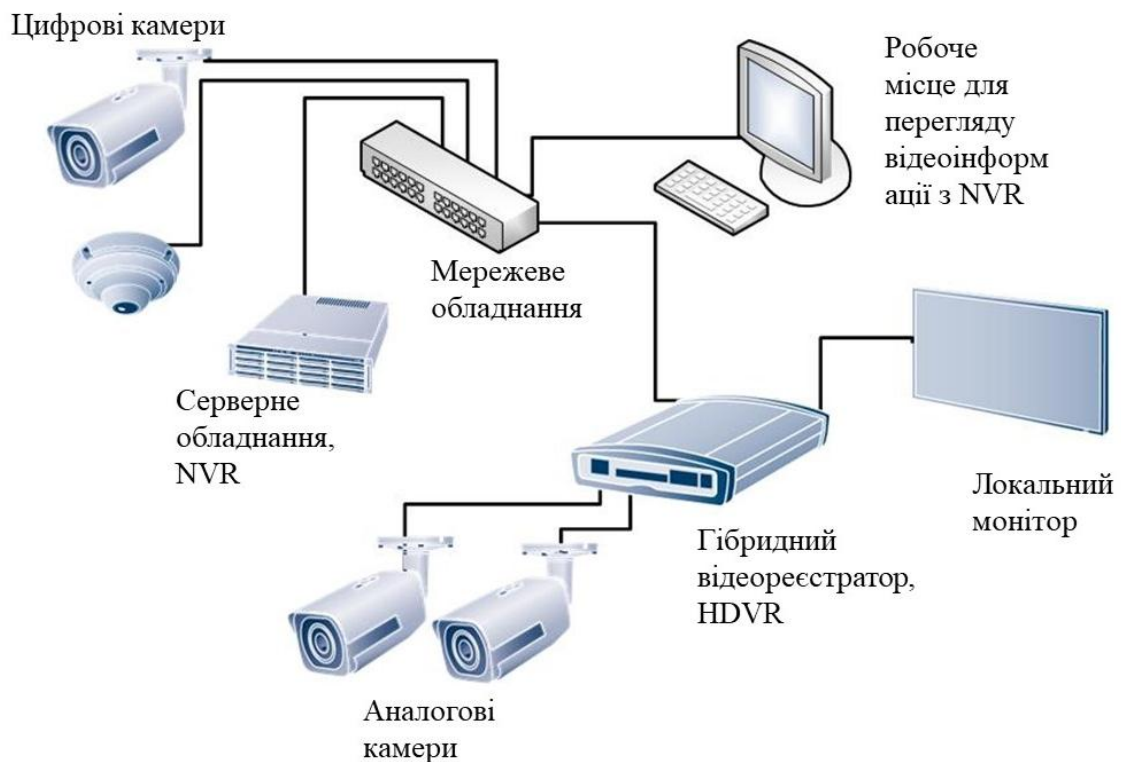


Рис. 1.3 – Комбінована система відеоспостереження

### **1.3 Проблемні аспекти функціонування сучасних технологій передачі відеоінформації**

Проблемні аспекти функціонування сучасних технологій передачі відеоінформації включають наступні складнощі:

Обмежена пропускна спроможність мережі: Більшість сучасних камер мають інтерфейс Ethernet з можливістю роботи зі стеком протоколів TCP/IP. Проте, пропускна спроможність найпоширенішого стандарту Fast Ethernet складає лише 100 Мбіт/с. Це може створювати перешкоди для передачі великого обсягу даних, таких як відео, особливо в умовах високої роздільної здатності та частоти кадрів.

Компресія відеоданих: Оскільки пропускна спроможність мережі обмежена, часто для передачі відеоданих використовується компресія. Проте це може призводити до втрати якості зображення. Зазвичай, в IP-камерах, зображення отримане з ПЗЗ або КМОП матриці, стискається за допомогою методів відеокомпресії, таких як покадрові методи (MJPEG) або потокові (MPEG-4, H.264, H.265) [13].

Деякі цифрові камери мають можливість передавати дані без компресії, що дозволяє зберігати високу якість зображення. Такі камери підтримують різні інтерфейси для передачі даних. HD CCTV та HD-SDI стандарти дозволяють передавати некомпресоване відео по коаксіальному кабелю на відстань до 170 метрів. Вони широко використовуються в системах відеоспостереження для забезпечення високоякісного зображення без втрати деталей. GigE Vision стандарт забезпечує передачу некомпресованого відео в мережі Gigabit Ethernet. Він часто використовується в промислових та наукових застосуваннях, де потрібна висока якість та швидкість передачі даних. USB 2.0 та USB 3.0 інтерфейси використовуються в невеликих за площею системах для передачі некомпресованого відео. Вони забезпечують зручний спосіб підключення камер до комп'ютерів та інших пристроїв. Camera Link інтерфейс також дозволяє передавати відео без компресії та забезпечує високу швидкість передачі даних.



Він часто використовується в вимірjuвальних системах та наукових дослідженнях.

Для забезпечення сумісності обладнання та технологій різних виробників у сфері систем мережевого відеоспостереження було створено ряд промислових стандартів, серед яких особливе місце займає ONVIF (Open Network Video Interface Forum). Заснований у 2008 році компаніями Axis Communications, Bosch Security Systems та Sony, ONVIF створив відкритий стандарт, який сприяє взаємодії різних пристроїв і систем. Специфікація стандарту ONVIF побудована на Web-сервісах, описаних мовою "WSDL", а також використовує протоколи RTP/RTSP, SOAP. Вона також враховує стандарти відеокомпресії, такі як H.264, H.265, MPEG-4, MJPEG та інші. ONVIF визначає протоколи взаємодії для різних пристроїв, включаючи IP камери, кодеки, відеореєстратори, системи управління відео, системи контролю доступу та інші. Стандарт визначає такі особливості взаємодії, як конфігурування мережевого інтерфейсу, налаштування потокової передачі відеоінформації та метаданих відеоаналітики, виявлення пристроїв за допомогою протоколу WS-Discovery, налаштування відеозапису та пошук в архіві, управління профілями роботи камери, керування поворотними механізмами камер (PTZ – Pan Tilt Zoom), передача та обробка подій, відеоаналітика та забезпечення безпеки, включаючи управління доступом та шифрування.

#### **1.4 Технології побудови інтелектуальних систем відеоспостереження та методи виділення фону у відеопослідовностях**

В сучасних цифрових системах відеоспостереження з функціями інтелектуальної обробки інформації, з метою зниження навантаження на мережі передачі даних обчислювальні процеси розподіляють таким чином що частина обчислювальних операцій виконується безпосередньо IP камерою (по причині невисокої продуктивності процесорів IP камер, на них виконуються прості функції відеоаналітики як «Перетин забороненої лінії», «Детектор руху», «Слідування по маршруту» та ін.), а такі процедури як співставлення виявлених

об'єктів з еталонними зразками, розпізнавання символів тощо, виконуються на серверному обладнанні. Для прикладу, компанія «Sony» розробила та запровадила технологію «DEPA» (Distributed Enhanced Processing Architecture), яка передбачає обробку відеоінформації у два етапи:

- на першому етапі (виконується на обладнанні IP камер) об'єкти відділяються від фону, класифікуються як рухомі чи нерухомі, після чого інформація про об'єкти перетворюється в метадані та передається в мережу окремо від цифрового відеопотоку;

- на другому етапі (виконується на серверах) отримана інформація записується на носії інформації, виділяються необхідні об'єкти відповідно до алгоритмів обробки даних, та при співпадінні певних умов виконуються відповідні операції чи то відображення на моніторі чи подача сигналу тривоги чи ін.

Переваги такої технології відображені на рис. 1.4.

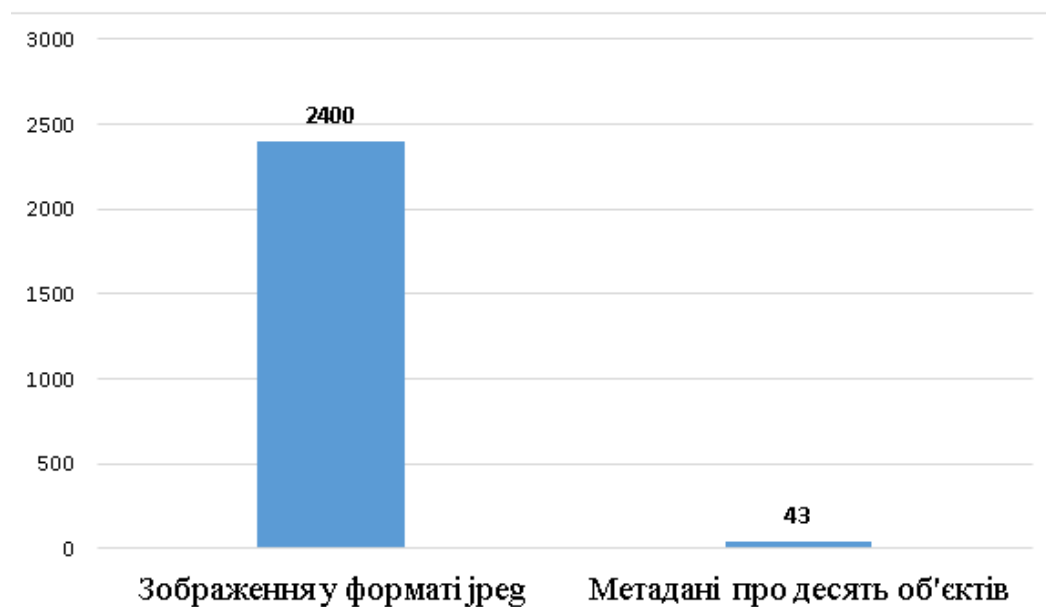


Рис. 1.4. Приклад навантаження на мережу передачі даних без використання та з використанням технології DEPA

Технології, аналогічні DEPA, є найбільш оптимальними при створенні систем відеоспостереження з інтелектуальною обробкою інформації. Однак такі системи повинні включати в себе відеокамери з вбудованими функціями

відеоаналітики. Відомі виробники обладнання для систем відеоспостереження пропонують такі камери за цінами, які значно перевищують вартість звичайних IP-камер відеоспостереження. Крім того, виробники камер відеоспостереження також розробляють власне програмне забезпечення для реалізації функцій інтелектуальної обробки інформації, яке, як правило, підтримує роботу тільки з власними камерами відеоспостереження.

Саме тому при створенні великих систем відеоспостереження рекомендується використовувати технології обробки інформації на серверному обладнанні, що дає можливість використовувати обладнання від різних виробників і забезпечує більшу гнучкість при модернізації систем відеоспостереження [5]. Схему системи відеоспостереження з інтелектуальною обробкою інформації на серверному обладнанні зображено на рис. 1.5.

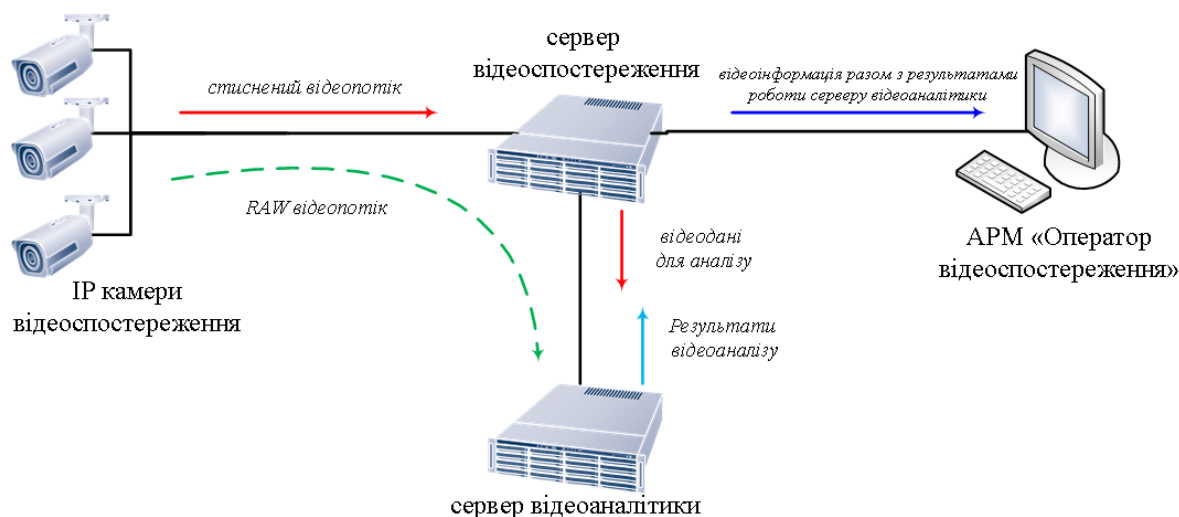


Рис. 1.5. Схема системи відеоспостереження з інтелектуальною обробкою інформації на серверному обладнанні

Розділивши всі завдання на функціональні групи, отримуємо список основних функцій відеоаналітики: виявлення об'єктів, класифікація об'єктів, ідентифікація об'єктів та супроводження об'єктів. Сучасні методи відеоаналітики [4] реалізують ці функції в системах відеоспостереження.

Замовники систем відеоспостереження, безсумнівно, часто бажають отримати максимальний набір функцій. Проте, не завжди є необхідністю

автоматична ідентифікація осіб, які перетинають "заборонену лінію", використовуючи для цього дорогі мегапіксельні камери відеоспостереження та значні обчислювальні ресурси серверного обладнання. Іноді системі досить виявити факт перетину цієї лінії та повідомити оператора. Тому для ефективної побудови системи відеоспостереження важливо збалансувати вимоги замовника та економічні аспекти.

Зазвичай процес автоматизованого інтелектуального аналізу інформації в системах відеоспостереження можна розділити на наступні кроки [4]:

Виділення переднього плану: цей етап спрямований на виділення рухомих об'єктів на зображенні та їх відокремлення від нерухомого фону.

Виділення та класифікація рухомих об'єктів: на цьому етапі рухомі об'єкти сегментуються та класифікуються. Це можуть бути люди, автомобілі, тварини і т.д.

Відслідковування траєкторії руху виявлених об'єктів: ця функція полягає в відслідковуванні руху об'єктів на протязі часу.

Розпізнавання та класифікація дій об'єктів інтересу: на цьому етапі розпізнаються та аналізуються дії виявлених об'єктів, наприклад, виявлення певних поведінкових паттернів.

Для вирішення завдань на кожному з цих етапів використовуються різні методи та методики. Наприклад, для виділення переднього плану використовують методи, що базуються на виділенні фону, ймовірнісні підходи, математичні моделі та інші методи. Для класифікації рухомих об'єктів застосовують геометричні методи, методи, що базуються на динамічних характеристиках та динамічних текстурах. Важливо відзначити, що для вирішення конкретних задач можуть використовуватися комбінації різних методів та алгоритмів.

Загальний процес автоматизованого інтелектуального аналізу інформації в системах відеоспостереження можна представити у вигляді блок-схеми, яка зображена на рис. 1.6.

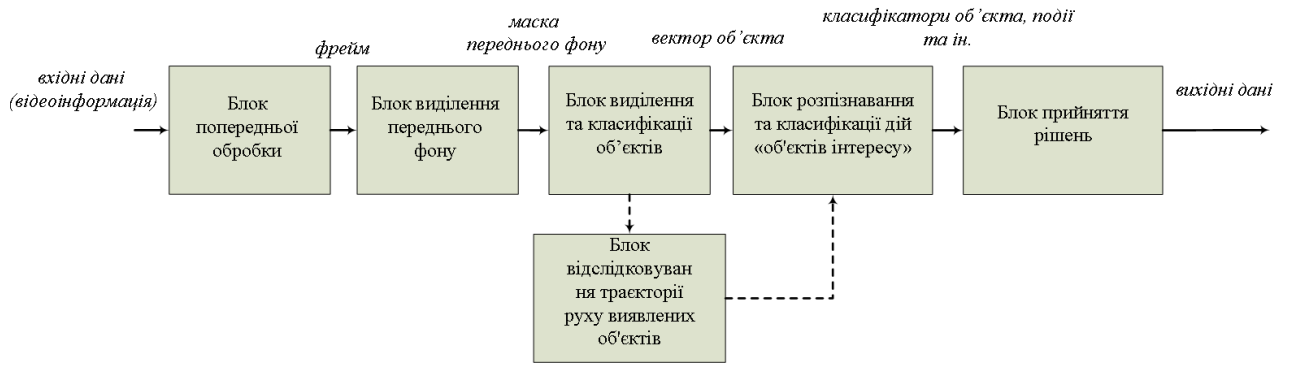


Рис. 1.6. Блок-схема системи відеоспостереження з функціями інтелектуальної обробки інформації

У сучасних системах відеоспостереження з автоматизованою обробкою відеоінформації однією з ключових задач є виявлення рухомих об'єктів, побудова їхніх траєкторій та аналіз цих траєкторій. Один з прикладів використання таких детекторів - це детектор перетину умовної лінії, який застосовується в інтелектуальних системах відеоспостереження. Робота цього детектора полягає в створенні умовної лінії оператором в програмному інтерфейсі. Якщо певні об'єкти (наприклад, особи, транспортні засоби або будь-які інші рухомі об'єкти) перетинають цю лінію, то спостерігач отримує візуальне або акустичне сповіщення [5].

Для досягнення цієї мети потрібно вирішити кілька часткових завдань, таких як виявлення динамічних об'єктів, їх локалізація, супроводження з кадру на кадр та фіксація моменту перетину між "об'єктом інтересу" та "умовною лінією" [5].

Важливим етапом в процесі виявлення рухомих об'єктів у відеопослідовності, отриманій зі стаціонарних (нерухомих) камер відеоспостереження, є виділення фону (background). Загальний підхід полягає у виділенні частини відеокадру, які значно відрізняються від фонові моделі, тобто у створенні маски переднього плану (foreground). Найпростішим методом виділення фону є використання відеокадру, в якому немає рухомих об'єктів, як еталону. Шляхом віднімання фону від наступних відеокадрів можна виявляти

рухомі об'єкти. Однак у реальних умовах існують проблеми, які ускладнюють процес виділення фону [14] (рис. 1.7).

Дослідження щодо вирішення цієї задачі почалися ще у 1990-х роках, і на сьогодні існує велика кількість методів виділення фону. У простому випадку фонові модель обчислюється шляхом віднімання поточного кадру від попереднього статичного зображення, яке не містить рухомих об'єктів. Цей метод називається "кадрова різниця" (Frame Difference). Однак цей метод використовує лише один попередній кадр і не може визначити рух пікселів всередині великого рухомого об'єкта.



Рис. 1.7. Проблемні фактори, що ускладнюють процес виділення фону

Методи виділення фону у відеопослідовностях є ключовими для виявлення рухомих об'єктів в системах відеоспостереження. Один із популярних підходів до цієї задачі - це використання алгоритмів Mixture of Gaussians (MOG), які були запропоновані С. Stauffer і W. E. L. Grimson у їх роботі [7]. У цих алгоритмах, розподіл інтенсивностей кольорів кожного пікселя моделюється як сума нормальних розподілів інтенсивностей випромінювання пікселів протягом певного часу. Пізніше ці алгоритми були вдосконалені Eric Nauman та Jan-Olof Eklundh [15] і отримали назву "Mixture of

Gaussians" (MOG). MOG-алгоритми виявляються ефективними в умовах невеликих коливань освітлення сцени. Однак при різких змінах освітлення або зашумленості кадрів ці алгоритми можуть виявляти помилкові фонові моделі. Щоб вирішити ці проблеми, були розроблені покращені версії MOG-алгоритмів, такі як MOG-2, GMM, GMG, TLGMM, STGMM, SKMGM, TAPPMOG та інші [16,17] та інші.

Існує різноманіття методів та алгоритмів для виділення фону, які можна класифікувати на кілька категорій: [10]

Базові методи та методи, що оперують середніми та дисперсійними значеннями: Ці методи базуються на розрахунках середніх та стандартних відхилень інтенсивності пікселів в часі. Вони добре працюють в стандартних умовах, але менш ефективні при різких змінах освітлення або наявності шуму в кадрах.

*Методи нечіткої логіки:* Ці методи використовують нечіткі множини та правила нечіткої логіки для виділення фону та об'єктів переднього плану.

*Гаусівські процеси:* Ці методи моделюють інтенсивність пікселів як випадкові процеси, які підпорядковуються розподілу Гауса.

*Непараметричні методи:* Ці методи не покладаються на конкретні розподіли і використовують статистичні методи для виділення фону та об'єктів переднього плану.

Використання нейронних мереж та інші: Сучасні методи також включають в себе використання глибокого навчання та нейронних мереж для виділення фону та об'єктів переднього плану.[8]

У роботі [6] була запропонована класифікація методів для виявлення рухомих об'єктів. Ця класифікація розділяє методи на три категорії: методи, що базуються на міжкадровій різниці, методи обчислення оптичного потоку і методи виділення фону.

Методи, які використовують міжкадрову різницю, ґрунтуються на обчисленні різниці між інтенсивністю пікселів на послідовних кадрах з метою виділення областей, де відбувається рух. Проте ці алгоритми можуть бути

неефективними в ситуаціях, коли об'єкти рухаються швидко, наявний динамічний фон (наприклад, хвилі або коливання дерев), або коли об'єкти мають велику площу, що може призвести до помилкового визначення фону як рухомого об'єкта.

Методи, що базуються на обчисленні оптичного потоку, використовують вектори потоку руху об'єктів для визначення рухомих областей на кадрі. Проте ці методи мають обчислювальну складність і можуть бути чутливими до шуму в кадрі.

Найпоширенішим підходом є методи виділення фону, які ґрунтуються на відніманні поточного кадру від попередньо сформованої фонові маски. Цей метод простий з обчислювальної точки зору і підходить для реального часу, але має високу чутливість до динамічного фону і різких змін в кадрі, таких як різкі зміни освітлення або тремтіння камери відеоспостереження. Блок-схему типового алгоритму виділення фону наведено на рис. 1.8.

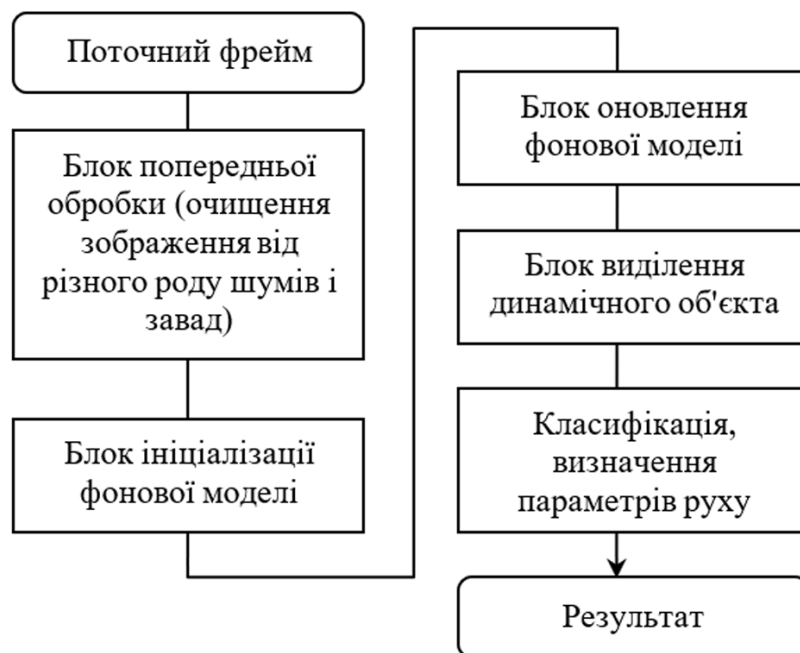


Рис. 1.8. Блок-схема типового алгоритму виявлення рухомих об'єктів основаного на методах виділення фону

Дослідження, спрямовані на вирішення завдання виділення фону у відеопослідовностях, розпочалися ще в 1990-х роках і з тих пір було



запропоновано різні підходи для розв'язання цієї задачі. В найпростішому випадку створення фонові моделі включає в себе обчислення абсолютної різниці між поточним кадром і попередньо визначеним статичним зображенням, яке не містить рухомих об'єктів. Цей метод називається "кадровою різницею" (Frame Difference) і ґрунтується на використанні лише одного попереднього кадру, тому він не здатний визначити рух пікселів всередині великого об'єкта, який рухається рівномірно. Крім того, цей метод виявляє чутливість до таких завад, як тремтіння камери, коливання об'єктів від вітру, рух крон дерев, хвилі на воді і т. д.

С. Stauffer і W. E. L. Grimson [2] запропонували метод, в якому інтенсивність кольорів кожного пікселя представлена як сума нормальних розподілів інтенсивностей випромінювання пікселів. Кожен фоновий піксель моделюється сумішшю ( $k$ ) Гауссівських розподілів, де  $k$  зазвичай варіюється від 3 до 5. Параметри цих сумішей розподілів відображають часові характеристики відображення пікселів. Пізніше цей алгоритм був вдосконалений Eric Nauman і Jan-Olof Eklundh [15] і отримав назву MOG. Цей підхід став популярним завдяки його здатності виділяти фонову модель навіть в умовах невеликих коливань освітлення сцени. Однак при різких змінах освітлення або наявності шуму в кадрі, цей алгоритм може помилково визначати фонову модель. Тому цей тип алгоритмів продовжував покращуватися різними науковцями. Результатом цих досліджень стали удосконалені алгоритми, такі як MOG-2, GMM, GMG, TLGMM, STGMM, SKMGM, TAPPMOG тощо [18]. Наприклад, алгоритм виділення фону MOG2 базується на методі відновлення фону та виявлення рухомих об'єктів і використовує моделі гаусівської суміші. Удосконалення MOG2 полягає у виборі певної кількості Гауссівських розподілів для кожного пікселя, що дозволяє краще адаптуватися до різких змін освітлення. Алгоритм GMG [12] використовує перші  $n$  кадрів для моделювання фону і комбінує методи статистичної оцінки фонові моделі з Баєсівським підходом до сегментації пікселів переднього плану. Для вирішення завдання відстеження динамічних

об'єктів використовуються апроксимація методами Калмана і Гейла-Шеплі. Основною ідеєю методу GMM [36] є моделювання функцій густини розподілу інтенсивностей пікселів протягом певного часового інтервалу та визначення фонових пікселів на основі параметрів цих розподілів. Однак недоліком цього методу є необхідність налаштування порогових значень вручну для кожного типу фону. Якщо фон містить різкі зміни інтенсивності пікселів, то ця модель може помилково класифікувати рухомі об'єкти як фон.

Метод ViVe, представлений у роботах [6-7], відрізняється високою ефективністю обчислень і швидкодією. Він комбінує в собі декілька переваг, таких як висока швидкість виявлення рухомих об'єктів, якість виділення цих об'єктів та низькі обчислювальні витрати.

Проте, не зважаючи на ці переваги, метод ViVe має деякі обмеження. Він є чутливим до різких змін освітлення, іноді недостатньо ефективно пригнічує тіні, які створюються динамічними об'єктами. В оригінальній версії алгоритму ViVe, запропонованій в роботі [3], використовується колірна схема RGB. Проте автори пізніше запропонували використовувати зображення у відтінках сірого з метою підвищення продуктивності методу[7].

За результатами аналізу розміщення камер відеоспостереження на певних об'єктах та секторах їх огляду визначено головні умови, в яких ці камери можуть застосовуватися:

Всередині будівель та зовні на відкритому повітрі, як у закритих, так і відкритих просторах.

У різних погодних умовах, включаючи різні рівні освітлення, навіть у випадках, коли є сильне освітлення або засліплення від джерел світла.

З використанням камер відеоспостереження різних типів та моделей.

При різних температурних умовах.

Умови функціонування цих камер практично не відрізняються від умов застосування на вулицях, в громадських місцях, на автомобільних дорогах та інших об'єктах. Для об'єктивної оцінки характеристик різних методів необхідно проводити порівняльний аналіз за допомогою типових наборів даних.

Для цього були створені спеціальні Інтернет-ресурси, такі як ChangeDetection.net [19], Stuttgart Artificial Background Subtraction (SABS) [7], Wallflower Test Images Sequences, BMC 2012 Background Models Challenge Dataset, SBM-RGBD 2017 Dataset, VSSN 2006 Test Images Sequences, OTCBVS 2006 Test Images Sequences, MAR - Maritime Activity Recognition Dataset, UCSD Background Subtraction Dataset та інші.

Для проведення досліджень необхідно вибрати відеопослідовності, які містять сцени з різними умовами освітлення, включаючи нормальне, недостатнє та з різкими змінами освітлення. Також важливо враховувати погодні умови, такі як хмарність, туман та вітряну погоду, а також виявлення як великих, так і малих рухомих об'єктів. Крім того, необхідно варіювати інтенсивність руху об'єктів для кожної сцени [1].

Поширений метод для оцінки алгоритмів виділення фону [5] полягає у порівняльному аналізі, і він зазвичай використовує такі метрики класифікації для оцінки результатів роботи алгоритмів:

True Positive (TP) - істинно позитивно: кількість пікселів або областей, які алгоритм визначив як рухомі об'єкти і дійсно вони є рухомими об'єктами.

True Negative (TN) - істинно негативно: кількість пікселів або областей, які алгоритм визначив як фон і дійсно вони є фоном.

False Positive (FP) - хибно позитивно: кількість пікселів або областей, які алгоритм помилково визначив як рухомі об'єкти, коли вони насправді є фоном.

False Negative (FN) - хибно негативно: кількість пікселів або областей, які алгоритм помилково визначив як фон, коли вони насправді є рухомими об'єктами.

Ці метрики дозволяють оцінити точність алгоритму в розпізнаванні рухомих об'єктів та фону. Для подальшого аналізу можна використовувати інші метрики, такі як точність (accuracy), чутливість (sensitivity), специфічність (specificity) та інші, які можуть допомогти отримати більш повну картину ефективності алгоритму виділення фону.

Припустимо, що  $A$  – це рішення алгоритму, яка вказує на приналежність пікселя чи групи пікселів до фону ( $A=1$  – рухомий об'єкт,  $A=0$  – фон),  $T$  – це справжня приналежність пікселя чи групи пікселів до фону ( $T=1$  – рухомий об'єкт,  $T=0$  – фон). Таким чином, матрицю рішень класифікатора можна відобразити у вигляді таблиці (табл. 1.1).

Таблиця 1.1

Матриця рішень класифікатора

Справжнє значення Відповідь алгоритму	$T = 1$	$T = 0$
$A = 1$	$TP$ – вірно визначено приналежність до рухомого об'єкта	$FP$ – помилково визначено приналежність до рухомого об'єкта
$A = 0$	$FN$ – помилково визначено приналежність до фону	$TN$ – вірно визначено приналежність до фону

Для оцінки якості роботи алгоритмів використовуються такі метрики як  $precision$ ,  $recall$ ,  $F$ - $міра$ , відсоток помилкових класифікацій ( $ВПК$ ), відношення пікового сигналу до шуму ( $PSNR$ ), індекс структурної подібності ( $SSIM$ ) та  $D$ - $score$ :

$$precision = \frac{TP}{TP + FP}, \quad (1.1)$$

$$recall = \frac{TP}{TP + FN}. \quad (1.2)$$

Показник точності ( $precision$ ) характеризує долю об'єктів (пікселів чи груп пікселів), які вірно класифіковані як фонові, а показник повноти ( $recall$ ) – яку долю об'єктів істинного класу із усіх об'єктів позитивного класу знайшов алгоритм.  $recall$  демонструє здатність алгоритму знаходити даний клас, а  $precision$  – здатність відрізнити цей клас від інших класів.

Метрика  $F$ - $міра$  представляє собою спільну оцінку точності та повноти.

$$F - measure = \frac{recall \times precision}{(\beta^2 \cdot precision) + recall}, \quad (1.3)$$

де  $\beta$  – вага точності в метриці.

Відношення пікового сигналу до шуму ( $PSNR$  – peak signal-to-noise ratio):

$$PSNR = \frac{1}{n} \sum_{i=1}^n 10 \lg \frac{m}{\sum_{j=1}^m \|S_i(j) - G_i(j)\|^2}, \quad (1.4)$$

де  $S_i(j)$  –  $j$ -й піксель  $i$ -го зображення в послідовності  $S$ , яка має довжину  $n$ .

Індекс структурної подібності ( $SSIM$ ):

$$SSIM(S, G) = \frac{1}{n} \sum_{i=1}^n \frac{(2\mu_{S_i} \mu_{G_i} + c_1)(2\text{cov}_{S_i G_i} + c_2)}{(\mu_{S_i}^2 + \mu_{G_i}^2 + c_1)(\sigma_{S_i}^2 + \sigma_{G_i}^2 + c_1)}, \quad (1.5)$$

де  $\mu_{S_i}, \mu_{G_i}$  – середні,  $\sigma_{S_i}^2, \sigma_{G_i}^2$  – дисперсії,  $\text{cov}_{S_i G_i}$  – коваріація  $S_i$  та  $G_i$ ,  $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$ ,  $L$  – динамічний діапазон пікселів,  $k_1 = 0,01$ ,  $k_2 = 0,03$  – константи.

Метрика  $D$ -score характеризує локалізацію помилок відповідно до розміщення об'єкта:

$$D - score(S_i(j)) = \exp(-\ln(2DT(S_i(j)) - \alpha)^2), \quad (1.6)$$

де  $\alpha$  – піковий параметр рівний 2,5,  $DT(S_i(j))$  – дистанція між пікселями  $S_i(j)$  та навколишніми опорними пікселями.

Основна ідея віднімання фону складатиметься лише з різниці між поточним значенням пікселя та значенням фонові моделі з наступним порівнянням цієї різниці з пороговим значенням для прийняття рішення. Однак у реальних життєвих ситуаціях у фоновому режимі виникає велика складність через велику кількість проблем, а саме зміни освітлення, динамічний фон, тремтіння камери та багато інших фонових проблем. Це призводить до розробки більш складних методів віднімання фону.

Методи імовірнісного віднімання фону, які також називають методами параметричного віднімання фону, зазвичай стосуються підходів, які моделюють

фон за допомогою нормального (гаусового) розподілу значень інтенсивності пікселів зображення. Одне з перших рішень було запропоновано Stauffer С [20]. Після цього з'явилося багато вдосконалених методів STGMM[9], SKMGM[10], SKMGM[10], TAPPMOG[11] і STAPPMOG[12] (де MOG означає суміш Гаусса), кожен із яких намагався вирішити один або більше недоліків оригінального алгоритму. Однак через його чутливість MOG не може бути точно налаштований, і його здатність успішно справлятися зі змінами у фоновому режимі є дискусійною.

Щоб вирішити недоліки ручного вибору параметрів у кожному середовищі, були запропоновані непараметричні підходи (Ahmed Elgemma [13]). Зазвичай вони стосуються методів на основі вибірки, але існує багато інших, таких як оцінка щільності ядра (KDE)[14], підхід Байєса[15] або рекурсивна оцінка щільності (RDE)[16].

Інша група методів — метод кодового словника, де для кожної позиції пікселя фон моделюється за допомогою кодового словника [17]. Тут кожному пікселю призначається кодове слово, яке складається з інтенсивності, кольору та часових характеристик. Щоб виконати сегментацію, інтенсивність і колір вхідних пікселів порівнюються з інтенсивністю та кольором кодових слів у кодовому словнику. Пізніше алгоритм обчислює відстані між пікселями та кодовими словами, порівнює їх із пороговим значенням і призначає мітку переднього плану, якщо відповідності не знайдено, або мітку фону в іншому випадку. Пізніше відповідне кодове слово оновлюється відповідним фоновим пікселем.

Підпросторовий підхід [8], в якому одна модель призначена для точного виявлення руху, а інша спрямована на досягнення представлення порожньої сцени. Відмінності у виявленні переднього плану додаткових моделей використовуються для визначення нових статичних областей. У цьому підході обчислюється набір зображень і відповідне середнє значення та коваріаційна матриця. Після цього виконується обчислення PCA коваріаційної матриці, проєкції з певною кількістю векторів і порівняння вхідних зображень з їх

проекціями на власні вектори, обчислюється відстань між зображенням і проекцією та порівнюється з відповідним пороговим значенням, щоб позначити піксель як піксель переднього плану чи фону.

Методи стиснення дозволяють зменшити кількість вимірювань, необхідних для представлення відео, використовуючи попередні знання про розрідженість вихідного сигналу[21], Сигнал може бути реконструйований з невеликого набору випадкових проекцій, за умови, що сигнал є розрідженим у певному базисі. Однак ці методи накладають певні умови на матрицю проектування.

Також важливо згадати методи, які використовують виявлення країв або виявлення патчів [22-23]. У той час як перший метод найчастіше використовується для сегментації зображення на основі різких змін інтенсивності зображення, другий використовує патчі для таких операцій, як зменшення шуму зображення.

Метод розроблений Брахамом і Дрогенбруком, полягає в тому, що використання згорткової нейронної мережі зазвичай відноситься до підходу глибокого навчання. Він використовує модель фіксованого фону, яка була згенерована на основі операції тимчасової медіани над  $N$  відеокадрами [12]. Його архітектура дуже схожа на мережу LeNet-5 для рукописної класифікації цифр [12], за винятком того, що субвибірка виконується з максимальним об'єднанням замість усереднення, а приховані сигмоїдні одиниці замінюються випрямленими лінійними одиницями для швидшого навчання. Однак цей метод, як правило, залежить від конкретної сцени.

Фонова модель оригінальної версії алгоритму ViBE, запропонована у [7], а також його покращення [6], запропоноване тими ж авторами — це непараметрична модель на основі пікселів, де кожен базовий піксель  $x$  фону моделюється набором зразків.

$$M = \{v_1, v_2, v_3 \dots v_N\}, \quad (1.7)$$

Він дотримується тієї ж структури, що й будь-який алгоритм віднімання фону, який має три підзавдання, а саме ініціалізацію фонові моделі, сегментацію або класифікацію та оновлення моделі.

Для кожного пікселя  $x$  набір  $M(x)$  заповнюється випадковим чином вибірками навколо просторової околиці  $x$ . Важливо зазначити, що ці значення беруться в першому кадрі, і робиться припущення, що сусідні пікселі мають подібний часовий розподіл. Тому оцінка фону дійсна, починаючи з другого кадру. Якщо  $t = 0$  індексує перший кадр і що  $N_G(x)$  є просторовою околицею розташування пікселя  $x$ , отже,

$$M^o = \{v^o(y | y \in N_G(x))\}, \quad (1.8)$$

де розташування  $y$  вибрано випадковим чином за єдиним законом.

На практиці ViBE не оцінює жодної функції щільності ймовірності, а використовує набір попередньо спостережених значень вибірки як піксельну модель. Якщо алгоритм хоче класифікувати значення  $v(x)$  пікселя  $x$ , він порівнює це значення з найближчими значеннями серед набору зразків, визначаючи сферу  $S_R(v(x))$  радіуса  $R$  з центром у  $v(x)$ . Якщо набір перетину цієї сфери та набору зразків  $\{v_1, v_2, \dots, v_N\}$  перевищує заданий радіус  $\#min$ , то піксель класифікується як піксель фону. Математично  $\#min$  порівнюється з:

$$\#\{S_R(v(x)) \cap \{v_1, v_2, v_3 \dots v_N\}\}, \quad (1.9)$$

Щоб оновити модель, алгоритм використовує три методи, а саме правила оновлення без пам'яті, випадкову вибірку та просторову дифузію.

Політика оновлення полягає у випадковій вибірці для оновлення. Якщо ймовірність того, що вибірка, присутня в моделі в момент часу  $t$ , збережеться після оновлення піксельної моделі, задана  $\frac{N-1}{N}$ , вона матиме значення

$$P(t, t + dt) = \left(\frac{N-1}{N}\right)^{(t+dt)-t}, \quad (1.10)$$

для будь-якого подальшого часу  $t + dt$ . Це можна представити як

$$P(t, t + dt) = e^{-\ln\left(\frac{N-1}{N}\right)dt}, \quad (1.11)$$



Правило випадкової підвибірki визначає, які пікселі оновлювати. За замовчуванням прийнятий коефіцієнт піддискретизації часу  $\phi$  дорівнює 16, що означає, що кожне значення фонового пікселя має 1 шанс із 16 бути обраним для оновлення своєї піксельної моделі

Просторова дифузія випадковим чином вибирає сусідів або фактичний піксель і оновлює їхній набір зразків відповідно до попередніх принципів. Таким чином забезпечується просторова узгодженість.

Інший механізм складається оновлення моделі «першим увійшов – першим вийшов», наприклад, застосованої в [24, 25], 300 зразків охоплюють часове вікно лише 10 секунд (при 30 кадрах на секунду). Піксель, накритий повільно рухомим об'єктом протягом більше 10 секунд, все одно буде включений у фонову модель. Більш сучасні методи, такі як [30], припускають, що один кадр сам по собі не може розрізнити, чи належать пікселі до переднього плану чи до фону. Тому пропонуються використовувати перші  $n$  кадрів відеопослідовності для завершення ініціалізації моделі фону. В [31], вважають, що для складного динамічного фону радіус  $R$  слід збільшити відповідним чином, щоб фон не можна було легко визначити як передній. З іншого боку, для простого статичного фону  $R$  слід зменшити, щоб виявити невеликі зміни переднього плану.

## **Висновки до розділу 1**

Історично системи відеоспостереження базувалися на телевізійних стандартах. Проте, з розвитком технологій, вимоги до цих систем зростали, і телевізійні стандарти стали обмежувати їх можливості. Це спричинило перехід від аналогових пристроїв та систем до цифрових.

На сьогоднішній день, при створенні систем відеоспостереження, використовуються як аналогові, так і цифрові пристрої, але зберігання відеоінформації здійснюється в цифровому форматі. Сучасні системи відеоспостереження не тільки забезпечують безпеку, але й допомагають керувати процесами на об'єкті.

На основі проведеного дослідження найпоширеніших методів виділення фону у відеопослідовностях було визначено їх ефективність і порівняно за кількісними характеристиками. Дані характеристики використовувались для вибору найкращих методів виділення фону. Результатом став запропонований загальний показник ефективності алгоритму, що застосовується для обробки відеопослідовностей, отриманих зі стаціонарних камер відеоспостереження.

Аналіз існуючих підходів до проектування інтелектуальних систем відеоспостереження показує, що не існує універсальних рішень та архітектур, які підходили б для будь-якої сфери застосування. Тому спочатку необхідно визначити ризики та загрози, на які система інтелектуального відеоспостереження повинна реагувати, а також методи, за допомогою яких вона забезпечить аналіз та сигналізацію.

## **РОЗДІЛ 2. РОЗРОБЛЕННЯ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ТА СТИСКАННЯ ТРАФІКУ З ЗАСТОСУВАННЯМ АЛГОРИТМУ ВІДНІМАННЯ ФОНУ ДЛЯ СТАЦІОНАРНИХ СИСТЕМ ВІДЕОСПОСТЕРЕЖЕННЯ**

### **2.1 Удосконалення методу виявлення динамічних об'єктів у відео послідовностях**

Запропонований метод має три основні модифікації. Першою зміною є введення нового механізму оновлення, який використовує формулу Кеніга для скорочення загальних обчислень. Це поєднується з використанням  $n$  останніх кадрів для моделювання фону. Тут значення  $n$  вибрано емпірично. Друга модифікація полягає у впровадженні механізму підрахунку пікселів для врахування пікселів переднього плану, які довго залишаються пікселями фону. Зазвичай це відбувається, при раптовій зміні фонові моделі. Ці дві модифікації призначені для швидкого усунення феномена привидів. Третя модифікація стосується впровадження модифікованої адаптивної політики радіуса з коригуванням параметрів, яка дозволяє алгоритму працювати зі складним фоном, коли це необхідно, і поводитися по-іншому (майже як оригінальний ViBE), коли фон є відносно простим.

Загальне функціонування оригінального ViBE можна розділити на три основні частини, а саме ініціалізацію фонові моделі, фонову сегментацію та оновлення фонові моделі. Ідея полягає в тому, що замість використання першого кадру для ініціалізації  $N$  значень вибірки кожного пікселя використати  $n$  перших кадрів. Це дає змогу враховувати зв'язок між  $n$  останніми кадрами  $i$ , таким чином, діє як фільтр. Крім того, з'являється механізм підрахунку пікселів і посилює процес, якщо піксель переднього плану залишається на передньому плані протягом  $K$  послідовних кадрів; він, безумовно, став фоновим пікселем, оскільки тепер він є статичним пікселем у сцені

Щоб побачити переваги використання  $n$  останніх кадрів для моделювання фону, потрібно проаналізувати рис. 2.1. На ньому показано різницю в

результатах між використанням одного кадру та використанням  $n$  останніх кадрів у порівнянні з ідеальним результатом очікування. Можна побачити, що краще вибрати  $n$  останніх кадрів, щоб результати були більш точними. Недоліком цього підходу є те, що він потребує трохи більше обчислювальних ресурсів (особливо пам'яті та часу), ніж той, що використовує один кадр, але це прийнятно порівняно з отриманими результатами.

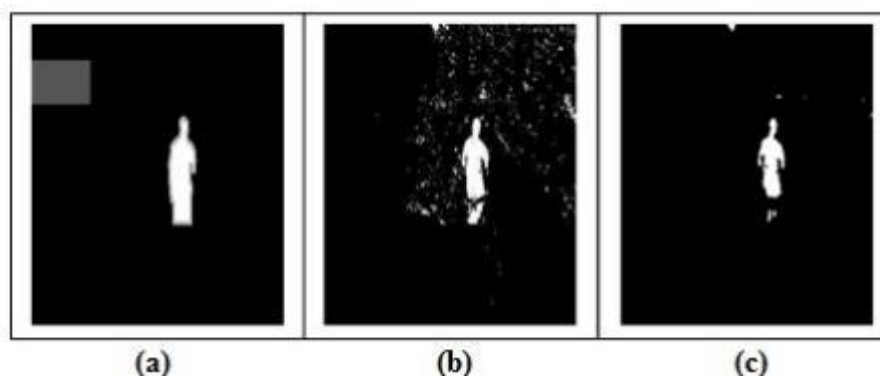


Рис. 2.1. Порівняння результатів набору даних (а) еталон (b) з використанням одного кадру (c) з використанням  $n$  останніх кадрів

З іншого боку, коли деякі зміни відбуваються на задньому плані, як у випадку руху гілок, повітря, дрібних часток або навіть невеликих хвиль на поверхні моря, алгоритм повинен впоратися з цією складністю. Це пов'язано з тим, що ці елементи в русі не належать до переднього плану, тому їх потрібно розглядати як елементи фону. Однак використання радіуса фіксації  $R$  може помилково класифікувати ці пікселі як пікселі переднього плану. Для цього використано адаптивний радіус ( $R_{ad}$ ), який має тенденцію бути сталим, якщо середня відстань між поточним пікселем і значеннями фону його  $N$  вибірки поступово збільшується для складного динамічного фону і має тенденцію до незначного зменшення, якщо середнє значення відстані має тенденцію бути стабільним.

На рис. 2.2 показано різницю в результатах між використанням фіксованого радіуса та адаптивного радіуса в порівнянні з ідеальним результатом очікування.

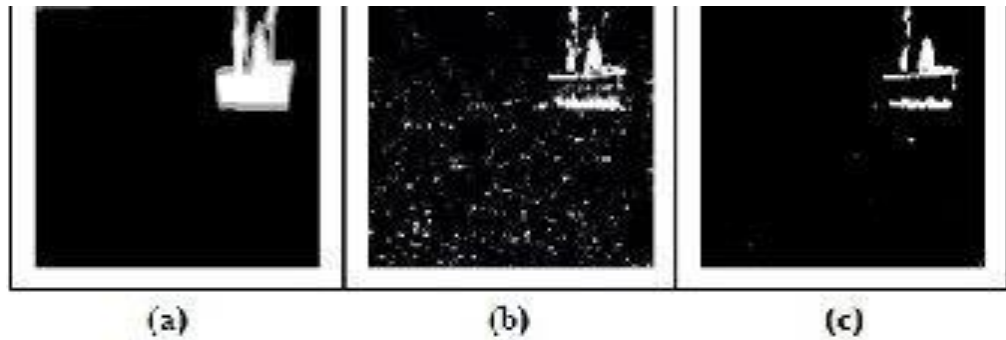


Рис. 2.2. Порівняння результатів на наборі даних човнів (а) еталон (b) використанням фіксованого радіуса (с) з використанням адаптивного радіуса.

Все це дозволило формалізувати весь процес схематично, і він представлений на рис. 2.3

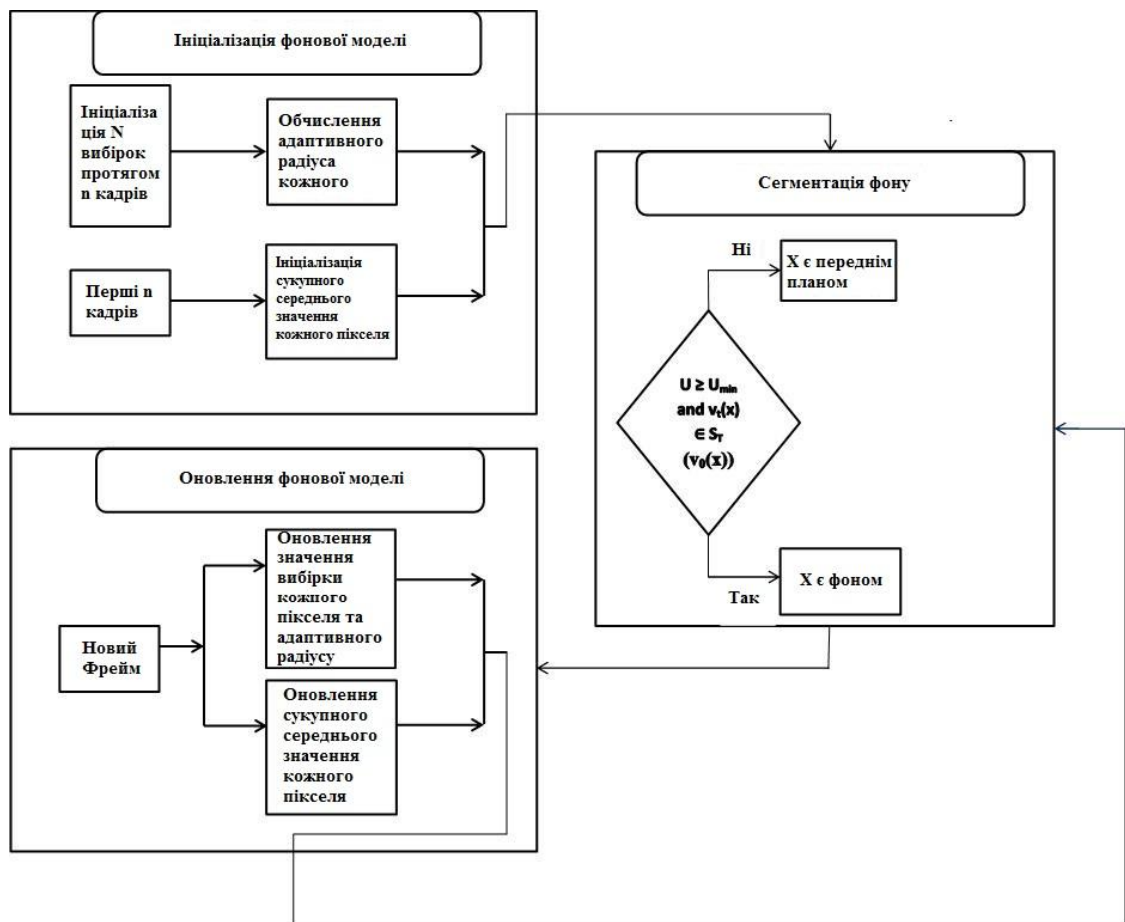


Рис. 2.3. Архітектура функціональної системи

Запропонований метод складається з трьох основних частин, а саме фонові ініціалізації, фонові сегментації та фонового оновлення, як показано на рис. 2.3. Перша частина виконується під час процесу ініціалізації. У той час

як перші  $n$  кадрів збираються для обчислення сукупного середнього для кожного пікселя, вибіркові значення також ініціалізуються. У той же час для кожного пікселя також обчислюється адаптивний радіус  $R_{ad}$ . Друга частина — це частина сегментації, яка класифікує піксель як піксель фону або переднього плану, якщо умови в блоці сегментації на рис. 2.3 перевірені чи ні. Третя частина виконується після процесу сегментації. Він використовує кожен новий вхідний кадр для оновлення своїх параметрів. У цій частині ми намагаємося оновити параметри, які використовуються за допомогою певного підходу. Оскільки обчислення середнього значення та стандартного відхилення (на кожній ітерації) для кожного пікселя є складним, можна використати той факт, що для обчислення нового середнього потрібно лише отримати значення першого пікселя та додати нове значення до попереднього середнього та зробити те саме для стандартного відхилення. Нове середнє буде функцією попереднього середнього, першого значення та нового спостережуваного значення. Нове стандартне відхилення використовуватиме той самий принцип, воно буде функцією попереднього стандартного відхилення, першого значення та нового спостережуваного значення. Крім того, щоб обчислити новий адаптивний радіус, уникається його обчислення (на кожній ітерації), замість цього обчислюється новий радіус тоді і тільки тоді, коли значення відповідного пікселя змінилися. У загальному вигляді, якщо позначити змінну, яка вказує, чи змінилися значення пікселя чи ні, механізм оновлення використовуватиме значення пікселя та значення зміни для обчислення нового радіуса.

Для ініціалізації фонові моделі використовується три підходи. По-перше, використовується процес ініціалізації звичайного методу ViBE для моделювання кожного фонового пікселя за допомогою  $N$  вибірових значень, взятих із їхнього безпосереднього сусідства. По-друге, використовується  $n$  перших кадрів, як у [30], але зі зміною його значення, оскільки збільшення кількості кадрів також підвищує точність результатів. З іншого боку, якщо  $n$  занадто велике, алгоритм стає повільнішим. Тому  $n$  вибране рівним 30 замість 20, як в оригінальному методі. Вибрані кадри використовуються для

обчислення середнього значення кожного пікселя, потрібно додати це значення до значень вибірки, щоб отримати  $N + 1$  значення вибірки. Це призводить до моделювання кожного пікселя

$$M = \{v_0(x), v_1, v_2, v_3 \dots v_N\}, \quad (2.1)$$

де

$$v_0(x) = \frac{1}{n} \sum_{t=1}^n v^t(x), \quad (2.2)$$

По-третє, використовується  $N$  вибірових значень, як у [31], щоб обчислити адаптивний радіус для кожного пікселя. Значення отримані емпіричним шляхом у порівнянні зі значеннями, вибраними в оригінальному методі. Після кількох тестів зауважено, що вихідні значення можна змінити, тобто  $\epsilon_c$ , трохи збільшити, а  $\epsilon_d$  зменшити. Вибрано  $\epsilon_c$  рівним 0,08, а не 0,06, а  $\epsilon_d$  рівним 0,35, а не 0,4. Вирішили позначити адаптивний радіус  $R_{ad}$  як

$$R_{ad} = \begin{cases} R(1 + \epsilon_c), & \text{якщо } R \leq d_{mean} \delta, \\ R(1 + \epsilon_c) \end{cases}, \quad (2.3)$$

$$d_{mean}(x) = \sum_{i=1}^N D^i(x), \quad (2.4)$$

Де  $D^i(x)$  це відстань між пікселем  $x$  і значенням вибірки  $v_i$ .

Якщо ми позначити  $V$  сегментоване зображення,  $V(x)$  буде значенням пікселя  $x$  у матриці  $V$ , яке дорівнює 0, якщо  $x$  є пікселем фону, і 255, якщо  $x$  є пікселем переднього плану. Тому, щоб класифікувати піксель як фоновий піксель або піксель переднього плану, використовуються такі рівняння для:

$$R_{ad} = \begin{cases} 0, & \text{якщо } U_{\min} i v^t(x) \in S_T(v_0(x)), \\ 255, \end{cases}, \quad (2.5)$$

$$U = |S_{R_{ad}}(v^t(x)) \cap \{v_1, v_2, v_3 \dots v_N\}|, \quad (2.6)$$

$$T = \begin{cases} R_{ad}, & \text{якщо } \sigma \leq \beta R_{ad} \\ \sigma / \beta, \end{cases}, \quad (2.7)$$

$$\sigma = \sqrt{\sum_{t=1}^n (v^t(x) - v_0(x))^2}, \quad (2.8)$$

Для параметра  $U_{\min}$  буде встановлено значення 2, як у [5,30,31], а для  $\beta$  буде встановлено значення 3, як у [30]. Крім того, зауважено, що в конкретній ситуації тремтіння камери та температурних умов вибір коефіцієнта 1,5 замість 4,5, як в оригінальній статті, для оновлення останньої історії фонові моделі призводить до кращих результатів

Щоб оновити модель, використано наступні припущення. По-перше, як зазначено в [5], вважається, що сусідні фонові пікселі мають подібний часовий розподіл і що новий фоновий зразок пікселя також повинен оновити моделі сусідніх пікселів. З цієї причини використано ті самі правила оновлення дати без пам'яті, а також вибірки часу. Це дає змогу вибіркочим значенням кожного пікселя адаптуватися до зміни фону.

По-друге, використано механізм підрахунку пікселів, який класифікує піксель, який залишається на передньому плані протягом  $K$  послідовних кадрів, як піксель фону. Для досягнення цієї мети підтримується матриця з розміром кадру. У цій матриці кожен елемент представляє кількість разів, коли піксель з'являвся на передньому плані протягом  $K$  послідовних кадрів. Коли піксель класифікується як піксель переднього плану, алгоритм збільшує його відповідний лічильник у матриці. Якщо лічильник перевищує максимальне значення, позначене  $\text{counter}_{\max}$ , піксель класифікується як піксель фону. Експерименти показали, що значення 10 для  $\text{counter}_{\max}$  призводить до хороших результатів.

Щоб уникнути обчислення параметрів алгоритму після кожного нового кадру, використовується формулу Кеніга, яка справляється краще. Щоб оновити параметри, просто потрібно додати нове значення та відняти попереднє, тому замість того, щоб зберігати середнє значення та стандартні відхилення, зберігається сума та сума квадратів. Це призводить до таких рівнянь:

$$\sum_{t=2}^{n+1} v^t(x) = \sum_{t=1}^n v^t(x) + v^{n+1}(x) - v^1(x), \quad (2.9)$$



$$\sum_{t=2}^{n+1} (v^t(x))^2 = \sum_{t=1}^n (v^t(x))^2 + (v^{n+1}(x))^2 - (v^1(x))^2, \quad (2.10)$$

Якщо позначити через  $v_0^1(x), \sigma^1(x), v_0^{t+1}(x), \sigma^{t+1}(x)$  значення кумулятивного середнього та стандартного відхилення в момент часу  $t$  і  $t + 1$  відповідно, отримаємо їх оновлення за такою формулою:

$$v_0^{t+1}(x) = \frac{1}{n} \sum_{t=2}^{n+1} v^t(x), \quad (2.11)$$

$$\sigma^{t+1} = \sqrt{\left( \sum_{t=2}^{n+1} (v^t(x))^2 \right) - (v^{t+1}(x))^2}, \quad (2.12)$$

Тепер, щоб оновити  $d_{\text{mean}}$ , використано той факт, що нове значення для даного пікселя має бути обчислене, якщо його вибіркові значення були оновлені. Таким чином, використана логічна матриця, де всі елементи ініціалізуються значенням false, і щоразу, коли значення вибірки даного пікселя змінюються, відбувається зміна на значення true для елемента у відповідній позиції в логічній матриці. Кожного разу, коли потрібно обчислити евклідову відстань, щоб переконатися, що вибіркові значення пікселя належать колу радіуса  $R_{\text{ad}}$ , перевіряється, чи його відповідне булеве значення встановлено як істинне. Якщо це так, нове  $d_{\text{mean}}$  обчислюється за допомогою рівняння 9, щоб отримати новий  $R_{\text{ad}}$ . Інакше нічого не робиться, ця методика дозволяє значно скоротити непотрібні обчислення. У процесі прийняття рішення використовуватимуться рівняння 6 – рівняння 13 із заміною  $v_0(x)$  на  $v_0^{t+1}(x)$  та  $\sigma$  на  $\sigma^{t+1}$ .

## **2.2 Метод стиснення відеопотоку на основі алгоритму віднімання фону для стаціонарних систем відеоспостереження**

Зростання попиту на системи відеоспостереження призвело до сплеску досліджень у напрямку розробки розумних систем відеоспостереження, які можуть боротися зі зростаючим рівнем незахищеності. Однак величезна кількість відеоданих, створених цими системами, перевищила можливості

зберігання та обробки аналітичних програм а також навантаження на середовище передачі відеоданих.

Сучасні відеокамери використовують безперервний запис [6, 23, 25], покадрову зйомку [17] або тригери руху [17, 29]. Безперервні відеозаписи потребують великої ємності пам'яті та пропускної здатності передачі. Запуск руху спрямований на те, щоб зменшити вимоги до зберігання та передачі, однак широко використовувані апаратні тригери, такі як датчики PIR, виявилися неефективними для виявлення рухомих об'єктів. Програмні тригери руху також були реалізовані в моніторингу, використовуючи такі методи, як сегментація переднього плану і фону [29] і глибоке навчання [28]. Підходи на основі сегментації переднього плану й фону сприйнятливі до хибнопозитивних виявлень, спричинених рухом листя або змінами освітлення. Для порівняння, тригери глибокого навчання можуть мінімізувати помилкові спрацьовування, записуючи відео лише тоді, коли в кадрі камери присутній комаха. Однак, щоб бути ефективними, ці моделі зазвичай вимагають значних обчислювальних ресурсів, спеціалізованого обладнання та моделей виявлення, навчених на різноманітних об'єктах. Це може обмежити їх цінність для автономних програм і зробити їх схильними до помилкового виявлення.

Пропонується новий ефективний метод, який суттєво зменшує розмір відео без шкоди для даних про рухомі об'єкти. Метод обробляє дані камери за допомогою модифікованого алгоритму ViVe [6], щоб ідентифікувати області зображення з рухом, відправляючи лише інформацію, важливу для моніторингу рухомих об'єктів і реконструкції траєкторії руху та взаємодії з середовищем а також кадр синхронізації приблизно кожні 15 – 25 кадрів для оновлення фонові моделі. Архітектура методу розроблена з урахуванням обмежених ресурсів потужності та пропускної здатності камер відеоспостереження, встановлених у віддалених місцях, одночасно покращуючи пропускну здатність.

Опис архітектури методу і його багатопотоковий підхід, призначений для підвищення пропускної здатності даних показано на рис. 2.4. Запропонований

метод має три основні компоненти: (1) Reader, (2) Motion Analysis і (3) Writer, реалізований за допомогою трьох окремих потоків.



Рис. 2.4. Огляд архітектури алгоритму.

Reader захоплює відеокадри з потоку камери або попередньо записаного відеофайлу та передає їх для обробки алгоритмом ViVe.

Алгоритм ViVe обробляє кадри, захоплені потоком Reader, щоб ідентифікувати області з рухом. Цей процес призначений для отримання лише інформації з відеокадрів, критичної для аналізу поведінки рухомих об'єктів. Він відкидає решту інформації, щоб зменшити навантаження на середовище передавання відеоінформації. Дані, що передаються, включають між кадрову різницю виявлену алгоритмом ViVe та інформацію про їх розташування у кадрі, яка потрібна для відновлення кадру та для алгоритмів подальшої обробки зображення [4].

Аналіз руху починається зі зменшення масштабу кожного захопленого кадру та перетворення його на градації сірого, щоб зменшити обчислювальне навантаження та підвищити ефективність обробки [1, 22]. Згодом зміни між кадрами виявляються шляхом обчислення абсолютної різниці інтенсивності між пікселями в сусідніх кадрах. Піксельні області, що відображають абсолютну різницю інтенсивності, що перевищує встановлене порогове значення, передається разом із навколишньою буферною областю. Порогове значення чутливості захоплення руху регулюється відповідно до середовища застосування. Це дозволяє методу передавати детальну інформацію про рухомі об'єкти, що полегшує подальший аналіз. Кадр, що містить області руху, перетворюється на двійкове зображення, де пікселі ненульової інтенсивності зберігаються, а решта встановлюється на нуль. Кадри без областей виявлення

руху повністю відкидаються. Кадри з областями виявленого руху піддаються масштабуванню двійкового зображення до початкового розміру кадру, і між розширеним двійковим зображенням і вихідним кадром створюється побітовий добуток. Отриманий кадр потім передається до компонента Writer для відновлення та збереження повного кадру. Крім того, алгоритм передає повні кадри через визначені проміжки часу (незалежно від того, чи було виявлено рух у цьому кадрі чи ні), щоб зафіксувати огляд сцени та зміни в навколишньому середовищі, які відбуваються поступово протягом періоду запису.

Разом із обробленими кадрами руху, надісланими до Writer, алгоритм ViVe передає номери кадрів, а також інформацію про те, чи був переданий повний кадр зображення. Ця інформація аіеористовується для подальшої реконструкції кадру та подальшої обробки наприклад знаходження обличчя у зоні де детектовано рух.

Компонент Writer отримує оброблені кадри від модифікованого алгоритму ViVe і збирає їх у відеофайл. Writer витягує частоту кадрів і роздільну здатність вихідного відео з вхідного відеофайлу. Крім того, Writer зберігає файл CSV разом із відеофайлом, що містить додаткові дані, надіслані компонентом ViVe.

Щоб оцінити метод, використано чотири набори даних реального світу. Вони містять кілька відеороликів, що охоплюють ряд типових сценаріїв моніторингу, складність сцени та режими запису (таблиця 2.1 і рис. 2.5).



a)



b)



c)



d)

Рис. 2.5. Зразок відеокадрів, що показує середовище програми в тестовому наборі даних.

У таблиці 2.1 показано «Середовище запису» описує середовище у якому виконувався запис відео (вулиця, приміщення). «Спосіб запису» представляє спосіб захоплення відео: MT = Motion Triggered, TL = Time Lapse, Cont. = Безперервний. «Камера» описує модель камери, яка використовується для запису відео. «Кодек» показує кодек, який використовується для запису відео. «Відеороздільна здатність». відображає роздільну здатність записаних відео. «FPS» показує записану частоту кадрів у кадрах за секунду.

Таблиця 2.1

Інформація про набір тестових відеоданих.

Датасет	Середовище запису	Спосіб запису	Камера	Кодек	Роздільна здатність	FPS
Ролик 1, 2	Вулиця	MT / TL	Hikvision DS-2CD2010F-I	H264	1920, 1080	30
Ролик 3, 4	Приміщення	Cont	Raspberry Pi V3	DIVX	1920, 1080	30
Ролик 1, 2	Вулиця	MT	Hikvision DS-2CD2010F-I	H264	1920, 1080	30
Ролик 3, 4	Приміщення	Cont.	Raspberry Pi V3	H264	1920, 1080	30

Запропоновано метод для стиснення відео, знятих камерами з обмеженим ресурсом, який використовує аналіз руху для видалення небажаних пікселів і

кадрів зображення. Аналізуючи відеокадри піксель за пікселем, метод досяг середнього стиснення тестових даних на 30%, зберігаючи при цьому всю інформацію, важливу для аналізу поведінки рухомих об'єктів. Це значне зменшення розміру файлу значно розширює можливості моніторингу за рахунок мінімізації вимог до пропускну здатності та полегшення ефективної передачі даних із пристроїв з обмеженими ресурсами. Розгортання цього методу на камерах має потенціал для значного просування моніторингу та збереження за рахунок оптимізації можливостей існуючих систем.

### **2.3 Модель системи розпізнавання обличчя для стаціонарних систем відеоспостереження**

Для створення інтелектуальних систем відеоспостереження необхідно виконати певну кількість задач, що стосуються комп'ютерного зору. Одна з основних задач - це виявлення рухомих об'єктів. Зазвичай це відбувається через сегментацію пікселів, що дозволяє відокремлювати пікселі інтересу від інших. Ці пікселі зазвичай називаються переднім планом, тоді як інші - пікселями фону. Це призвело до розробки багатьох підходів вченими, кожен з них намагається подолати виклик у конкретному сценарії, такому як складний фон, високе освітлення, погана якість вхідного зображення. Цей процес називається відніманням або усуненням фону. Запропонований підхід оброблятиме колекцію зображень з відео, які називаються кадрами, і повинен бути виконаний достатньо швидко, щоб його можна було застосувати реальному часі. Серед згаданих операцій існує виявлення та локалізація у випадку обробки конкретного об'єкта. Якщо маємо справу з людьми, виявлення буде включати локалізацію облич у вхідному кадрі, який обробляє весь кадр, щоб знайти положення облич. У великих системах відеоспостереження це створює проблему використання ресурсів, особливо пам'яті та обчислювальної потужності. Ідея полягає в тому, щоб зменшити кількість відеопослідовностей, що підлягають обробці, обмеживши подальші обчислення вхідного кадру лише до області інтересу. У цій статті пропонується підхід, який досягає цих цілей і

показує вплив таких операцій на споживання ресурсів, а також точність результатів.

Алгоритми віднімання фону можна розділити на параметричні та непараметричні. Параметричні (також відомі як ймовірнісні підходи) стосуються методів, які моделюють фон за допомогою нормального (гаусівського) розподілу значень інтенсивності пікселів зображення [1], тоді як непараметричні методи моделюють пікселі за значеннями вибірки та середнім значенням обчислень. Піксель класифікується як належний до фону або до переднього плану. Деякі ймовірнісні підходи включають GMM [2], TLGMM [3], STGMM [4], а непараметричні - [5], байєсівський підхід [6], підхід оцінки щільності ядра (KDE) [23]. Детальнішу інформацію про алгоритми віднімання фону можна знайти в [1,7].

Виявлення рухливих об'єктів передує виявленню або локалізації обличчя. Ці підходи можна класифікувати як методи Viola John [8], які були одними з перших запропонованих. Тут використовуються певні шаблони Adaboost[9], для перевірки наявності або відсутності обличчя на вхідному кадрі. Інший метод на основі кольору шкіри, де [10] для пошуку облич, використовується метод, заснований на кольорі, порівнюючи колір обличчя з фоном, використовуючи той факт, що у людських облич значна колірна розподіленість, яка відрізняється від фонових об'єктів. Інша класифікація - це використання методу опорних векторів SVM [11] з альтернативою, що називається підходом на основі вигляду [12].

методів розпізнавання обличчя настільки багато, що їх можна розглядати як методи на основі вигляду, на основі ознак та штучних нейронних мереж. У методах на основі вигляду зображення представлено з математичним розподілом, щоб здійснити зменшення розмірності. У цьому контексті можна виділити такі методи, як метод Фішера [13], аналіз головних компонент [14] та метод опорних векторів [11]. Підхід на основі ознак, намагається знайти вектор, який зазвичай називається вектором ознак для кожного зразка в базі знань. Деякі підходи, пов'язані з цим, - це модель прихованих марківських моделей

(НММ), активна модель вигляду (ААМ) - двовимірні змінні та тривимірні змінні моделі [15]. Існують підходи на основі штучних нейронних мереж для знаходження інтуїтивного представлення обличчя. Деякі з найвідоміших - це модель FaceNet [16], модель VGG Google [17], модель OpenFace [18] та модель MobileNet [19]. Детальніша інформація про розпізнавання обличчя доступна в [15], [17].

Модель розпізнавання облич на основі видалення фону базується в основному на шести кроках, на відміну від чотирьох кроків у звичайному підході. У звичайному підході перший крок - це попередня обробка вхідного зображення, яка може включати фільтрацію, зміну розміру тощо, за якою слідує алгоритм виявлення обличчя, крок вилучення ознак і завершується розпізнаванням обличчя або ідентифікацією особи рис. 2.6. Крок виявлення служить для виявлення об'єкта, який потрібно аналізувати або ідентифікувати. Це може включати особу, автомобіль, тварину. У випадку цього дослідження основний акцент робиться на виявленні облич, оскільки модель використовується у відеоспостереженні. Таким чином, другий крок буде полягати у виявленні облич за допомогою одного з підходів, описаних у [21] [22]. Заключний крок полягає у ідентифікації виявлених облич. Цей крок також відомий як крок класифікації. У випадку відеоспостереження за особами крок ідентифікації буде полягати у визначенні осіб. Іншими словами, він буде полягати у наданні імені виявленій особі.



Рис. 2.6. Етапи виявлення та розпізнавання обличчя



Для обробки відеопотоків, безумовно, потрібно більше ресурсів, ніж у звичайних випадках. Відповідно, розумним буде пропонувати рішення, які зменшать споживання зазначених ресурсів. У випадку відеоспостереження вважається розумним спочатку виявляти наявність рухомих об'єктів, перед запуском обробки на обрізаному вхідному кадрі, отриманому після сегментації пікселів. Якщо виявлені рухомі об'єкти, відбудеться обмеження зони інтересу, яка в майбутньому сформує нове вхідне зображення. Для цього використовується набір передніх пікселів, виявлених сегментацією, що впливає з ViBE [20]. На цьому етапі є бінарна матриця розміром вхідного кадру, де передні пікселі у білому (255), а фонові пікселі - у чорному (0). Потім обчислюється площа, зайнята цими передніми пікселями, після чого виконується виявлення країв цієї області. Як тільки ця зона буде обмежена, будь-який піксель всередині вважається пікселем інтересу. Таким чином виконується вилучення цієї зони інтересу за допомогою механізму обрізки, який матиме той самий розмір, що і вхідний кадр.

Архітектура моделі розпізнавання обличчя представлена на рис. 2.7 і складається з наступних кроків:

- Крок 1 Попередня обробка: Як у звичайному сценарії, вона включає всі механізми попередньої обробки, які можуть включати отримання зображення, шумопониження тощо.

- Крок 2 Виявлення руху: Це полягає в виявленні рухомих об'єктів, оскільки дослідження проводиться в контексті відеоспостереження, ідея полягає в тому, щоб сфокусуватися тільки на рухомому об'єкті та проігнорувати решту зображення. Використовується підхід віднімання фону, який був запропонований в [20].

- Крок 3 Прийняття рішення: Це полягає в зупинці всього процесу, якщо об'єктів для аналізу немає, і поверненні до кроку 1, що допомагає заощадити обчислювальні ресурси. У випадку виявлення рухомих об'єктів слідує сегментація та вилучення, виконується вилучення зони зображення, що цікавить.

- Крок 4 Виявлення обличчя: Тепер, коли отримано зображення (кадр), алгоритм продовжується, виконуючи виявлення обличчя для пошуку людського обличчя на зображенні. Тут використаний алгоритм CenterFace [22], який складається з каскаду нейронних мереж для ефективного виявлення обличчя.

- Крок 5 Вилучення ознак: З нормалізованого обличчя необхідно за допомогою глибокої нейронної мережі вилучити вектор ознак, які будуть достовірно його описувати для подальшого розпізнавання. Для цього була обрана модель ResNet [23]. Вона була навчена за допомогою функцій втрат ArcFace [24].

- Крок 6 Нарешті, виконується розпізнавання виявленої особи. Проаналізувавши переваги та недоліки методів, доцільно використовувати для невеликої кількості класів SVM, через те, що він описується математично і витрачає менше ресурсів і пам'яті [25].

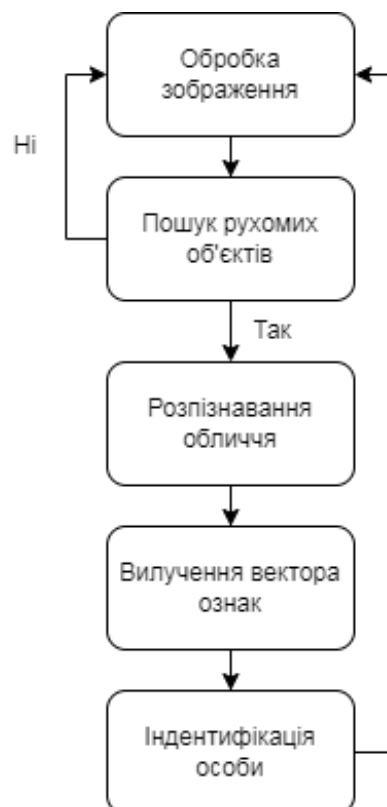


Рис. 2.7. Загальна архітектура запропонованого підходу

## **2.4 Дослідження ефективності обробки великих даних в системах відеонагляду**

Системи відеонагляду мають широке застосування у навчальних закладах, медичних установах, банках, на спортивних заходах та в торгових центрах. Організація ефективного розташування камер і вибір відповідного обладнання для збору відеоданих є ключовими аспектами їх експлуатації.

Після накопичення великої кількості даних у системах відеоспостереження використання сучасних методів обробки інформації набуває особливої важливості. Оскільки інформація зазвичай зберігається обмежений час, інтелектуальний аналіз цих даних стає необхідним. Для досягнення цієї мети використовуються методи обробки та оптимізації великих даних, а також методи машинного навчання.

Сучасне відеоспостереження є необхідним елементом у різних сферах, включаючи вулиці, робочі приміщення, торгові центри та приватні будинки, де воно інсталюється для забезпечення безпеки. Проте системи відеонагляду також використовуються як потенційне джерело аналітичних даних.

У відеоспостереженні передаються значні обсяги даних, що вимагає спеціалізованих методів аналітики для ефективного використання. Використання таких даних дозволяє покращити ефективність функціонування комерційних підприємств, використовуючи онлайн та архівні відео для аналізу та оптимізації бізнес-процесів.

Також важливим є використання даних з систем відеонагляду для збору та аналізу поведінки людей, що є одними з найпоширеніших методів. Це може включати визначення кількості людей та інтенсивності транспортного потоку для подальшої аналітики і прийняття управлінських рішень.

Для того щоб великі компанії могли надавати персоналізовані пропозиції щодо нових товарів своїм клієнтам, вони залучають значні обсяги даних, зокрема з камер відеонагляду. Ці методи дозволяють оперативно отримувати інформацію про клієнтів і майже миттєво робити рекомендації. Наприклад, у магазинах можна відображати на великих екранах статистику про

найпопулярніші товари з різних категорій, яка отримана завдяки аналізу даних з відеокамер.

Для підприємств критично важливе обслуговування клієнтів з максимальною ефективністю. Для досягнення цієї мети вирішальну роль відіграє обробка великих обсягів інформації. Наприклад, за допомогою відеокамер можна ідентифікувати та сповістити клієнта про його місце в черзі за товаром. Цей метод вважається одним з найпростіших. Крім того, можна проводити аналіз покупок клієнта і пропонувати рекомендації щодо нових товарів, які можуть відповідати його інтересам, базуючись на інтелектуальному аналізі зібраної інформації. Також використання систем відеонагляду, які аналізують місцезнаходження покупців і їхні уподобання, дозволяє ефективніше розміщувати стенди з товарами.

Зараз системи відеонагляду швидко впроваджуються в різних галузях. З використанням потужних центрів обробки даних і сучасних високоінтелектуальних методів аналізу інформації можна проводити значно більш складні методи обробки даних. Це дозволяє виявляти складні дії користувачів, розпізнавати різноманітні об'єкти і значно підвищувати якість надання послуг.

Зокрема, системи відеонагляду знаходять застосування на підприємствах для контролю поведінки власних працівників. Це сприяє запобіганню правопорушенням на робочому місці, покращенню ефективності навчання працівників і раціональнішому розподілу робочого часу. Ці досягнення стали можливими завдяки стрімкому розвитку технологій обробки даних з відеокамер.

За допомогою аналізу даних з відеокамер можна ефективно виявляти потенційні загрози для конфіденційності даних користувачів та коштів підприємств. Крім того, інформацію з систем відеонагляду часто аналізують разом з даними з комп'ютерних систем підприємств для формування повної картини стану системи інформаційної безпеки.

На сьогоднішній день існує багато програмного забезпечення для аналізу даних з відеокамер та зручного відображення статистики. Це програмне забезпечення розроблене як для професіоналів у галузі маркетингу та інформаційної безпеки, так і для звичайних користувачів, які хочуть отримувати звіти з системи відеонагляду на своєму телефоні.

Для систем відеонагляду критично важливо забезпечити ефективну структуру мережі та високоякісні компоненти. Для передачі сигналів в таких системах використовуються як провідні, так і безпроводні технології передавання інформації.

Провідне середовище передачі включає в себе мідні та оптичні кабелі. Мідні кабелі часто застосовуються для коротких відстаней і в малих масштабах, таких як внутрішні мережі будівель. Оптичні кабелі є переважним вибором для великих відстаней і вимогливих середовищ, таких як довгі транспортні мережі або великі площі спостереження.

У безпроводних середовищах для передачі даних використовується технологія Wi-Fi. Вона дозволяє зручно і ефективно забезпечити підключення до систем відеонагляду на різних відстанях і в різних областях застосування.

Забезпечення зручної архітектури систем відеонагляду відіграє критичну роль у зборі, передаванні та аналізі даних. Основні аспекти забезпечення ефективної архітектури включають швидке передавання інформації з відеокамер на обчислювальні пристрої, які виконують обробку даних і подальше їх аналіз.

Впровадження потужних програмних засобів для обробки великих потоків даних з відеокамер справді відіграє важливу роль у покращенні ефективності систем відеонагляду. Особливо важливими є алгоритми аналізу даних та методи машинного навчання, які забезпечують автоматизацію процесів обробки і аналізу великих обсягів відеоданих. Ці підходи сприяють значному покращенню якості обслуговування, збільшенню безпеки та оптимізації використання ресурсів у системах відеонагляду. Вони дозволяють

оперативно реагувати на зміни в середовищі та забезпечувати надійність та ефективність в різних умовах експлуатації.

Від належної якості компонентів систем відеонагляду та правильної експлуатації залежить можливість ефективного використання отриманих даних і отримання максимально корисної інформації. При проектуванні таких систем важливо враховувати потребу у постійному збільшенні пропускної здатності каналів передавання даних у зв'язку зі сталим зростанням обсягу трафіку з кожним роком. Крім того, необхідно включити в проект резервні лінії зв'язку та пристрої для обробки даних, щоб забезпечити безперебійну роботу системи у випадку виникнення збоїв в роботі окремих її частин.

На рис. 2.8 представлені основні компоненти системи відеонагляду, а саме:

- відеокамери, призначені для запису відеоданих.
- сервер системи відеонагляду.
- середовища передавання даних.
- кінцеві пристрої, які використовуються для отримання оброблених даних.
- інтернет-мережа, яка забезпечує комунікацію між компонентами системи.

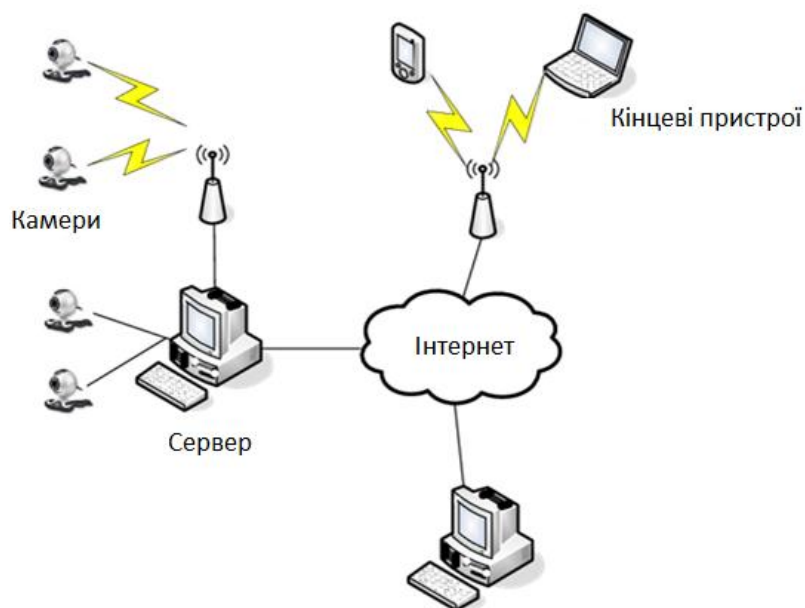


Рис.2.8. Архітектура систем відеоспостереження

Сучасні інформаційні системи потребують фіксації та обробки даних про рухомі об'єкти, часто в реальному часі. Раніше ці завдання виконували спеціалісти, які самостійно аналізували рух в кадрі. Однак сьогодні для цих цілей використовуються спеціалізовані алгоритми обробки даних.

У системах відеонагляду для визначення загроз використовують кілька методів, всі з яких базуються на порівнянні двох сусідніх кадрів і виявленні відмінностей у їх характеристиках. У перших етапах розробки подібних алгоритмів оцінка характеристик виконувалась не для всього кадру, а лише для певної частини. Наприклад, на зображенні визначалися контрольні точки, і зафіксовані зміни яскравості кольору у цих точках на різних кадрах. Якщо ці зміни перевищували певне встановлене порогове значення, вважалося, що відбувся рух.

Інший метод визначення руху за допомогою відеокамер полягає у розподілі кадрів на менші підкадри, які складаються з певної кількості пікселів. Кількість таких підкадрів позначається як  $K$ . На наступному етапі обчислюється середнє значення яскравості в кожному з підкадрів, не враховуючи параметрів сусідніх підкадрів.

Після отримання значень яскравості порівнюють отримані  $K$  результати з тими, що були отримані на попередньому кадрі. Якщо розбіжності перевищують встановлений поріг, це свідчить про фіксацію руху.

Такі методи мають значні переваги, такі як пристосованість до роботи на різних обчислювальних пристроях і простота у проектуванні систем аналізу відеоданих. Тому вони широко застосовувалися в найпростіших системах.

Однак вони мають певні недоліки. Такі методи чутливі до впливу шумів і мають обмежену масштабованість. Вони також вразливі до змін атмосферних умов, освітлення, вологості повітря і т. д., що може призводити до помилок у виявленні об'єктів.

Алгоритм міжкадрової різниці є одним з найпопулярніших сучасних методів обробки відеоданих, який став доступним завдяки зростанню

обчислювальних можливостей пристроїв. Цей метод працює на основі порівняння кожного пікселя між двома кадрами, що дозволяє обчислити так звану "міжкадрову різницю".

Запропонований алгоритм міжкадрової різниці є ефективним і часто використовується в системах безпеки. Ось як він працює:

- обираються два сусідні кадри для порівняння.
- кожен піксель в обох кадрах порівнюється на відповідних позиціях.
- для кожного пікселя обчислюється середнє значення яскравості головних кольорів (наприклад, червоного, зеленого і синього).
- отримане значення порівнюється з попередньо встановленим пороговим значенням.
- на основі результатів порівняння встановлюються двійкові маркери на тих ділянках кадрів, де виявлено рух.
- після завершення роботи алгоритму отримується двійкове зображення, яке відображає місця руху для кожного пікселя у трьох компонентах основних кольорів (RGB).

У методі міжкадрової різниці, як описано вище, якщо різниця значна на певній ділянці між двома кадрами, то відповідний маркер встановлюється в одиницю, що вказує на виявлення руху. Проте в такому методі можуть виникати помилкові визначення різниць для окремих пікселів, наприклад, через шум або зміни освітлення.

Варто зауважити, що не завжди порівнюються два безпосередніх кадри. Іноді використовуються кадри, що знаходяться на певному часовому інтервалі один від одного. Підвищення цього часового інтервалу може дозволити виявити менші зміни між елементами на одному кадрі, що забезпечує точнішу ідентифікацію руху. При порівнянні сусідніх кадрів такі незначні зміни часто ігноруються через вплив шуму.

Щодо методу базових кадрів або віднімання фону, він використовується для розділення пікселів кадру на дві групи: одна група відповідає пікселям, що відносяться до об'єктів переднього плану, а інша - заднього плану. Цей метод



дозволяє виділити об'єкти, що рухаються на тлі, і відрізняється від попереднього методу, який порівнює кадри без обов'язкового поділу на області об'єктів

У розглянутому алгоритмі міжкадрової різниці і встановлення маркерів руху, який базується на порівнянні двох кадрів, що можуть бути сусідніми або мати певний часовий інтервал між ними. Проте можна також використовувати метод, який шукає відмінності між поточним кадром і початковим, де руху не спостерігається, відомим як еталонний кадр. Цей підхід дозволяє більш точно визначити рухомі об'єкти та їхнє поточне розташування. Відповідно до цього підходу використовується метод віднімання фону, який схожий на метод міжкадрової різниці, але він зосереджується на виявленні відмінностей між поточним кадром і вибраним еталонним кадром.

Основною викликом у роботі даного методу є правильний вибір еталонного кадру. Якщо еталонний кадр є реальним, то він має бути якомога ближчим за часом до поточного кадру. У випадку штучного створення еталонного кадру необхідно мінімізувати наявність рухомих об'єктів, щоб уникнути неправильних результатів обробки через те, що деякі об'єкти можуть бути присутні на еталонному кадрі, але відсутні на поточному.

Такі ситуації можуть внести шум у приймач зображення, тому необхідно проводити фільтрацію еталонних кадрів і очищати їх від непотрібних елементів. Це дозволяє забезпечити більш точну і надійну роботу алгоритму.

Для ефективної обробки даних важливо використовувати спеціалізовані методи та технології, які дозволяють оптимізувати процес визначення рухомих об'єктів і зменшити вплив наведених шумів на результати аналізу.

У сфері інформаційних систем для обробки даних використовуються спеціалізовані пристрої, які відповідають за збір, аналіз і подальше надання інформації користувачеві. Проте в сучасних системах відеонагляду спостерігається значний зріст обсягу даних, і навіть найпотужніші пристрої не можуть опрацювати потік відеоданих в реальному часі. (Див. рис. 2.9)

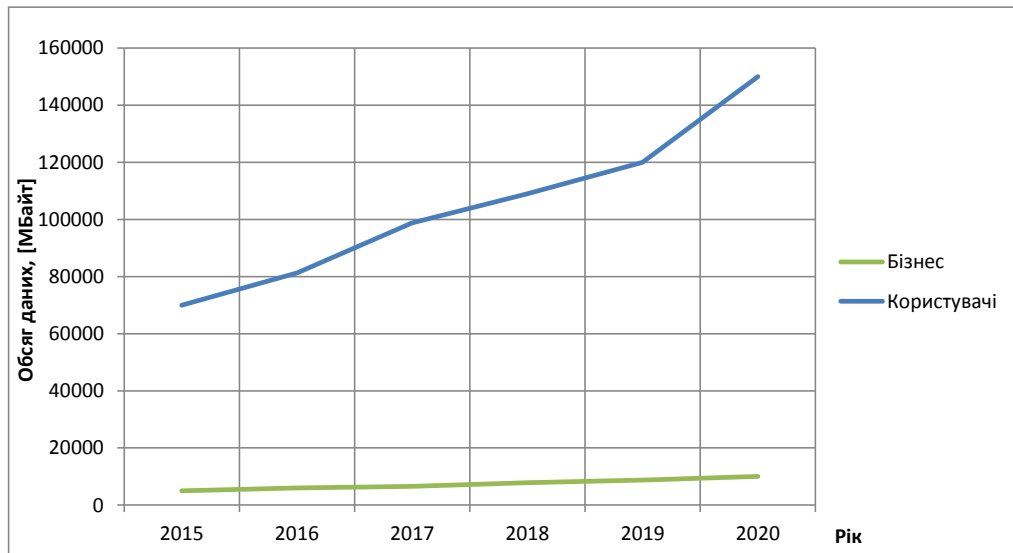


Рис. 2.9. Приріст великих даних в системах відеоспостереження

На рис 2.9 видно стрімкий ріст обсягу даних у системах відеоспостереження в останні роки. Для покращення ефективності обробки даних використовуються розподілені системи. Це кластери обчислювальних пристроїв, між якими розподіляються дані, що потребують обробки. Після цього опрацьовані дані збираються на центральному сервері. Такий підхід значно зменшує час обробки інформації та розподіляє навантаження на обчислювальні пристрої.

У системах відеонагляду використання розподілених систем має особливе значення, оскільки критичним є швидкість надання даних. Інформація, яка обробляється з запізненням, часто втрачає свою актуальність, тому важливо модернізувати системи відеонагляду для скорочення часу обробки.

До недоліків розподілених систем можна віднести складність проектування і необхідність забезпечення координації між пристроями в мережі. Також важливим є забезпечення безпеки інформації, що передається між вузлами системи. У порівнянні з нерозподіленими системами, в розподілених можна застосовувати менш потужні обчислювальні пристрої.

Однією з ключових проблем використання розподілених систем є питання безпеки. Дані не обробляються централізовано, а передаються через мережу на кілька вузлів, де вони обробляються і повертаються назад. Це

створює потребу в застосуванні спеціальних протоколів шифрування для забезпечення безпеки і цілісності інформації.

Розподілені системи мають переваги відмовостійкості, оскільки при відмові одного пристрою обробка даних може продовжуватися на інших, у відміну від нерозподілених систем, де відмова одного обчислювального пристрою призводить до непрацездатності всієї системи.

У розподілених системах використовуються різні технології для обробки інформації, серед яких особливо популярні Hadoop і Spark. Hadoop використовує програмну модель MapReduce, де дані розподіляються між робочими вузлами, оброблюються і передаються головному вузлу для отримання кінцевого результату. Проте процес запису і зчитування в Hadoop триває значний час через роботу з постійною пам'яттю на пристроях.

Технологія Spark, натомість, використовує оперативну пам'ять пристроїв для тимчасового зберігання даних, що дозволяє значно підвищити швидкість обробки даних (в 10-100 разів порівняно з Hadoop). Проте використання великого обсягу оперативної пам'яті є витратним за економічними мірками.

Використання технології Spark виявляє значну ефективність при обробці великих обсягів даних. Тому, для задач, які потребують швидкої і надійної обробки дуже великих масивів інформації, варто вибирати саме цей підхід. Однак для менших потоків даних з меншими вимогами до швидкості обробки можна використовувати менш складні і менш витратні технології розподілених обчислень.

Запропоновано використання технології Spark для обробки даних у системах відеонагляду. Для цього була розроблена програмна модель обробки масивів інформації на мові програмування Python. На Рис. 2.10 зображено залежність тривалості обробки даних різного обсягу з використанням Spark порівняно з нерозподіленими системами.

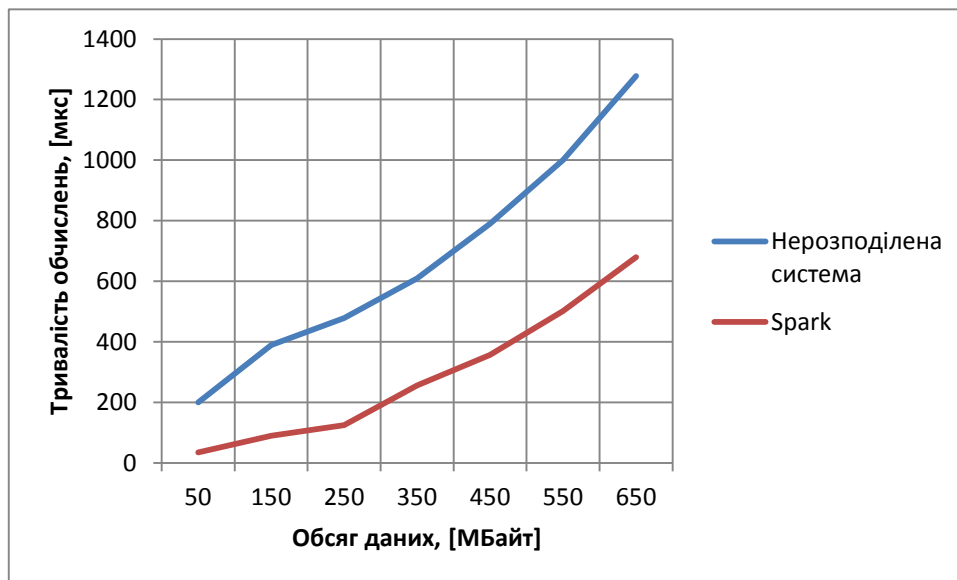


Рис. 2.10. Ефективність обробки даних різного обсягу з та без використання технології Spark

На рис. 2.10 видно, що технологія Spark є більш ефективною для обробки великих обсягів даних у системах відеонагляду. Тому, зі збільшенням обсягів даних, що потребують обробки, і в системах, де необхідно швидко проводити аналіз інформації, використання розподілених систем обробки даних є доцільним. Технологія Spark виявляється особливо ефективною для підвищення продуктивності таких систем і для швидкого аналізу даних, що надходять від відеокамер.

Системи відеонагляду надають значну кількість інформації, яку можна аналізувати як у реальному часі, так і з певним запізненням. Для ефективного аналізу таких даних важливо мати високоякісні відеокамери з високою роздільною здатністю та оптимальною конфігурацією.

В процесі відеонагляду велику роль відіграють методи обробки даних, які мають швидко опрацьовувати інформацію з систем спостереження. Наприклад, вони здійснюють виявлення рухомих об'єктів та визначають кількість людей, що перебувають у кадрі. Такі завдання ускладнюються зміною освітлення та погодних умов. Додатково, при розпізнаванні осіб можуть виникати складнощі через зміну одягу та зовнішності, що вимагає впровадження методів інтелектуального аналізу даних.

Для забезпечення ефективної роботи систем відеонагляду широко використовують розподілені системи, де дані розподіляються між кількома пристроями. Це підходить для швидкого та ефективного аналізу інформації. У дослідженні розроблено програмну модель обробки даних у системах відеонагляду, де ефективність технології Spark підтверджена у практиці. Використання цієї технології забезпечує швидку роботу алгоритмів навіть при великому обсязі даних завдяки оптимальній організації обчислень.

## **Висновки до розділу 2**

У розділі запропоновано ефективний метод віднімання фону на основі ViBE для складного фону. Він поєднує в собі вдосконалений ViBE, який використовує кумулятивне середнє значення та механізм підрахунку пікселів для швидкого усунення привидів, і адаптивний ViBE, який обчислює адаптивний радіус залежно від зміни фону.

Запропоновано метод оптимізації відеопотоків, який базується на алгоритмі віднімання фону. При його роботі передається лише різниця між кадрами та інформації про її розташування у кадрі, за допомогою якої на приймальній стороні відбувається реставрація повного кадру. Це рішення дозволяє зменшити навантаження на телекомунікаційну мережу до 90% в залежності від кількості рухомих об'єктів у кадрі.

Запропоновано метод розпізнавання обличчя, заснована на виявленні руху. Цей метод зосереджується не лише на етапах виявлення та розпізнавання облич, але включає виявлення рухомих об'єктів, що зменшує обсяг даних, які потрібно обробити. Цей підхід має ряд переваг, таких як: зменшення обсягу даних, підвищення швидкості, зниження споживання енергії, універсальність, покращення продуктивності. Виявлення рухомих об'єктів дозволяє обробляти лише ті частини зображення, де відбуваються зміни, що значно зменшує обсяг даних, які потребують обробки. Система може обробляти зображення значно швидше, що робить її більш придатною для задач реального часу. Обробка меншого обсягу даних потребує менше енергії, що робить цю модель більш

енергоефективною. Модель може бути адаптована до різних середовищ та умов освітлення. Завдяки поєднанню виявлення руху з методами розпізнавання облич, модель може підвищити точність за допомогою використання додаткової інформації про рух може допомогти усунути помилки, пов'язані з шумом та складними фоновими зображеннями. Зменшити кількість помилкових спрацювань за рахунок реагування лише на ті об'єкти, які дійсно рухаються, що мінімізує ймовірність помилкового розпізнавання облич.

Розроблена програмна модель обробки даних для систем відеонагляду, де ефективність технології Spark була підтверджена практичними результатами. Використання Spark дозволило значно покращити швидкість обробки даних навіть у випадку збільшення обсягів інформації, завдяки оптимальній організації обчислювальних процесів.

### РОЗДІЛ 3. ДОСЛІДЖЕННЯ РОЗРОБЛЕНИХ МЕТОДІВ ТА МОДЕЛЕЙ РОЗПІЗНАВАННЯ ОБЛИЧ ТА СТИСКАННЯ ТРАФІКУ ІЗ ЗАСТОСУВАННЯМ МЕТОДУ ВІДНІМАННЯ ФОНУ

#### 3.1 Дослідження удосконаленого методу виявлення динамічних об'єктів у відео послідовностях

Реалізація виконана з використанням мови Python/C платформи openCV 3.0[32]. OpenCV використовувався лише для того, щоб скористатися його функціями для захоплення зображень. Набір даних, який використовується для вимірювання продуктивності алгоритмів ChangeDetection.net[33].

Програмна реалізація базується на [34] випуску в липні 2014 року. Також використано BSLibrary[35] для моделювання. Це структура, яка містить до 53 алгоритмів фонового віднімання, які вже реалізовані від найстарішого до найновішого та складного. Тому модифікувавши структуру, додано три нові алгоритми, а саме вдосконалений ViBE, який використовує n останніх кадрів для моделювання фону, адаптивний ViBE, який відповідно змінює радіус, і запропонований ViBE.

Коли серія пікселів виявляється як рухомі цілі, але насправді ці пікселі не є рухомими об'єктами [27], утворюється явище привида. Таким чином, метод повинен швидко усунути його видимість. Однак оригінальний ViBE не видаляє його так швидко, як хотілося б, це один із його головних недоліків. Щоб подолати цю складність, використано кумулятивне середнє, обчислене з використанням n останніх кадрів, щоб посилити класифікацію пікселів.

Механізм підрахунку пікселів, який класифікує як фоновий піксель, який залишається на передньому плані більше ніж  $counter_{max}$  послідовних кадрів. Це дозволяє покращити усунення привидів. Щоб чітко проілюструвати, як запропонований метод усуває привид швидше, ніж оригінальний ViBE, використано набір даних зимового під'їзду, де два автомобілі стоять на місці протягом тривалого періоду часу (понад 1800 кадрів), перш ніж один із них вирішить рухатися. Було зроблено наступні зауваження:

- Оригінальний ViBE, запропонований у 2011 році (див. лінію c на рис. 3.1), потребує більш тривалого періоду часу (понад 600 кадрів), щоб усунути появу привидів

- Для версії, запропонованої в 2014 році тим же автором (див. лінію d на рис. 3.1), усунення феномену привидів набагато краще. Однак точність і якість результатів знизилася.

- Запропонований метод (див. лінію e на рис. 3.1) потребує менше 200 кадрів для виконання того самого завдання. Крім того, виявлений привид швидко видаляється (уже видаляється в кадрі 2100), поки він все ще присутній у кадрі 2100 і кадрі 2200 як в оригінальному, так і в покращеному ViBE.

Як було сказано раніше, для складного динамічного фону радіус слід збільшити, щоб фон не можна було легко визначити як передній план. Це пояснюється тим, що в базовому алгоритмі ViBE зіставлення моделі завжди використовує фіксований глобальний радіус. Тим не менш, у [31] показано, що проста політика радіуса ігнорує складність місцевого середовища та невизначеність змін. Тому важко ефективно виявити ціль у складному середовищі. Тому третій внесок полягає в обчисленні для кожного пікселя модифікованого адаптивного радіуса ( $R_{ad}$ ), який залежить від змінного параметра  $d_{mean}$ , обчисленого за допомогою рівняння (2.4). Він також використовує фіксовані параметри  $\epsilon_d = 0,08$ ,  $\epsilon_d = 0,35$ ,  $\delta = 5$ .



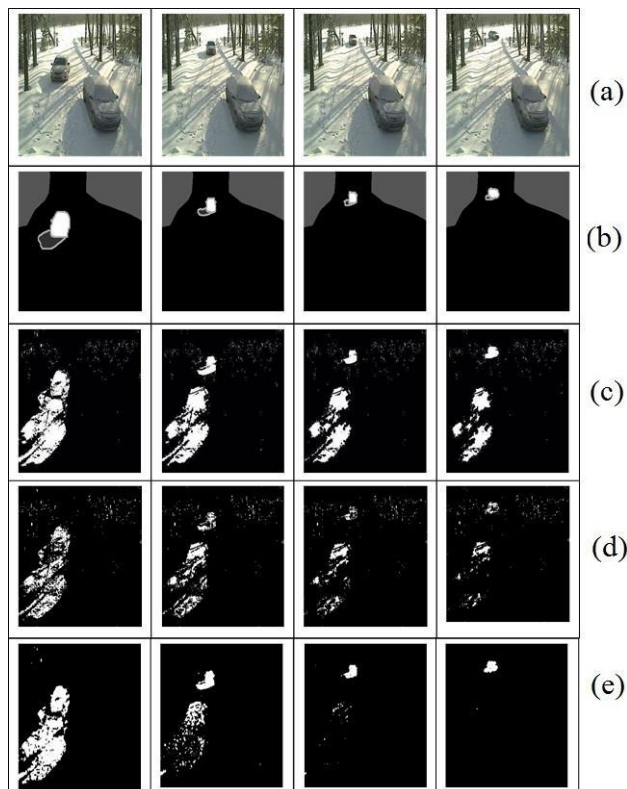


Рис. 3.1. Ціль рухається зі стаціонарного положення (зимовий шлях руху).

(a) 1900-й, 2000-й, 2100-й, 2200-й кадри; (b) еталон, наданий набором даних; (c) результати виявлення оригінального ViBE; (d) результати виявлення покращеної версії ViBE; (e) результати виявлення запропонованого методу.

Щоб проілюструвати ефект адаптивного радіуса, використано набір даних каное, що рухається по морю, з невеликими хвилями на поверхні. Було зроблено наступні зауваження:

- Зазвичай хвилі на поверхні моря відносяться до фону, але ViBE (див. лінію c на рис. 3.2 має тенденцію виявляти його як передній план
- Покращення (2014) намагається вирішити цю проблему (див. рядок d на рис. 3.2, але якість виявленого рухомого човна стає гіршою.
- Запропонований метод вирішує ці проблеми, обчислюючи радіус адаптивним способом, вплив хвилястого фону значно зменшується (див. лінію e на рис. 3.2).

Виміряна точність, відкликання та F-вимірювання дев'яти алгоритмів: оригінальний алгоритм ViBE, виконання ViBE з використанням кумулятивного середнього та механізму підрахунку пікселів, виконання ViBE з використанням

адаптивного радіуса, запропонований метод та деякі складні алгоритми наприклад Codebook[17], KDE[14], MOG[6], PBAS[36] і SUBSENSE[37].

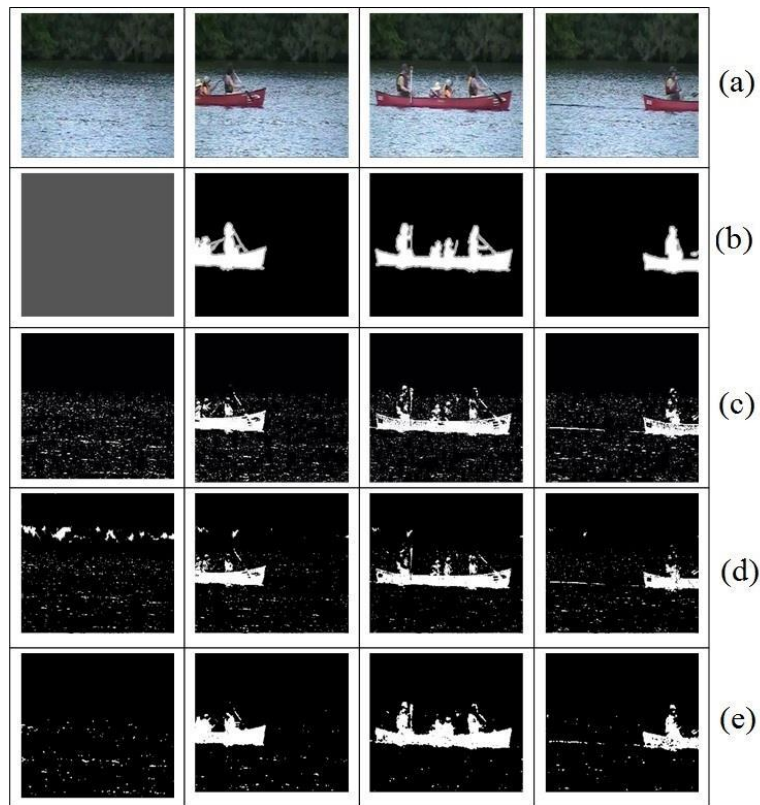


Рис. 3.2. Оцінка адаптивного радіусу (каное). (a) 600-й, 900-й, 960-й, 1040-й кадри; (b) реальний фон (a); (c) результати виявлення оригінального ViBE; (d) результати виявлення вдосконаленої версії ViBE; (e) результати виявлення запропонованого методу

Використані набори даних CDNet 2014, і щоб отримати результати, використано програмне забезпечення BMC-wizard для обчислення цих значень. Процес полягає в наступному: беремо сегментовані зображення, отримані за допомогою вибраного алгоритму, як перший набір параметрів, а другий набір параметрів для майстра BMC є оригінальною основною істинністю вхідних даних, наданих набором даних.

Математично точність  $P_r$ , відкликання  $R$  і  $F$ -measure обчислюються за формулами 2.1, 2.2, 2.3.

Ефективність модифікованого методу з точки зору запам'ятовування порівняно зі згаданими алгоритмами перевершує всі вибрані алгоритми із

середнім відсотком у 65% випадків (тобто серед 53 різних складних фонів). ситуації, запропоновані набором даних), за винятком алгоритму PBAS, де він перевершує лише 30% випадків. Зрештою, об'єднання цих результатів для вимірювання F-міри (також називається F-показником) показує середній відсоток перевищення продуктивності становить 85% для інших алгоритмів і 40% для алгоритму PBAS.

На рис. 3.3 представлено результати, отримані від різних виконання алгоритмів для кадрів, взятих випадковим чином у чотирьох наборах даних, а саме: хуртовина (1), парк (2), диван (3) і шлагбаум (4). Рядок (A) — вхідний кадр, рядок (B) — основне зображення істинності, рядок (C) — виконання алгоритму PBAS, рядок (D) — виконання алгоритму Subsense, рядок (E) — виконання оригінального алгоритму ViBE, рядок (F) ) виконання покращеного алгоритму ViBE з ініціалізацією  $n$  кадрів, виконання Line (G) адаптивного алгоритму ViBE та Line (H) виконання модифікованого алгоритму.

Таким чином, можна побачити, наскільки точними є запропоновані результати підходу порівняно з іншими алгоритмами щодо основного зображення.

Також можна помітити, що, незважаючи на кількісний аналіз, результати PBAS перевершували модифікований метод. Це сталося тому, що PBAS виявляє багато пікселів фону як пікселя переднього плану, що призвело до модифікації параметрів виявлення (TP, FP і FN), і, як наслідок, були отримані хороші кількісні результати.

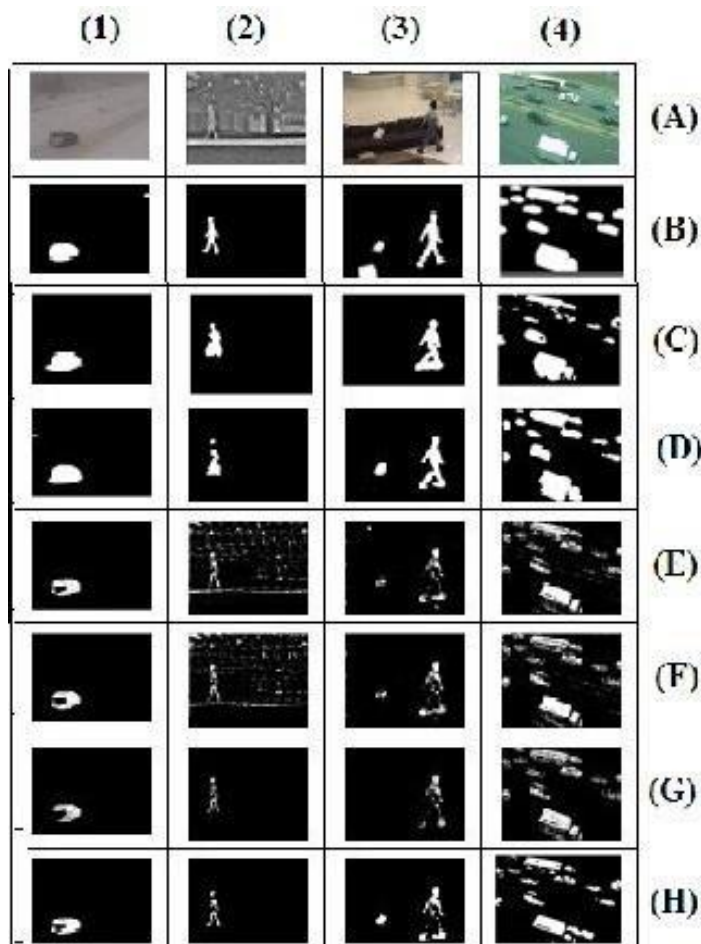


Рис. 3.3. Аналіз результатів

### 3.2. Дослідження методів розпізнавання облич з використанням машинного навчання

Дослідження методів виявлення облич спрямоване на вибір найбільш оптимального алгоритму та архітектури нейронної мережі для системи безпеки підприємства на основі розпізнавання осіб з камер відеоспостереження. Відповідно до вимог системи безпеки особливу увагу було приділено компромісу між швидкістю та надійністю аналізованих методів для їх використання як першого етапу системи безпеки. Після дослідження джерел для аналізу були обрані наступні навчені моделі нейронних мереж (табл. 3.1):

- Багатозадачні каскадні згорткові мережі (MTCNN)
- FaceBox • Подвійний детектор обличчя (DSFD)
- RetinaFace;

- CenterFace;
- Одноступінчастий каскадний детектор залишкового обличчя (SCRFD)

У таблиці 3.1 подано короткий опис досліджуваних моделей. RetinaFace розглядався у двох варіантах з використанням Resnet50 і Mobilenet0.25. Для мережі SCRFD обрано модель SCRFD-500MF, яка потребує 500 мегафлопсів для вхідного зображення з роздільною здатністю VGA (640x480).

Кожна з нейронних мереж має свої характеристики. MTCNN використовує каскад із трьох згорткових нейронних мереж для виявлення облич на зображеннях. Він особливо добре підходить для виявлення облич різного розміру та орієнтації. Головною особливістю моделі FaceBoxes є здатність розпізнавати обличчя з високою швидкістю. Ця здатність моделі робить її ідеальною для додатків, що вимагають високої швидкості обробки, таких як системи безпеки. DSFD — це модель, яка використовує два етапи для розпізнавання обличчя. Вона має дві гілки згорткових нейронних мереж, кожна з яких ідентифікує області з різним масштабом і виявляє обличчя у відповідних областях. Такий підхід дозволяє цій моделі виявляти обличчя різних розмірів та під довільним кутом на зображеннях. Головна характеристика моделі RetinaFaceResNet125 полягає в тому, що вона використовує дві згорткові нейронні мережі: RetinaFace для виявлення облич і ResNet125 для виділення ознак. Таким чином, вона дозволяє виявляти обличчя з високою точністю та визначає більш детальну інформацію про обличчя на зображеннях. RetinaFaceMobileNet0.25 — це модель, подібна до RetinaFaceResNet125, але замість ResNet125 вона використовує простішу мережу MobileNet0.25, що робить її більш придатною для використання на мобільних пристроях [4]. CenterFace — це модель, яка використовує центральну точку обличчя для визначення його положення на зображенні. Вона здатна виявляти маленькі та повернуті обличчя на зображеннях. SCRFD — це модель, яка використовує кілька згорткових шарів для виявлення облич на зображеннях. Вона має високу швидкість обробки та може виявляти обличчя різного розміру на зображеннях.

## Коротка інформація про моделі виявлення

Назва методу	Рік створення	Наявність орієнтирів	СТ у наборі перевірки для WIDER FACE (Hard)
MTCNN	2016	+	0,598[1]
FaceBoxes	2018	-	0,395 [1]
DSFD	2018	-	0,904 [1]
RetinaFaceResNet125	2019	+	0,918 [1,5]
RetinaFaceMobileNet0.25	2019	+	0,78 [5]
CenterFace	2020	+	0,875 [1]
SCRFD-500MF	2021	-	0,685 [2,3]

Середня точність (СТ) часто використовується для порівняння методів виявлення. СТ — це площа під кривою точності запам'ятовування, яка може бути розрахована як наближення кривої точності запам'ятовування з прямокутниками.

У цьому дослідженні СТ не розраховувалася, а була взята із публікацій і репозиторіїв, де моделі знаходяться у відкритому доступі: для MTCNN, FaceBoxes, DSFD, CenterFace; для RetinaFace; для SCRFD-500MF. Для всіх моделей СТ розрахована за набором даних перевірки WIDER FACE (Hard).

Усі ці мережі можуть працювати з обличчями різного розміру та змінними кутами. Дослідження, які б описували діапазон значень масштабного коефіцієнта та кутів повороту обличчя, в межах яких методи мають заявлену точність.

Крім того, швидкість моделей, представлених в існуючих джерелах, була виміряна на різних пристроях і для різних розмірів зображення, що унеможливило порівняння часових витрат цих моделей. Наприклад, у швидкість методу MTCNN становила 16 FPS (2,60 ГГц CPU) і 99 FPS (Nvidia Titan Black GPU); швидкість FaceBox 20 FPS (Intel Xeon E5-2660v3@2.60GHz) [12]; модель RetinaFaceResNet125 становила 13 FPS для зображень VGA (640 × 480 пікселів), RetinaFaceMobileNet0.25 – 40 FPS (зображення 4K (4096 × 2160 пікселів) на графічному процесорі NVIDIA Tesla P40), 20 FPS (HD-зображення (1920 × 1080 пікселів)) на процесорі Intel i7-6700K), 16 FPS (зображення VGA на ARM-RK3399) [4]. У роботі [5] наведено порівняння швидкості роботи

моделей DSFD та CenterFace на NVIDIA GTX2080TI, де отримано такі результати: 78,08 мс та 5,51 мс для зображень із роздільною здатністю VGA відповідно. Також показано, що модель CenterFace демонструє швидкість 30 FPS на CPU I7-6700@2.6 із найбільшою точністю. У [6] визначають, що SCRFD0.5GF перевершує RetinaFaceMobileNet0.25 на 45,57% у часі. Метод DSFD має дуже низьку швидкість, яка не може використовуватися без GPU.

Наведена інформація дозволяє зробити висновок, що модель SCRFD-0.5GF можна назвати найшвидшою з розглянутих моделей, а DSFD, швидше за все, найповільнішою. Проте питання швидкості потребує подальшого дослідження. Необхідно вимірювати швидкість для моделей MTCNN, FaceBoxes, DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0.25, CenterFace і SCRFD-500MF на одному пристрої для зображень однакового розміру.

Оскільки метою даного дослідження є вибір найефективнішого методу розпізнавання, який буде використовуватися на першому етапі системи розпізнавання осіб, то його результат суттєво вплине на кінцевий результат розпізнавання, в зв'язку з цим в роботі досліджуватися наступні параметри методів виявлення:

- стійкість до поворотів у різних площинах (кут зміни навколо вертикальної та горизонтальної осей від -90 до 90 градусів);
- стійкість до спотворень геометричного масштабу (зміна розміру зображень обличчя від 20×20 до 310×310);
- витрати часу (для визначення обличчя у двох розмірах 640×480 (VGA) пікселів та 1280×720 (HD) пікселів).

В експериментах використано нейронні моделі з загальнодоступних репозиторіїв: MTCNN, FaceBoxes, DSFD, RetinaFace, CenterFace, SCRFD-500M. Значення точності ідентифікації обличч отримані цими моделями використовувалися для оцінки якості виявлення.

Експерименти проводилися на наборах даних, створених за допомогою редакторів Generated і Character Creator.

Часові затрати вимірювалося на відеокарті NVIDIA GeForce GTX 780 на шині PciExpress 2.0.

Для порівняння методів виявлення обличчя при різних кутах повороту голови людини створено власний набір даних, у якому людські обличчя були згенеровані за допомогою Generated Photos. На основі цих зображень створені 3D-моделі. Приклади зображень зі згенерованого набору даних показані на рис. 3.4. Ця 3D-модель була відрендерена в графічному редакторі Character Creator з кроком обертання на один градус по двох осях окремо зліва направо від -90 градусів до +90 градусів (обертання навколо осі X) і кут підйому знизу вгору від -90 градусів до +90 градусів (обертання навколо осі Y). Це означає, що для кожної 3D-моделі було створено 181 зображення. На рис. 3.5 показано приклади зображень із синтезованого набору даних отриманих 3D-моделей.

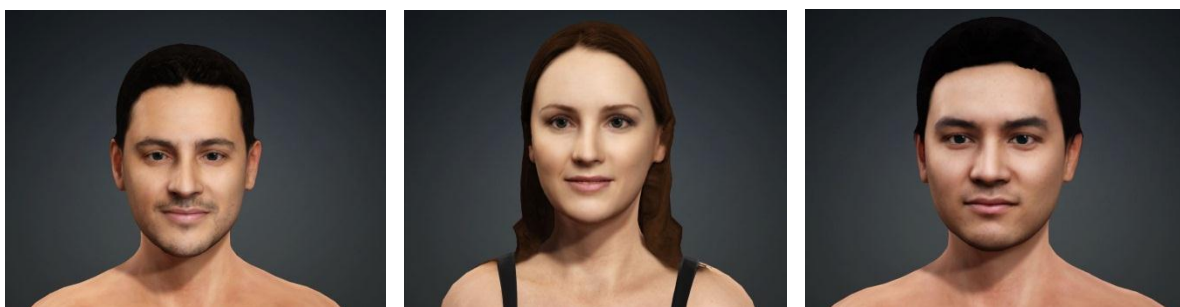


Рис. 3.4. Приклади зображень, створених за допомогою редактора Generated Photos

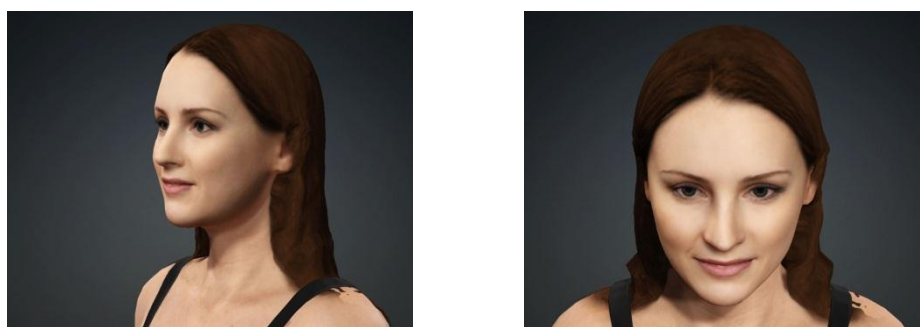


Рис. 3.5. Приклади зображень зі штучного набору даних, створених за допомогою редактора Character Creator: поворот ліворуч навколо осі Y на -50 градусів (рух вгору-вниз); вправо - поворот навколо осі X на -20 градусів (рух вліво-вправо)



На рис. 3.6 і 3.7 наведені значення середньої точності у випадку обертання навколо осі Y та X відповідно.

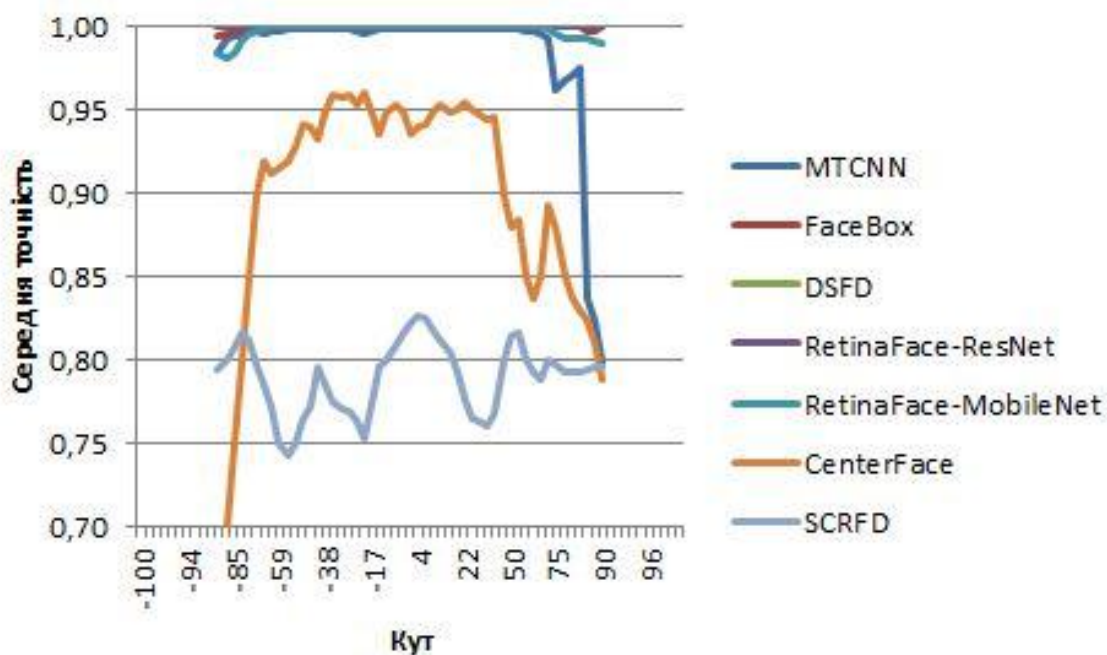


Рис. 3.6. Залежність точності ідентифікації обличчя від кута повороту навколо осі Y (рух вліво-вправо)

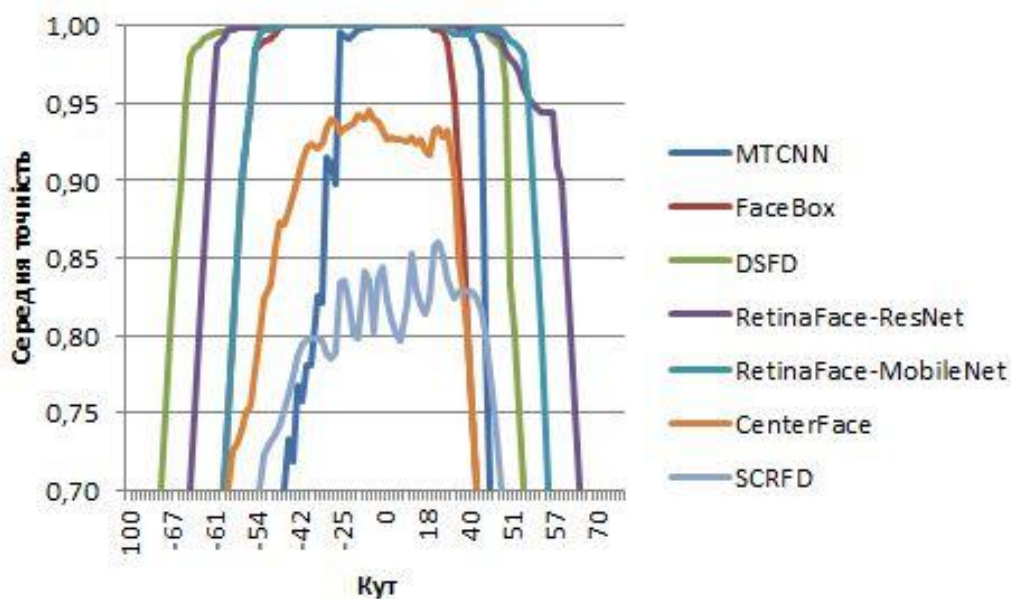


Рис. 3.7. Залежність точності ідентифікації обличчя від кута повороту навколо осі X (рух вгору-вниз)

Експериментальні результати щодо точності виявлення в умовах просторового повороту

Метод	Діапазон кутів повороту навколо осі Y (рух вліво-вправо), середня точність $\geq 0,9$	Діапазон кутів повороту навколо осі X (рух вгору-вниз), середня точність $\geq 0,9$
MTCNN	[-89;83] (range=172)	[-31;47] (range=78)
FaceBoxes	[-90;90] (range=180)	[-62;40] (range=102)
DSFD	[-90;90] (range=180)	[-69;50] (range=119)
RetinaFaceResNet125	[-90;90] (range=180)	[-65;62] (range=127)
RetinaFaceMobileNet0.25	[-90;90] (range=180)	[-60;55] (range=115)
CenterFace	[-71;46] (range=117)	[-42;37] (range=79)
SCRFD0.5GF	∅	∅

У випадку поворотів навколо осі Y (рух ліворуч-праворуч) середнє значення точності вище 0,9 і максимально можливий діапазон змін кута [-90;90] у FaceBoxes, DSFD, моделі RetinaFaceResNet125, RetinaFaceMobileNet0.25. MTCNN виявляє із середнім значенням точності вище 0,9 у діапазоні [-89;83]. Далі йде модель CenterFace з діапазоном значно меншим за [-71;46] і середньою точністю не вище 0,97. Найбільш нестабільною виявилася модель SCRFD-500MF, середня точність якої змінювалася синусоїдально на 0,7 значення і не перевищувала значень вище 0,83. Крім того, FaceBoxes, DSFD і RetinaFaceResNet125 мають середнє значення точності, близьке до 0,99 у всьому діапазоні. Моделі RetinaFaceMobileNet0.25 і MTCNN мають середню точність близько 0,99 у діапазоні [-75;75] і [-77;70] відповідно (таблиця 3.2).

Що стосується обертання навколо осі X (рух вгору-вниз), то моделі FaceBoxes, DSFD, RetinaFaceResNet125 і RetinaFaceMobileNet0.25 також показали кращі результати. Однак навколо осі X діапазон значень кута повороту був набагато меншим порівняно з віссю Y. В експериментах із середньою вірогідністю вище 0,9 найкращими моделями були RetinaFaceResNet125, DSFD, RetinaFaceMobileNet0.25 з діапазонами [-65;62], [-

69;50], [-60;55] відповідно, за ними йшли FaceVox з діапазоном [-62;40]. Моделі MTCNN і CenterFace мають подібний діапазон [-31;47] і [-42;37] відповідно, але середня точність для CenterFace не перевищує 0,95. Середнє довірче значення SCRFD500MF змінюється нестабільно, графік синусоїдальний і не перевищує 0,87. Також можна відзначити, що всі розглянуті моделі, крім CenterFace, SCRFD-500MF, досягають середніх значень точності, близьких до 0,99 (DSFD, RetinaFaceResNet125, RetinaFaceMobileNet0,25 в діапазоні [-50;47], FaceBoxes – [-48;25). ], MTCNN – [-20;40].

Загалом, з точки зору точності виявлення за наявності повороту обличчя, розглянуті методи можна ранжувати наступним чином: RetinaFaceResNet125, DSFD > RetinaFaceMobileNet0.25 > FaceBoxes > MTCNN >> CenterFace >> SCRFD-500MF, де «>» означає, що метод ліворуч більш надійний, ніж той, що праворуч, ">>" означає, що існує значна різниця між надійністю методів.

Було створено додатковий штучний набір даних для вивчення точності методів розпізнавання обличчя в залежності від розміру обличчя. Спочатку було створено 10 облич за допомогою редактора Generated Photos. На наступному кроці для кожного обличчя були згенеровані тестові зображення, щоб представити їх у різних розмірах. Розмір обличчя було змінено з кроком 10 пікселів. Найменший розмір — 20×20 пікселів, найбільший — 310×310 пікселів.

Для моделей виявлення, які використовують немаксимальне придушення (NMS) для об'єднання результатів, для NMS було встановлено поріг 0,3. Зображення були передані в методи без додаткової нормалізації або зміни розміру.

На рис. 3.8 і 3.9 показано кілька прикладів результатів виявлення зображень із штучного набору даних. Виявлені обличчя оточені прямокутником, колір якого залежить від значення точності, яке повертає модель виявлення, і значення точності також відображається поруч із ним у вигляді цифр.

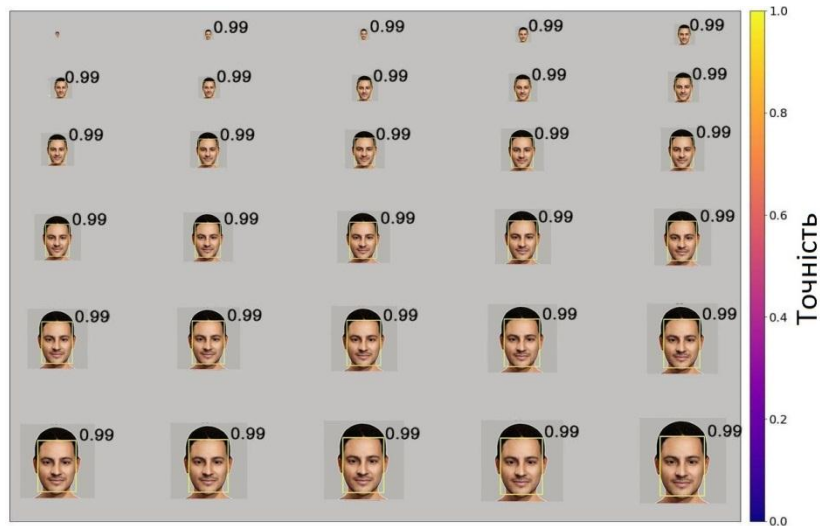


Рис. 3.8. Залежність точності для MTCNN від розміру зображення, який змінюється від  $20 \times 20$  до  $310 \times 310$  (px)

Середні значення точності були розраховані для кожного розміру обличчя на основі значень точності для всіх експериментів. На рис. 3.10 показано залежність середньої точності від зміни розміру обличчя.

Можна зробити висновок, що моделі MTCNN, FaceBoxes, DSFD, RetinaFaceRes-Net125, RetinaFaceMobileNet0.25 дуже стабільно визначають обличчя розміром від  $75 \times 75$  до  $310 \times 310$  (px) (середня точність  $\geq 0,99$ ). Крім того, модель MTCNN дає середню точність  $\geq 0,99$ , починаючи з розміру обличчя  $50 \times 50$  (px), а в деяких випадках вона виявляла обличчя  $30 \times 30$  (px) з точністю 0,99. Це означає, що MTCNN виявляє найменший розмір обличчя з високою точністю, часто навіть  $30 \times 30$  (px), як показано на рис. 3.9, мають середню точність  $\geq 0,99$ . Для DSFD, RetinaFace-ResNet125, розмір першого обличчя має бути принаймні  $60 \times 60$  (px), щоб отримати середню точність  $\geq 0,99$ . Моделі FaceBoxes, RetinaFace-MobileNet0.25 мають середню точність  $\geq 0,99$  для розмірів зображення, починаючи з  $70 \times 70$  (px). Середня точність моделі CenterFace не перевищує 0,95 у всьому діапазоні, але демонструє найвищу середню точність 0,62 для дуже маленьких зображень  $20 \times 20$  (px). Для порівняння: для зображень такого розміру моделі MTCNN, RetinaFaceResNet125, FaceBoxes, DSFD мають набагато нижчу середню точність. CenterFace має середню точність вище 0,90, лише починаючи з  $80 \times 80$

(px). Тим не менш, цей метод зміг знайти найменший розмір  $20 \times 20$  (px) і має менший розкид ймовірностей із зміною масштабу. Модель SCRFD0.5GF виявляє зображення, починаючи з  $30 \times 30$  (px), але середня точність не досягає вище 0,9, і вона має 0,9 лише в невеликому діапазоні розмірів від  $140 \times 140$  до  $170 \times 170$  (px). Крім того, під час експериментів модель показала нестабільну роботу.

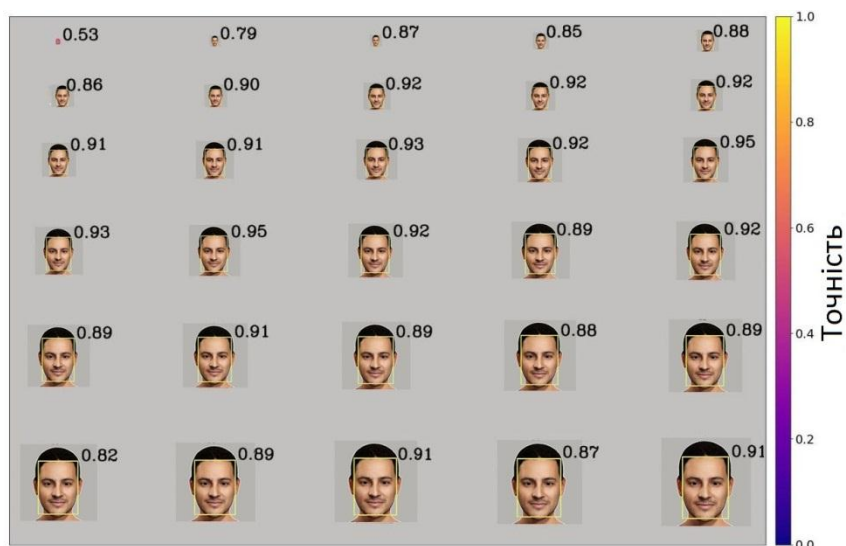


Рис. 3.9. Залежність точності для CenterFace від розміру зображення, який варіюється від  $20 \times 20$  до  $310 \times 310$  (px)

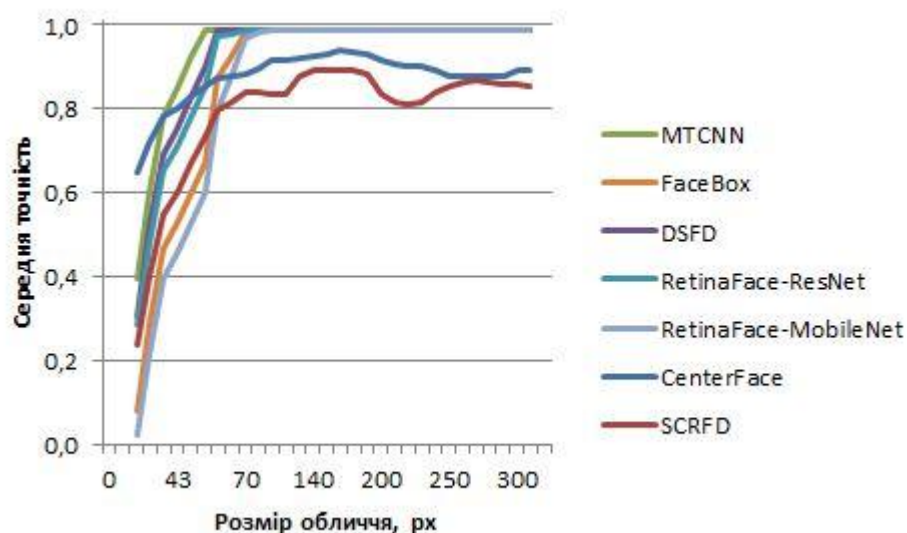


Рис. 3.10. Залежність середніх значень точності від зміни розміру зображення від  $20 \times 20$  до  $310 \times 310$  (px)

В таблиці 3.3 наведено результати ідентифікації при зміні мінімального розміру обличчя із середньою точністю більше 0,9 та 0,99.

Таблиця 3.3

Результати досліджень методів виявлення за умов зміни розмірів обличчя

Метод	Мінімальний розмір обличчя (px), середня точність $\geq 0,9$	Мінімальний розмір обличчя (px), середня точність $\geq 0,99$
MTCNN	45×45	50×50
FaceBoxes	60×60	70×70
DSFD	50×50	60×60
RetinaFaceResNet125	53×53	60×60
RetinaFaceMobileNet0.25	63×63	70×70
CenterFace	80×80	∅
SCRFD0.5GF	140×140	∅

Підсумовуючи, можна сказати, що моделі можна ранжувати за найбільшим діапазоном зміни розміру обличчя із середньою точністю 0,99 або вище, в такій послідовності: MTCNN>DSFD, RetinaFace-ResNet> FaceBoxes, RetinaFace-MobileNet0.25>> CenterFace> SCRFD-500MF.

Методи порівнювалися з точки зору затрат часу на двох зображеннях з різною роздільною здатністю: VGA (640×480 пікселів) і HD (1280×720 пікселів). Це пов'язано з тим, що ці розміри є звичайними для камер, і, як було показано в попередніх тестах, виявлення може відбуватися на обличчях розміром лише 20 x 20 (px). Тому необхідно досліджувати продуктивність мереж за різних умов, оскільки можна зменшити або збільшити роздільну здатність кадрів і, таким чином, змінити витрати часу в ту чи іншу сторону. Всі вимірювання проводилися з попереднім «розігрівом» мережі, пропускаючи перше виявлення, на відеокарті NVIDIA GeForce GTX 780.

Результати вимірювання швидкості розпізнавання обличчя для 100 зображень наведені на діаграмах, де по осі X відкладено значення часу (мс), а

по осі Y – нормалізована кількість кадрів з певним часом обробки кадру (рис. 3.11, 3.12).

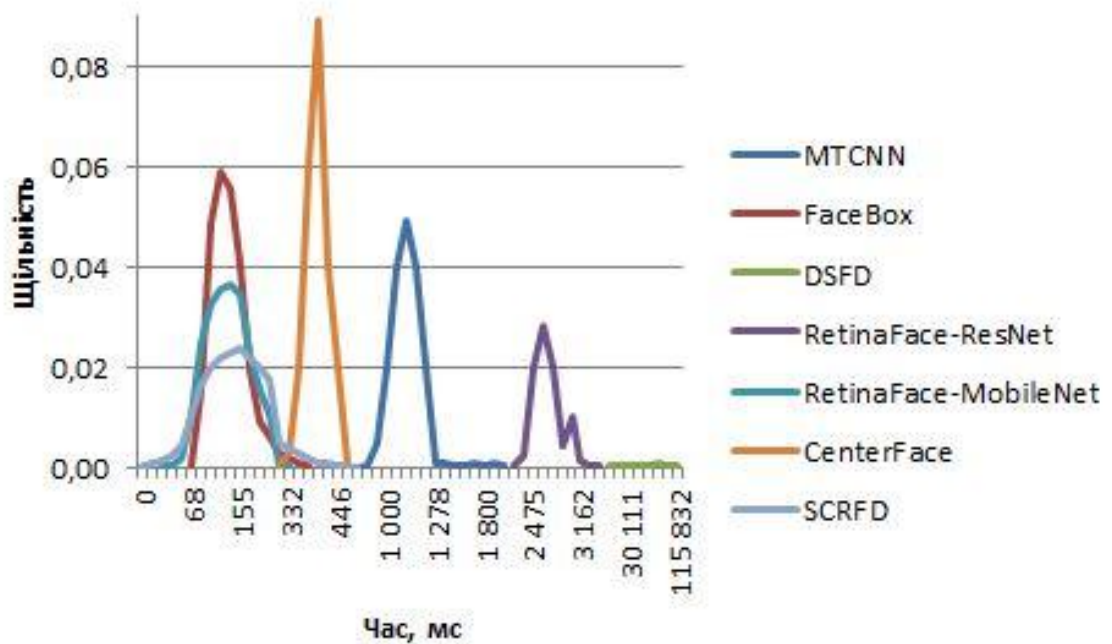


Рис. 3.11. Щільність розподілу часу обробки кадру для зображень VGA

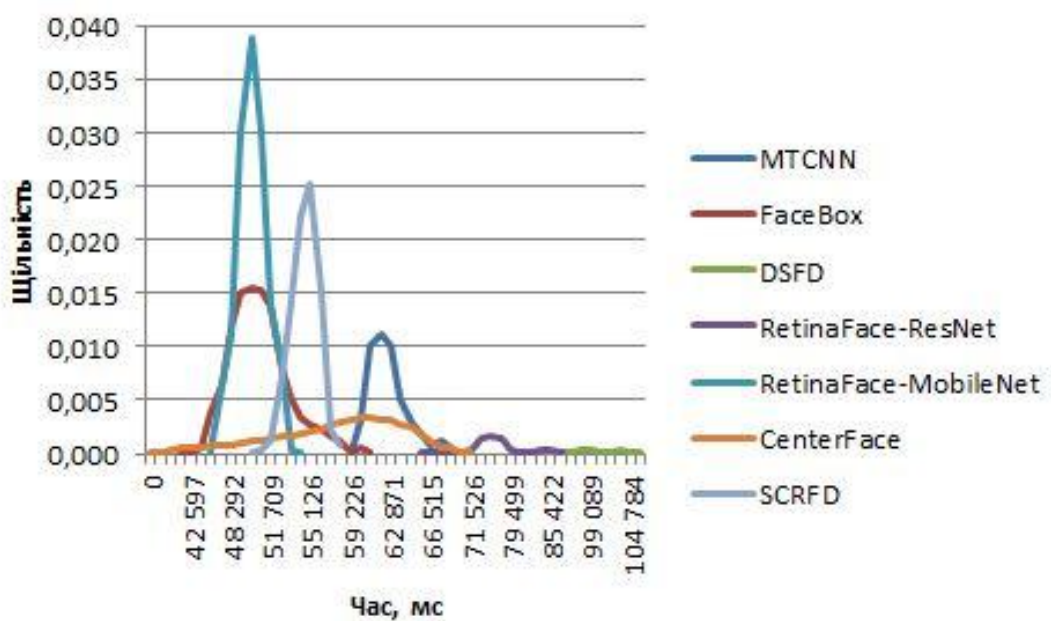


Рис. 3.12. Щільність розподілу часу обробки кадру для HD-зображень

На рис. 3.13 наведено графік, що показує основну статистику вимірювань часу, таких як медіани, перший 0,25% і третій 0,75% квантилі та викиди, для



всіх методів із зображеннями  $640 \times 480$  (px). Статистичні моделі для зображень  $1280 \times 720$  подібні. Однак метод DSFD має ще більший розкид, а метод RetinaFace-ResNet125 має 26-кратне збільшення розсіювання (рис. 3.14).

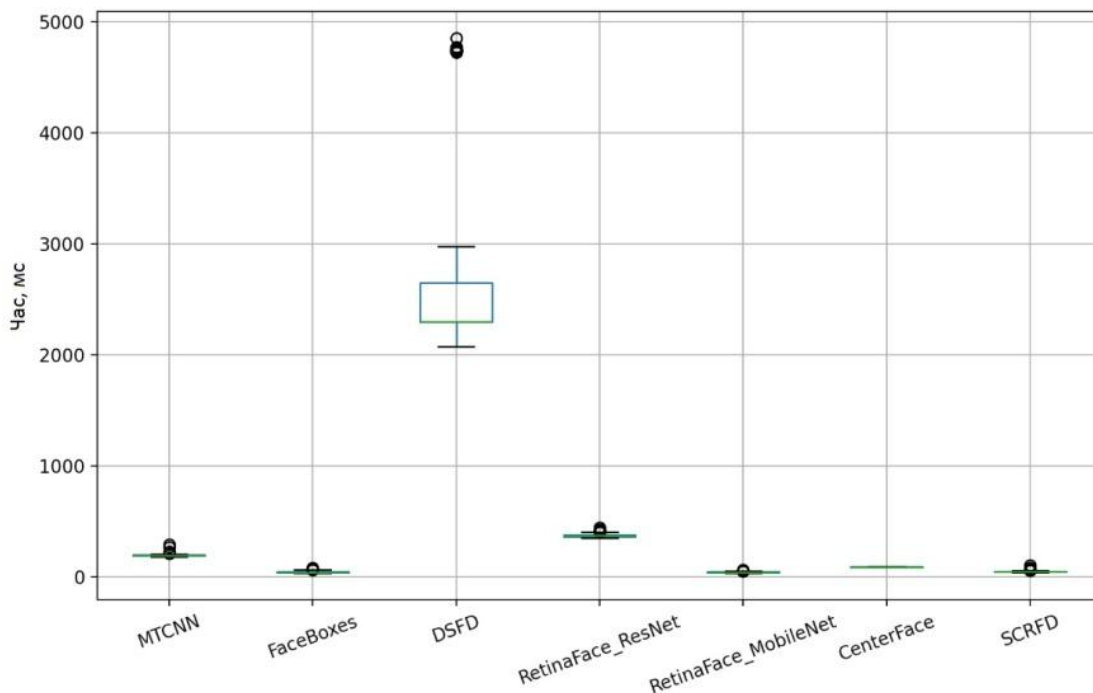


Рис. 3.13. Статистика часу обробки кадру для зображень у форматі VGA

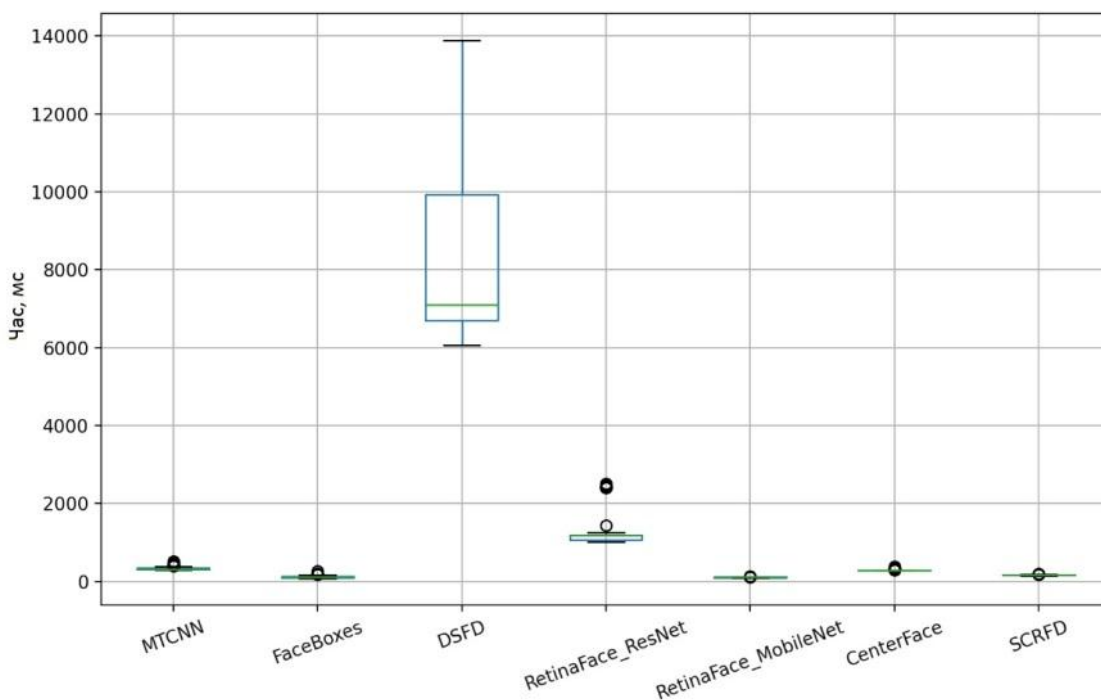


Рис. 3.14. Статистика часу обробки кадру для зображень у форматі HD



Значення середнього часу, медіани та стандартне відхилення часу виявлення обличчя.

Метод	Середній час обробки кадру, мс 640×480	Медіана часу обробки кадру, мс 640×480	Стандартне відхилення часу обробки кадру, мс 640×480	Середній час обробки кадру, мс 1280×720	Медіана часу обробки кадру, мс 1280×720	Стандартне відхилення часу обробки кадру, мс 1280×720
MTCNN	193	191	12	342	315	56
FaceBoxes	42	40	8	112	105	31
DSFD	2863	2292	1005	8754	7090	2802
RetinaFace-ResNet	375	370	17	1373	1175	493
RetinaFace-MobileNet	40	38	4	102	101	4
CenterFace	86	86	1	276	272	15
SCRFD-500MF	43	42	7	155	153	7

На основі рис.в 3.10–3.14 і даних у таблиці 3.4 можна зробити висновок, що найшвидшими моделями є RetinaFaceMobileNet0.25, FaceBoxes і SCRFD-500MF, оскільки вони мають подібний середній час обробки кадрів, який становить 42 мс, 44 мс, і 46 мс для зображень 640 × 480 пікселів і 103 мс, 115 мс і 157 мс для зображень 1280 × 720 пікселів відповідно. Крім того, зміна середнього часу обробки кадру для цих моделей дуже мала від експерименту до експерименту. Далі йде модель CenterFace, яка має час виявлення 89 мс і 274 мс для зображень 640 × 480 пікселів і 1280 × 720 пікселів відповідно. Цей метод у 2,1 рази швидший за RetinaFace-MobileNet0.25 для 640 × 480 і в 2,7 рази швидше для 1280 × 720. Час виявлення моделі MTCNN у 2,2 рази швидший, ніж CenterFace для зображень 640 × 480, але лише в 1,2 рази швидше для зображень 1280 × 720. З RetinaFace-ResNet125 розпізнавання обличчя займає

ще більше часу. Ця мережа вже працює в 9 разів довше, ніж RetinaFaceMobileNet0.25 для 640×480 і в 11,5 разів довше для 1280×720. Метод DSFD значно відрізняється від інших, він визначає в 68 разів повільніше, ніж RetinaFace-MobileNet0.25 для 640 × 480 пікселів і в 85 разів повільніше для 1280 × 720 пікселів. Крім того, DSFD має великий розкид порівняно з іншими методами, який становить 1007 мс (640×480). Наприклад, для методу CenterFace розкид становить 1 мс, RetinaFaceResNet125 — 19 мс.

На підставі проведених експериментів розглянуті методи можна ранжувати за часовими витратами наступним чином: RetinaFace-MobileNet0.25, FaceBoxes, SCRFD-500MF > CenterFace > MTCNN > RetinaFaceResNet125 >>> DSFD.

Дослідження поділяється на чотири частини: виявлення обличчя, обробка зображень обличчя, розпізнавання обличчя та порівняння класифікації на основі ознак. На етапі навчання дані зображення обличчя завантажуються в систему виявлення обличчя CenterFace [10], після чого відбувається обробка даних для зображень обличчя, наприклад доповнення даних і вирівнювання обличчя. Потім оброблені дані зображення вводяться в глибоку згортову нейронну мережу для навчання, що призводить до вилучення векторів ознак обличчя. Навчена модель зберігається для використання в майбутньому. На етапі прогнозування обличчя, яке потрібно перевірити, проходить виділення ознак за допомогою попередньо навченої структури моделі обличчя. Два вбудовані вектори ознак порівнюються, щоб визначити, чи вони однакові. Однак для ефективного розпізнавання обличчя недостатньо лише наявності дискримінаційних ознак, отриманих із навченої моделі. Тому необхідно доповнити навчену модель класифікатором SVM, щоб досягти бажаного ефекту розпізнавання обличчя.

Схема потоку дослідження показана на рис. 3.15.

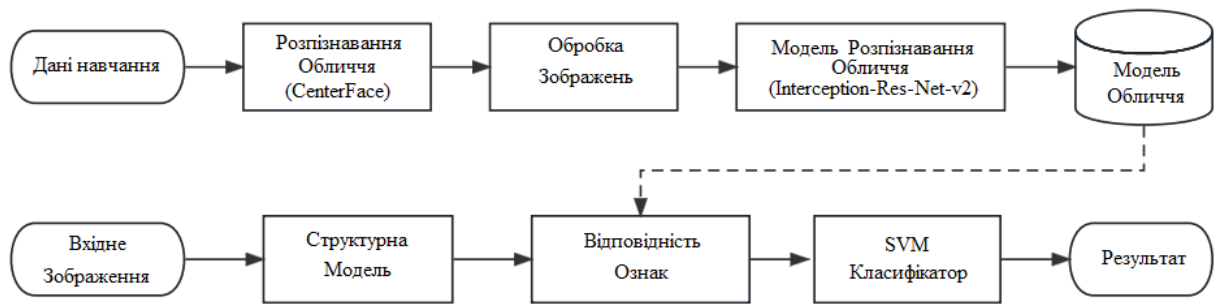


Рис. 3.15. Блок-схема розпізнавання обличчя.

Після завершення визначення обличчя обробка зображення служить не тільки для усунення перешкод, але й для перетворення зображення в ідеальний формат для наступних завдань, таких як виділення ознак, з метою покращення загальної продуктивності розпізнавання.

У реальному світі люди можуть мати набір даних зображення, знятих в обмежених сценах. Однак цільове застосування може включати різні умови, такі як різні орієнтації, положення, коефіцієнти масштабування та рівні яскравості. Тому необхідно навчити нейронні мережі справлятися з такими різноманітними ситуаціями шляхом включення синтетично згенерованих даних. Збільшення даних – це техніка, яка штучно розширює обмежені дані для створення більш цінних даних, долаючи проблеми недостатніх і однорідних даних навчання. Цей метод знаходить широке застосування в області глибокого навчання. Однак через відмінності між згенерованими та реальними даними він неминуче створює деякі проблеми. Суть доповнення даних полягає в тому, щоб підвищити адаптивність моделі навчання до невідомих даних. Використовується п'ять методів збільшення даних, включаючи перегортання, обертання, кадрування, масштабування та зсув.

Процес нормалізації зображення складається з двох частин. Перша частина — це перетворення градацій сірого, головним чином спрямоване на зменшення інформації про зображення. Друга частина — це вилучення зображень, метою якого є швидке отримання ключових точок обличчя, видалення непотрібних фонових зображень і нормалізація розміру зображення для підвищення ефективності моделі.

Перетворення відтінків сірого: у колірному просторі RGB кожен піксель містить три компоненти — червоний, зелений і синій — зі значеннями в діапазоні від 0 до 255. У зображеннях у відтінках сірого яскравість кожного пікселя представлено значенням у діапазоні від 0 до 255. Для перетворення кольорового зображення в градації сірого використовується формула.

$$Gray(R, G, B) = 0,299R + 0,587G + 0,144B, \quad (3.1)$$

Цей метод покращує контрастність, при цьому яскравість на зображенні підсвічує крайові риси, завдяки чому риси обличчя краще фіксуються під час виділення рис.

Вилучення зображень. Вилучення зображень насамперед спрямоване на видалення непотрібної інформації із зображень обличчя. Оскільки кожне зображення обличчя має різний розмір, стандартизація країв обличчя вимагає виділення бажаних частин за допомогою виділення області зображення. Під час цього процесу потрібно видалити невідповідні зовнішні частини зображень обличчя, що також є частиною нормалізації.

Вирівнювання обличчя стосується налаштування виявленого обличчя до стандартизованої пози для кращої адаптації до наступних завдань розпізнавання обличчя або аналізу. Метою вирівнювання обличчя є забезпечення узгодженої пози обличчя та зовнішнього вигляду, що підвищує надійність і точність наступних завдань.

Після виконання визначення ключових точок обличчя за допомогою алгоритму виявлення обличчя CenterFace, згаданого вище, виявлену інформацію про ключові точки можна використовувати для застосування афінного перетворення для обертання та масштабування, зрештою досягаючи ефекту вирівнювання обличчя.

Афінне перетворення є різновидом лінійного перетворення. Для двовимірного афінного перетворення його можна представити як комбінацію множення матриці. Припускаючи двовимірну точку  $(x, y)$ , результуюча точка після афінного перетворення позначається як  $(x', y')$ . Якщо зображення зазнає

кута повороту  $\theta$  та коефіцієнтів масштабування  $S_x$  і  $S_y$  у напрямках  $x$  і  $y$ , афінне перетворення можна виразити такою формулою.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} S_x \cos(\theta) & -S_x \sin(\theta) & c \\ S_x \sin(\theta) & S_x \cos(\theta) & f \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, \quad (3.2)$$

У даному випадку вирівнювання обличчя досягається шляхом визначення ключових точок обличчя для отримання координат цих ключових точок. Згодом орієнтація обличчя регулюється на основі координат ключових точок обличчя, завершуючи процес вирівнювання обличчя. Підхід передбачає спочатку визначення координат ключових точок за допомогою CenterFace, а потім використання афінного перетворення в OpenCV для вирівнювання.

Традиційні алгоритми ознак обличчя включають PCA (аналіз основних компонентів) і LBP (локальний двійковий шаблон). З розвитком згорткових нейронних мереж (CNN) виокремленням рис обличчя зараз займаються переважно CNN. З одного боку, CNN дозволяють навчати великі пакети даних, а з іншого боку, вони значно покращують точність вилучення ознак, даючи кращі результати.

CNN обробляють дані зображення за допомогою таких компонентів, як згорткові шари, шари об'єднання та повнозв'язані шари. Локальні об'єкти виділяються за допомогою згорткових операцій, розміри даних зменшуються за допомогою операцій об'єднання, а остаточна класифікація виконується з використанням повнозв'язаних шарів. Основна ідея полягає в тому, щоб отримувати високорівневі абстрактні об'єкти зображень за допомогою пошарового укладання, що дозволяє мережі автоматично вивчати шаблони та особливості в даних. Навпаки, ResNet [14] вводить концепцію залишкового навчання. Завдяки розробці залишкових блоків і впровадженню пропускаючих з'єднань у мережі, градієнти можуть поширюватися безпосередньо через ці пропускаючі з'єднання, ефективно вирішуючи проблеми зникнення та розриву градієнта під час глибокого навчання нейронної мережі. Конструкція ResNet полегшує навчання мережі, дозволяючи побудувати більш глибокі мережеві

структури та призвести до значних досягнень у практичних застосуваннях. Це представляє основний фокус поточного дослідження, мережеві архітектури Inception і Inception-ResNet.

Inception-v3 працює наступним чином. Від вхідного зображення і до початкового модуля, є 3 модулі Inception-A, 5 модулів Inception-B і 2 модулі Inception-C. Модулі Inception — це ретельно розроблені згорткові модулі, які можуть генерувати розрізнявальні функції, одночасно зменшуючи кількість параметрів. Кожен модуль Inception складається з паралельних згорткових шарів і шарів об'єднання.

Після згорткових шарів і початкових модулів розмір карти функцій становить 5x5 із 2048-вимірним вектором. Вилучення ознак використовує операції Stride, Padding і MaxPooling у межах згортки для обчислення вектора ознак зображення обличчя. Вектор ознак піддається обробці за допомогою середнього об'єднання, вилучення, рівня повнозв'язності та нормалізації L2.

Натомість в Inception-v4 є 4 модулі Inception-A, 7 модулів Inception-B і 3 модулі Inception-C, а також модулі Reduction-A Module і Reduction-B Module. Кожен модуль Inception складається з паралельних згорткових шарів і шарів об'єднання.

Після згорткових шарів і початкових модулів розмір карти функцій становить 5x5 із 1536-вимірним вектором. Вилучення ознак використовує операції Stride, Padding і MaxPooling у межах згортки для обчислення вектора ознак зображення обличчя. Вектор ознак піддається обробці за допомогою середнього об'єднання, вилучення, рівня повнозв'язності та нормалізації L2.

Inception-v4 додатково збільшує глибину з 48 шарів до 73, включаючи 4 модулі Inception із загальною кількістю параметрів 42 мільйони. Inception-v4 усуває допоміжні класифікатори, присутні в Inception-v3, вибираючи спрощену та оптимізовану структуру. Він також представляє такі методи, як залишкові підключення від ResNet. Ці вдосконалення спрямовані на покращення продуктивності моделі, включаючи вищу точність і кращі можливості узагальнення. Вибір моделі для використання часто залежить від складності

завдання та наявності обчислювальних ресурсів. Inception-v4 розроблено для адаптації до більш складних завдань комп'ютерного зору, забезпечуючи надійну підтримку для прогресу глибокого навчання в області обробки зображень.

Архітектура Inception-ResNet-v1 виглядає наступним чином. Від вхідного зображення і до модуля Inception-ResNet, є 5 модулів Inception-ResNet-A, 10 модулів Inception-ResNet-B і 5 модулів Inception-ResNet-C разом із Reduction-A Module і Reduction-B Module. Вилучення функцій використовує операції всередині згортки, включаючи Stride, Padding і MaxPooling, щоб обчислити вектор ознак зображення обличчя. Вектор ознак обробляється за допомогою AveragePooling, Dropout, FullyConnected Layer і L2 Normalization Layer.

Архітектура Inception-ResNet-v2 суттєво не відрізняється від Inception-ResNet-v1. Відмінності полягають у вихідних розмірах після обчислення через різні структури Stem, причому Inception-ResNet-v2 має вищі розміри, ніж Inception-ResNet-v1. Крім того, кожен модуль має більше параметрів. Вилучення ознак включає операції в згортці, включаючи Stride, Padding і MaxPooling, для обчислення вектора ознак зображення обличчя. Вектор ознак обробляється за допомогою AveragePooling, Dropout, Fully Connected Layer і L2 Normalization Layer.

У мережі Residual-Inception модуль Inception-ResNet використовує модуль Inception з нижчою обчислювальною складністю, ніж вихідний модуль Inception. Примітно, що термін «Початковий» передбачає включення залишкової структури ResNet, де вихід кожного рівня Inception-ResNet додається до його входу, поглиблюючи структуру мережі. Модуль Inception-ResNet — це ретельно розроблений згортковий модуль, який здатний генерувати відмітні особливості, зменшуючи кількість параметрів.

З точки зору структури, Inception-ResNet-v1 подібний до Inception-ResNet-v2, головна відмінність полягає в тому, що останній є глибшим і складнішим, що призводить до більшої кількості параметрів і досягнення вищої точності. Істотна відмінність полягає в частині попередньої обробки, де

Inception-ResNet-v2 приймає більш складну структуру Stem. Вихідний розмір в Inception-ResNet-v2 є 384-вимірним вектором, більшим за 256-вимірний вектор Inception-ResNet-v1. Складна структура Stem призводить до дещо нижчої швидкості навчання для Inception-ResNet-v2, але забезпечує кращу загальну продуктивність.

SVM — це контрольований алгоритм навчання, який в основному використовується для завдань класифікації та регресії. Мета SVM — знайти оптимальну гіперплощину, яка розділяє набір даних на два класи, максимізуючи запас між двома класами. Опорні вектори є навчальними зразками, найближчими до гіперплощини, які відіграють вирішальну роль у визначенні гіперплощини.

Основна ідея SVM полягає у відображенні необроблених даних у високовимірному просторі, знаходячи в цьому просторі гіперплощину для досягнення лінійного розділення даних. Зазвичай використовуювані функції ядра включають лінійні, поліноміальні, RBF та інші. Вибираючи різні функції ядра, SVM може адаптуватися до різних типів розподілу даних.

У глибокому навчанні магістральні мережі, такі як Inception-ResNet-v2, часто використовуються для вивчення представлень функцій високого рівня, які краще фіксують інформацію в зображеннях. Однак моделі глибокого навчання можуть страждати від надмірного оснащення або недостатнього узагальнення, особливо якщо вони навчаються на обмежених даних або мають справу зі складними завданнями.

Щоб вирішити цю проблему, традиційні класифікатори машинного навчання, такі як SVM, можна використовувати для класифікації високорівневих функцій, отриманих моделями глибокого навчання. Цей підхід поєднує в собі здатність глибокого навчання до узагальнення традиційного машинного навчання.

Таким чином, використовуючи класифікатор SVM після Inception-ResNet-v2, точність класифікації обличчя можна додатково покращити на основі моделі глибокого навчання, підвищуючи стійкість до невеликих даних. Цей



комбінований підхід продемонстрував відмінні результати в багатьох завданнях розпізнавання обличчя.

Навчальний набір даних зображення обличчя розділений на набори для навчання, тестування та перевірки. У цьому дослідженні використовується база даних CASIA-WebFace [15], причому 90% окремих зображень обличчя віднесено до навчального набору та 10% до набору перевірки для навчання моделі. Модель вчиться порівнювати вектори рис обличчя в просторі ознак, щоб визначити, чи мають вони подібні характеристики. Нарешті, для тестування використовується база даних LFW [16], яка служить еталоном для оцінки точності моделі.

База даних CASIA-WebFace містить 10 575 осіб із 494 414 зображеннями, наданими Інститутом автоматизації Китайської академії наук (CASIA). Він містить різноманітну колекцію зображень обличчя різних людей, які демонструють складність із різними позами, виразами та умовами освітлення. Це робить базу даних CASIA-WebFace складним набором даних, цінним для тестування надійності та здатності до узагальнення методів розпізнавання обличчя.

База даних LFW — це стандартний набір оціночних даних для розпізнавання обличчя. Він містить понад 13 000 зображень обличчя, зібраних з Інтернету, що представляють приблизно 1680 різних ідентичностей 5749 осіб. База даних містить реальні зображення обличчя, зняті в різних середовищах, охоплюючи різний вік, етнічну приналежність, стать і вираз обличчя. Це створює базу даних LFW як стандартний еталон для оцінки продуктивності алгоритмів розпізнавання обличчя.

Система розпізнавання обличчя в реальному часі в першу чергу розділена на два компоненти. Перший компонент — це архітектура моделі тренування обличчя, а другий — класифікатор SVM. В архітектурі моделі навчання обличчя ми вводимо зображення обличчя, а потім, після навчання з Inception-ResNet-v2, виводимо вектори ознак зображення обличчя. Відстань між векторами представляє подібність зображень, яка далі класифікується за

допомогою класифікатора SVM. Наступне зображення ілюструє архітектуру моделі тренування обличчя (рис. 3.16).

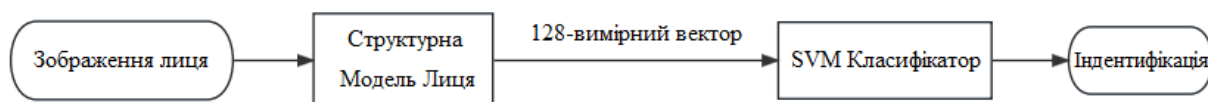


Рис. 3.16. Система розпізнавання обличчя в реальному часі.

Продуктивність архітектури моделі тренування обличчя безпосередньо впливає на загальну точність усієї системи розпізнавання обличчя. Структура моделі тренування обличчя базується на FaceNet, відмінність полягає у використанні глибокої згорткової нейронної мережі Inception-ResNet-v2 і Center Loss + Softmax Loss як функції втрат, на відміну від FaceNet (рис. 3.17).



Рис. 3.17. Навчальна структура моделі розпізнавання обличчя.

Пакетне введення попередньо оброблених зображень обличчя в модель. Глибока архітектура базується на різних архітектурах CNN, таких як Inception-v3, Inception-v4, Inception-ResNet-v1, Inception-ResNet-v2 тощо. Пакетна нормалізація передбачає нормалізацію функцій до узгодженого діапазону після попередньої обробки для отримання більш розрізнявальних характеристик. Вбудовування — це процес введення 128-вимірного вектора, створеного навченими функціями Inception-ResNet-v2 і Batch Normalized, у функцію втрат для диференціювання векторів ознак. Center Loss + Softmax Loss отримує функцію вбудовування від вектора, де риси між тими самими обличчями є ближчими, а риси між різними обличчями максимально відокремлені, що робить визначення того, чи два зображення обличчя належать одній людині, ефективніше.

Після навчання моделі отримують 128-вимірні вектори ознак. Використовуючи попередньо оброблені дані, знайдено гіперплощину, яка розділяє дві частини категорій. Після навчання класифікатора SVM цей метод

можна використовувати для швидкої та точної класифікації. Нарешті, завдяки поєднанню рис обличчя з навченою моделлю та класифікатором SVM покращується система розпізнавання обличчя.

Метод розпізнавання обличчя поєднує глибоку згорткову нейронну мережу Inception-ResNet-v2. Як вхідні дані використовуються зображення розміром  $160 \times 160$ , отримані після визначення обличчя та вирівнювання за допомогою CenterFace. Вимірні вектори 256 і 128 протестовано, і результати показують, що 128-вимірний векторний вхід працює найкраще, що відповідає FaceNet. Порівнюються при оптимізатори, RMSPROP, ADAM і ADAGRAD, причому ADAM демонструє найкращу продуктивність. Швидкість навчання встановлена на -1, що дозволяє моделі сходитися до локального мінімуму. Відсіювання підбиралося зі значеннями 0,2, 0,4, 0,6 і 0,8, при цьому 0,8 є оптимальним вибором під час навчання. Оптимальне значення ваги розпаду -  $5e-4$ . На відміну від FaceNet, використовується ADAM як оптимізатор.

Під час навчання даних вибрано розмір пакета 90, що вказує на те, що в кожному навчальному пакеті використовується 90 зображень. Кількість епох встановлено на 100, що відповідає 100 циклам навчання. Розмір епохи встановлено на 1000, тобто кожен цикл навчання складається з 1000 ітерацій. У дослідженні використовується ADAMoptimizer для мінімізації значення втрат, а точність нейронної мережі обчислюється після кожних 1000 кроків, керуючи навчанням до досягнення 100 епох.

Це дослідження також включає порівняння точності та швидкості чотирьох згорткових нейронних мереж — Inception-v3, Inception-v4, Inception-ResNet-v1 і Inception-ResNet-v2 — у наборі даних LFM. Результати наведено в таблиці 3.5.

Ефективність регресії для кожної моделі.

Модель	Точність	Швидкість
Inception-ResNet-v1	97.98%	0.34s
Inception-ResNet-v2	98.79%	0.47s
Inception-v3	96.64%	0.46s
Inception-v4	92.53%	0.47s

Згідно з наведеною вище таблицею, враховуючи як швидкість навчання, так і точність, оптимальною моделлю визначено Inception-ResNet-v2. Тому її було обрано як основу для моделі навчання обличчя. Найкраща модель розпізнавання обличчя, передбачає використання CenterFace для виявлення обличчя, введення оброблених даних в мережу Inception-ResNet-v2 і класифікацію обличчя за допомогою моделі SVM. Кінцева точність розпізнавання обличчя становить 98,79%.

### **3.3 Дослідження методу стиснення відеопотоку на основі алгоритму віднімання фону**

Оцінка продуктивності стиснення відео відбувалася порівнюючи навантаження на середовище передавання, передаючи цілий файл і оброблений методом (таблиця 3.6). Щоб оцінити ефективність методу та забезпечити збереження даних, необхідних для подальшого аналізу, збережено вихідну роздільну здатність відео, частоту кадрів і відеокодек. Тестові дані обробляли за допомогою одноплатного комп'ютера Raspberry Pi 3 (4 ГБ), який часто використовується для обробки відео та зображень. "Кадри" та «Бітрейт (МБ)» для необроблених і оброблених відео показують загальну кількість кадрів і навантаження на середовище передавання. Бітрейт для оброблених відео – це загальна сума стисненого відео та додаткової інформації яка необхідна для відновлення кадру та може застосовуватися для подальшої

обробки. «Стиснення (%)» показують у відсотках зменшення загального навантаження при передачі тестового відеофайлу.

Таблиця 3.6

Результати стиснення відео.

Датасет	Цілий відеофайл		Оброблений відеофайл		Втрата кадрів. (%)	Стиснення. (%)
	Кадри	Бітрейт (МВ)	Кадри	Бітрейт (МВ)		
Ролик 1	5445	12	5445	9	0	25
Ролик 2	179912	15	179912	10	0	30
Ролик 3	790	5	790	2	0	60
Ролик 4	5471	8	5471	4	0	50

Оцінка здатності методу передавати відповідну інформацію для подальшого аналізу рухомих об'єктів виконана порівнюючи кількість появ рухомих об'єктів, виявлених у необробленому та стисненому відео, використовуючи як ручні, так і автоматизовані методи. Для цього експерименту використано створений датасет, оскільки він має найвище стиснення відео і, ймовірно, найвищу ймовірність втрати інформації. Дотримувалися процедуру [25], щоб вручну записати події у цьому ролику. Результати наведені в таблиці 3. Порівняння ручного підрахунку комах у необроблених/стиснутих відео. «Необроблене відео» представляє спостереження з необроблених відео. «Оброблене відео» представляє спостереження з стиснутих відео. «Втрати» підраховує появу рухомих об'єктів, які не спостерігаються у стисненому відео, але враховуються на основі необроблених відеоспостережень. «Нові об'єкти» показує появу об'єктів, спостережувану в стисненому відео, не зафіксоване під час спостережень, зроблених на необробленому наборі відеоданих. Підрахунки, пов'язані з «Нові об'єкти», включені до «Оброблене відео» (таблиця 3.7).

## Підтвердження кількості рухомих об'єктів.

Набір даних	Кількість рухомих об'єктів			
	Без алгоритму	Оброблене відео	Втрати	Нові об'єкти
Ролик 1	20	20	0	0
Ролик 2	6	9	0	3
Ролик 3	4	5	0	1
Ролик 4	10	13	0	3

Щоб оцінити придатність стисненого відео для автоматизованого відстеження об'єктів, використано Polytrack [21, 25] для вилучення траєкторій і положення із стисненого відео. Для цього експерименту використано попередньо підготовлену модель виявлення об'єктів YOLOv8 із програмними конфігураціями за замовчуванням [21, 25]. Результати представлені як траєкторії руху об'єктів на рис 3.18.

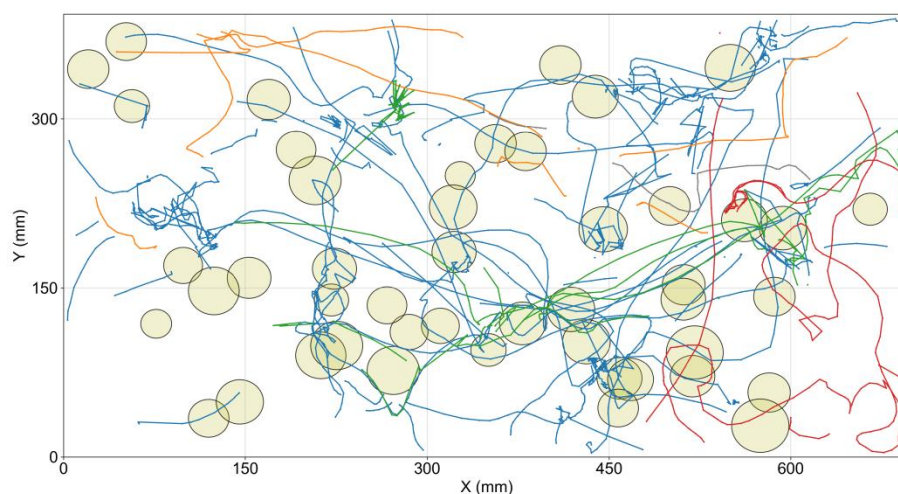


Рис. 3.18. Траєкторії і положення, витягнуті зі стисненого відео за допомогою програмного забезпечення Polytrack [21].

Метод стиснення аналізує попиксельний рух у відеокадрах, щоб видалити кадри та окремі пікселі, позбавлені інформації про рух, зберігаючи при цьому важливі дані. Він досяг середнього стиснення 30% для відеоспостереження у різноманітних середовищах (таблиця 3.6), зберігаючи ключові дані для подальшої обробки (таблиця 3.7, рис. 3.18). Значне зменшення розміру файлу

призводить до зменшення потреб у сховищі та пропускну́ї здатності, що є корисним для обмежених ресурсів. Запропонований метод послідовно досягав відсоткового зменшення розміру файлу, що перевищує відсоток зменшення кадру в усіх наборах даних, особливо у відео середовищ із незначними змінами фону (таблиця 3.6). У цих випадках попіксельний аналіз руху вибірково видаляв дані з нерухомих пікселів, але зберігав для аналізу дані з пікселів з рухом. Це демонструє адаптивність підходу до різних умов навколишнього середовища, великої кількості рухомих об'єктів у кадрі.

У експериментах метод зберігав усі дані, необхідні для оцінки кількості об'єктів (таблиця 3.7). Крім того, ручні спостереження за стисненими відеозаписами виявили більшу кількість об'єктів порівняно з необробленими відеоспостереженнями. Можливо, це пов'язано з тим, що стиснені відео спрощують завдання, зосереджуючи увагу на ключових частинах зображення, щоб зменшити втому користувача [7, 27, 30].

Використано алгоритм ViVe як ініціатор руху без оцінки подій або об'єктів, що викликають рух на пристрої. Це реєструє зовнішній вигляд об'єктів з невеликою кількістю хибно негативних результатів (таблиця 3.7) і тому зберігає важливі дані. Але стиснуті відео містили хибні спрацьовування, спричинені змінами вітру та освітлення, які надмірно збільшували розмір стисненого відеофайлу в динамічному середовищі. Майбутні методи зменшення помилкових спрацьовувань покращать продуктивність стиснення, можливо, за рахунок вбудованої обробки та витрат на ресурси.

#### **3.4. Дослідження моделі системи розпізнавання обличчя для стаціонарних систем відеоспостереження**

Запропонований метод змодельовано на наборі даних CDnet і порівняно з популярними алгоритмами віднімання фонового, підсумок представлено в таблиці 3.8. Було помічено, що модифікований ViBE перевершує рідний алгоритм ViBE у всіх сценаріях. Такі ж спостереження можна зробити з іншими алгоритмами, крім SUBSENSE. Це можна виправдати тим фактом, що навіть

якщо він перевершує модифікований ViBE у деяких випадках, він є відносно повільним і, отже, не підходить для програм реального часу.

Після виявлення об'єктів, що рухаються, зроблено опис зони інтересу, який у майбутньому складатиметься з новим вхідним зображенням. Для цього використовується набір пікселів переднього плану, визначений сегментацією, виданою з модифікованого ViBE. На цьому етапі існує двійкова матриця з розміром вхідного кадру, де пікселі переднього плану білі (255), а пікселі фону чорні (0). Потім обчислюється площа, зайнята цими пікселями переднього плану, з подальшим виявленням краю цієї області. Після того, як цю зону було обмежено, будь-який піксель усередині вважається пікселем інтересу. Таким чином, виділення цієї зони інтересу виконується за допомогою механізму кадрування, який у гіршому випадку матиме той самий розмір, що й вхідний кадр, інакше він матиме знижену роздільну здатність.

*Таблиця 3.8*

Підсумкова таблиця f-міри

Типи відеофайлів	Codebook	KDE	MOG	SUBSENSE	ViBE	MOD ViBE
Shadow	0.53	0.51	0.66	0.69	0.63	0.67
Blizzard	0.54	0.52	0.57	0.59	0.55	0.59
Baseline	0.48	0.51	0.64	0.68	0.65	0.67
Camera Jitter	0.54	0.51	0.54	0.58	0.57	0.57
Dynamic Background	0.45	0.51	0.56	0.61	0.58	0.59
Intermittent Object	0.54	0.51	0.55	0.54	0.56	0.59
Night Videos	0.52	0.49	0.51	0.52	0.52	0.53
Low Frame Rate	0.54	0.50	0.53	0.51	0.54	0.54

Для тестування використовувалось відео тривалістю декілька хвилин, до якого застосовувався алгоритм сегментації ViBE [20]. Цей метод виділяє зону інтересу в кадрі, яка є частиною, що містить рухомий об'єкт. Тоді було помічено, що з точки зору використання системної пам'яті використання підходу фонового віднімання не впливає суттєво на пам'ять, яка



використовується для всього процесу, оскільки цей підхід є легким. Це показано на рис. 3.19, де можна побачити зміну споживання пам'яті з відніманням фону та без нього. З іншого боку, було зроблено запис розмірів зображень, які обробляються під час операції. Це дозволило помітити, що при застосуванні алгоритму віднімання фону, розмір обробленого зображення менший, ніж розмір зображення, яке використовується без підходу віднімання фону, завдяки обробці лише частина зображення.

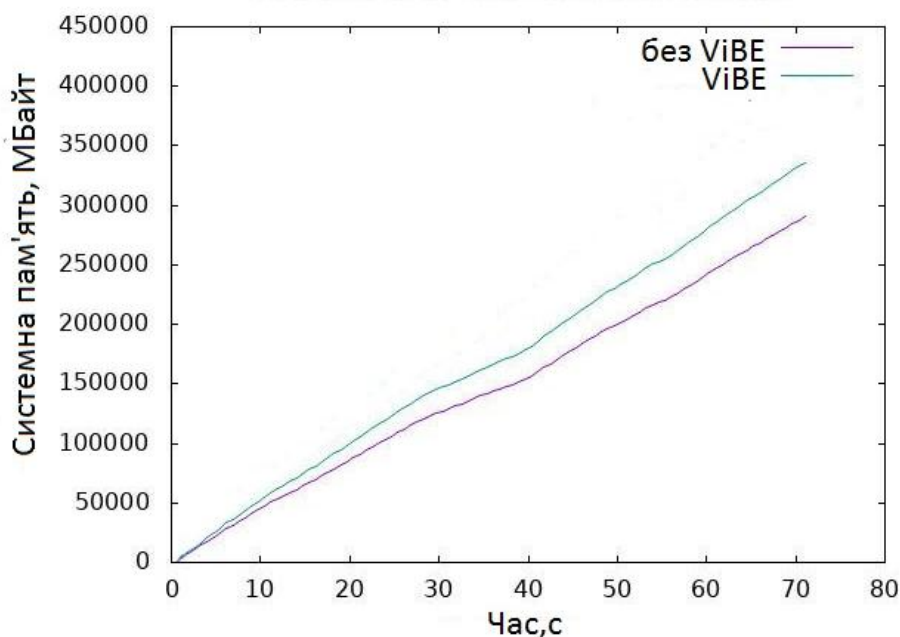


Рис. 3.19. Використання системної пам'яті в залежності від часу

Крім того, можна помітити, що краще використовувати EFF-ViBE, а не звичайний підхід сегментації ViBE. З іншого боку, було зроблено запис розмірів зображень, які обробляються під час операції. Це дозволило помітити, що при застосуванні підходів віднімання фону, зокрема ViBE та модифікований ViBE, розмір обробленого зображення менший, ніж розмір зображення, яке використовується без підходу віднімання фону. Це зрозуміло в тому сенсі, що обробляється лише частина зображення. Ілюстрація зроблена на рис. 3.20.



Рис. 3.20. Виявлення та вилучення рухомих об'єктів, стовпець А: вхідний кадр, стовпець: В вилучення рухомих об'єктів за допомогою ViBE, стовпець С: вилучення рухомих об'єктів за допомогою модифікованого ViBE

На цьому малюнку видно, що зображення, обрізане за допомогою ViBE, займає меншу кількість системної пам'яті. Також на рис 3.21 представлено зміну розміру зображення, і можна побачити значний ефект використання підходів віднімання фону, тому що без них розглядається весь кадр (зображення).

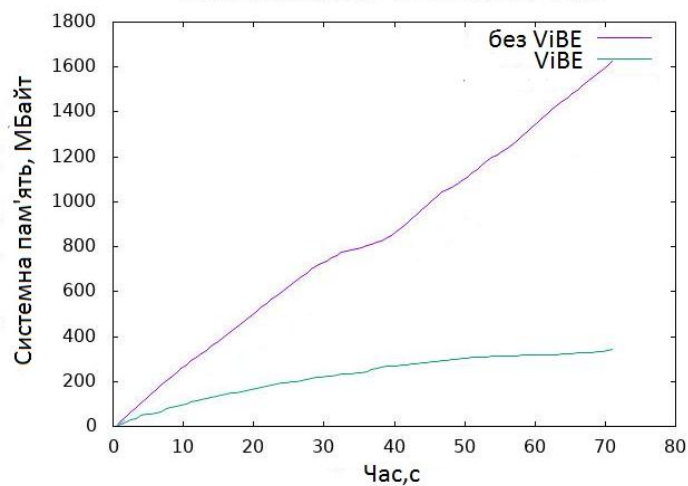


Рис. 3.21. Зміна розміру використовуваного зображення в залежності від часу

Було навчено модель, яка множить розмір вхідного зображення обличчя на рис. 3.20; еволюція функції втрат є функцією кількості епох, яка дорівнює

100. За допомогою цієї навченої моделі стало можливим покращення роздільної здатності зображення. З метою демонстрації 400 зображень 20 осіб (20 на особу) було використано для навчання, 60 для перевірки (3 на особу). Для тестування 200 зображень (по 10 на кожну людину) було зменшено з роздільною здатністю 25×25, 35×35, 45×45, 55×55. Ідея полягала в тому, щоб показати, як покращення зображення покращує виявлення та розпізнавання облич. Як результат, підсумкова кількість виявлених облич, а також відповідні відсотки з вищезгаданою роздільною здатністю наведені в таблиці 3.9. З іншого боку, середній відсоток хорошого розпізнавання осіб, взятих на етапі тестування, наведено в таблиці 3.10.

*Таблиця 3.9*

Кількість облич, визначених за роздільністю вхідного зображення

Роздільна здатність	25×25	35×35	45×45	55×55
Нормальне розпізнавання обличчя	175	189	189	191
Покращене виявлення облич	191	195	195	197

Можна помітити, що низька роздільна здатність зображення призводить до поганих результатів як на етапі виявлення обличчя, так і на етапі розпізнавання. Зі збільшенням роздільної здатності зображення також спостерігається прямий вплив на продуктивність.

*Таблиця 3.10*

Середній відсоток доброго розпізнавання в функції роздільної роздільності вхідного зображення

Роздільна здатність	25×25	35×35	45×45	55×55
Нормальне розпізнавання обличчя	5	36	63.5	79
Покращене виявлення облич	32.5	58	76	88

Крім того, виходячи зі значень, наведених у двох таблицях, механізм суперроздільності добре впливає на результати в тому сенсі, що він значно збільшує кількість виявлених облич, а також відсоток розпізнавання

### **Висновки до розділу 3**

Проведено дослідження запропонованого методу віднімання фону. Ефективність модифікованого методу з точки зору запам'ятовування порівняно зі згаданими методами перевершує всі вибрані алгоритми із середнім відсотком 65% за винятком алгоритму PBAS, де він перевершує лише 30% випадків. Зрештою, об'єднання цих результатів для вимірювання F-міри показує середній відсоток перевищення продуктивності становить 85% для інших алгоритмів і 40% для алгоритму PBAS.

Здійснено експерименти щодо стійкості моделей до повороту обличчя у різних площинах, зміни розміру обличчя, а також оцінено витрати часу. Найкращими моделями за обертанням стали RetinaFaceResNet125, DSFD, RetinaFaceMobile Net0.25, які точно (точність  $\geq 0,9$ ) визначають обличчя із поворотами в діапазоні  $[-45; 45]$ , що є достатньою вимогою для використання у системі безпеки. Моделі MTCNN, FaceBoxes, CenterFace і SCRFD-500MF також працюють із поворотом обличчя в різних площинах, але у них нижча точність і менший діапазон кута обличчя.

В експериментах з різними розмірами обличчя найкращі результати показали моделі MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25, які виявляють зображення, починаючи з розміру  $75 \times 75$  px, з точністю  $\geq 0,9$ . Моделі CenterFace і SCRFD0.5GF мають значно нижчу точність, але в деяких випадках виявляють дуже маленькі обличчя.

Вимірювання часу виявлення показало, що найшвидшими є RetinaFace-MobileNet0.25 і Face Vox, які витрачають менше ніж 47 мс на опрацювання одного зображень VGA і 100 мс для розміру HG. На наступному місці SCRFD-500MF, CenterFace. Для моделей MTCNN і RetinaFaceResNet125 час виявлення

перевищує 100 мс навіть для зображень VGA. Модель DSFD не можна застосовувати в реальному часі навіть для зображень VGA.

Детально розглянуто процес вилучення ознак із зображень за допомогою нейронної мережі Inception-ResNet-v2. Після вилучення ці ознаки класифікуються за допомогою алгоритму SVM. Цей метод забезпечує точну і надійну класифікацію облич у системі.

Проведено аналіз запропонованого методу розпізнавання облич, який показав зменшення споживання обчислювальних ресурсів за рахунок зменшення зони пошуку обличчя.

Аналіз запропонованого алгоритму оптимізації відпотоків показав зменшення навантаження на телекомунікаційну мережу до 90% в залежності від кількості рухомих об'єктів у кадрі.

## РОЗДІЛ 4. ПРОГРАМНА РЕАЛІЗАЦІЯ МОДЕЛІ РОЗПІЗНАВАННЯ ОБЛИЧ НА ОСНОВІ АЛГОРИТМУ ВІДНІМАННЯ ФОНУ

### 4.1. Обґрунтування вибору технології для програмної реалізації моделі

Розглянемо вибір мови програмування для розробки прототипу системи, яка включає виявлення руху, розпізнавання облич і їх класифікацію. Серед популярних варіантів для задач машинного навчання можна виділити Python, C++ і Java. Кожна з цих мов має свої переваги та недоліки.

C++ забезпечує високу продуктивність і контроль над ресурсами, що робить його підходящим для розробки високоефективних моделей. Наприклад, фреймворк TensorFlow має основну частину написану саме на C++, що дозволяє створювати state-of-the-art нейронні мережі. Однак, основний недолік C++ – це складність розробки, велика кількість коду для реалізації навіть базових функцій, а також залежність від платформи, що вимагає перекомпіляції коду під різні архітектури.

Java також є поширеним вибором у задачах дата-саєнсу, завдяки своїй кросплатформеності та можливості запуску коду на будь-якій платформі з JVM. Проте її недолік полягає в обмеженій кількості інструментів для глибокого навчання. Крім того, хоча Java-код і забезпечує високу стабільність, його продуктивність нижча порівняно з C++, що може бути критичним у задачах, пов'язаних з реальним часом.

Python став де-факто стандартом для розробки прототипів у сфері машинного навчання завдяки великій кількості бібліотек і фреймворків. Його основні переваги – простота написання коду, велика кількість готових інструментів і активна спільнота. Хоча Python поступається C++ у швидкості виконання, цей недолік частково компенсується можливістю інтеграції з високопродуктивними бібліотеками на C++, такими як TensorFlow, що мають Python API. Ця гнучкість робить Python оптимальним вибором для розробки експериментальних прототипів і дослідницьких проєктів.

Python є найкращим вибором, оскільки він забезпечує швидке прототипування та має широкий спектр бібліотек для роботи з даними, зображеннями та нейронними мережами. При необхідності код на Python можна оптимізувати, використовуючи низькорівневі бібліотеки або переписавши частину коду на C++.

Для розробки системи на Python було обрано такі бібліотеки:

1. OpenCV: Ця бібліотека є однією з найпопулярніших для обробки зображень. Вона включає широкий набір інструментів для аналізу і трансформації зображень, таких як детекція облич, фільтрація, масштабування тощо. У контексті нашої системи OpenCV використовується для попередньої обробки кадрів, детекції об'єктів і підготовки даних для подальшого аналізу.

2. NumPy: Це фундаментальна бібліотека для роботи з багатовимірними масивами і матрицями. У задачах глибокого навчання numpy використовується для підготовки даних, обчислень та інтеграції з іншими бібліотеками, такими як TensorFlow. Нейронні мережі зазвичай працюють із 4-вимірними масивами (кількість зображень, висота, ширина, канали), і numpy надає необхідний набір інструментів для їх ефективної обробки.

3. TensorFlow: Основний фреймворк для розробки та тренування нейронних мереж. У нашій системі він використовується для вилучення ознак облич із кадрів та підготовки даних для подальшої класифікації. TensorFlow надає потужні інструменти для роботи з моделями глибокого навчання, а також дозволяє оптимізувати їх для виконання на різних платформах.

4. socket: Це вбудований модуль Python, який забезпечує доступ до низькорівневих функцій роботи з мережевими сокетми. За допомогою цього модуля можна реалізувати як клієнтську, так і серверну частини системи. У нашому проекті socket використовується для встановлення з'єднання між клієнтом і сервером, передачі кадрів, даних про виявлений рух, а також синхронізації фонового зображення.

5. pickle: Це стандартний модуль Python для серіалізації та десеріалізації об'єктів. Серіалізація перетворює Python-об'єкти в потік байтів,

що дозволяє їх передавати через мережу або зберігати у файлі. У нашій реалізації `pickle` використовується для серіалізації кадрів або інших структур даних перед їх передачею через сокет, а на сервері – для відновлення (десеріалізації) цих даних.

6. `struct`: Модуль `struct` використовується для роботи з форматованими даними. Він дозволяє перетворювати Python-об'єкти на C-подібні структури, що є корисним при передачі через мережу. У нашому проекті `struct` використовується для упаковки (формування) заголовка повідомлень, які передаються через сокети. Наприклад, перед відправкою кадру ми можемо спочатку передати його розмір за допомогою `struct`, що дозволяє серверу коректно отримати й інтерпретувати дані.

7. `CenterFace`: Це модель детекції облич на основі нейронних мереж, яка спеціально оптимізована для високої точності та продуктивності. Вона працює в реальному часі і використовується для точного визначення облич у відео або на зображеннях. У нашій реалізації `CenterFace` застосовується для детекції облич на кадрах, де було виявлено рух. Це дозволяє значно підвищити ефективність обробки, оскільки аналіз виконується лише у визначених областях.

8. Бібліотека `scikit-learn (sklearn)` є однією з найпопулярніших у сфері машинного навчання. Вона надає широкий спектр алгоритмів для класифікації, регресії, кластеризації та інших завдань. У нашому проекті `sklearn` використовується для класифікації облич за допомогою моделі `Support Vector Machine (SVM)`.

У нашій системі модель `Support Vector Machine (SVM)` використовується для класифікації облич після їх вилучення за допомогою `CenterFace` та попередньої обробки в мережі `Inception-ResNet-v2`. `SVM` є потужним інструментом для класифікації, особливо коли йдеться про невеликі набори даних або задачі, де класи межують на високорозмірних просторах ознак.

З огляду на вимоги до прототипу та необхідність гнучкості у розробці, мова Python є оптимальним вибором. Вона дозволяє зосередитися на розробці



дослідницьких компонентів, мінімізуючи витрати часу на низькорівневі аспекти програмування. Python надає доступ до потужних бібліотек і фреймворків, що спрощує роботу з даними та розробку моделей машинного навчання.

#### **4.2. Реалізація алгоритму віднімання фону ViVe та його програмна оптимізація**

Алгоритм віднімання фону ViVe (Visual Background Extractor) є популярним методом для виявлення рухомих об'єктів у відеопотоках, особливо у задачах відеоспостереження та комп'ютерного зору.

Реалізація алгоритму ViVe для віднімання фону на чипах ARM, які часто використовуються у вбудованих системах, таких як камери, потребує врахування ряду апаратних особливостей та обмежень. Вбудовані системи зазвичай мають обмежені обчислювальні ресурси (низька частота процесора, обмежений обсяг пам'яті) та енергоспоживання, тому важливо максимально оптимізувати алгоритм.

Реалізація цього алгоритму з використанням Cython дозволяє оптимізувати продуктивність за рахунок прискорення обчислень у порівнянні з чистим Python.

ViVe є простим та ефективним підходом до моделювання фону, заснованим на статистичному підході. Основні етапи роботи алгоритму:

- Ініціалізація моделі фону: Кожен піксель зображення має власну модель фону, яка складається з набору зразків (наприклад, 20 значень). Ці зразки зберігаються та оновлюються під час роботи алгоритму.
- Класифікація пікселя: Для кожного пікселя нової рамки зображення перевіряється, чи схожий він на зразки з моделі фону. Якщо піксель не схожий на більшість зразків, його вважають рухомих (належить передньому плану).
- Оновлення моделі фону: Якщо піксель класифіковано як фон, його значення може бути випадково додане до моделі фону. Це дозволяє алгоритму адаптуватися до змін у сцені.

Якщо алгоритм ViVe реалізується на Cython для ARM, слід враховувати такі аспекти, як типізація змінних та робота з масивами, підтримка simd-інструкцій та паралелізація обчислень. ARM-чипи зазвичай мають менше потужності, ніж десктопні CPU, важливо ефективно працювати з типами даних і оптимізувати доступ до масивів (наприклад, використовуючи cdef для статичної типізації). Cython дозволяє використовувати інструкції NEON на рівні C, що значно підвищує продуктивність обробки зображень. Якщо ARM-система має кілька ядер, можна використовувати паралельні цикли через prange в Cython для одночасної обробки кількох блоків кадру.

Для реалізація ViVe на Cython потрібно створити .pyx файл, де буде реалізований алгоритм. Окрім цього, додаються конфігураційні файли для компіляції setup.py.

1. Ініціалізація моделі фону: Модель фону зберігається у вигляді тривимірного масиву, де для кожного пікселя є кілька зразків. Наприклад:

```
cdef int height = image.shape[0]
cdef int width = image.shape[1]
cdef int num_samples = 20
cdef int background_model[height][width][num_samples]
```

2. Класифікація пікселя: Для кожного пікселя нового кадру обчислюється відстань до зразків фону. Якщо відстань до достатньої кількості зразків менша за порогове значення, піксель вважається фоном:

```
cdef int count = 0
for i in range(num_samples):
    if abs(image[x, y] - background_model[x][y][i]) < threshold:
        count += 1
    if count >= min_matches:
        is_background = True
        break
```

3. Оновлення моделі фону: Якщо піксель визнаний фоном, з певною ймовірністю його значення додається до моделі фону. Це допомагає адаптувати модель до поступових змін в освітленні чи текстурі сцени.

```
if is_background:  
    if random.randint(0, update_factor) == 0:  
        background_model[x][y][random.randint(0, num_samples - 1)] =  
image[x, y]
```

4. Компіляція Cython коду: Створюється файл setup.py, який використовується для компіляції Cython коду:

```
from distutils.core import setup  
from Cython.Build import cythonize  
setup(  
    ext_modules = cythonize("vibe.pyx")  
)
```

Після компіляції код можна використовувати так само, як і звичайний Python модуль, але з помітно кращою продуктивністю. Приклад роботи коду показано на рис. 4.1.



Рис 4.1. Виявлення рухомих об'єктів у відео потоці.

Використання Cython для реалізації алгоритму ViVe дозволяє значно покращити продуктивність, зберігаючи при цьому гнучкість та простоту Python. За рахунок оптимізацій, таких як явна типізація, швидкі цикли та

паралелізація, така реалізація є ефективною для задач відеоспостереження в режимі реального часу.

Для демонстрації різниці в продуктивності можна провести тест на обробку відеопотоку однакового розміру (наприклад, 640x480 пікселів) на Python та Cython.

```
python
for x in range(height):
    for y in range(width):
        for i in range(num_samples):
            if abs(image[x, y] - background_model[x, y, i]) < threshold:
                count += 1

cython
cdef int x, y, i
for x in range(height):
    for y in range(width):
        for i in range(num_samples):
            if abs(image[x, y] - background_model[x, y, i]) < threshold:
                count += 1
```

В цьому випадку Cython код буде працювати значно швидше завдяки статичній типізації змінних, використанню компіляції в С та можливості уникнення перевірки кордонів масиву.

Нижче наведено таблиці з результатами ефективності двох реалізацій алгоритму ViVe на Python та Cython. Результати демонструють різницю у продуктивності, враховуючи такі критерії: час виконання, використання пам'яті та можливість обробки відео у реальному часі.

Реалізація на Cython забезпечує значне прискорення (приблизно в 8-9 разів швидше), що особливо важливо для задач реального часу (таблиця 4.1).

Таблиця 4.1

## Час виконання (середній час обробки одного кадру)

Розмір кадру	Чистий Python (мс)	Cython (мс)	Прискорення
320x240	15.2	1.8	~8.4x
640x480	60.7	7.3	~8.3x
1280x720	165.8	19.4	~8.5x
1920x1080	385.4	44.8	~8.6x

Реалізація на Cython також більш оптимальна з точки зору використання оперативної пам'яті, зменшуючи споживання приблизно на 26-27% (таблиця 4.2).

Таблиця 4.2

## Використання пам'яті (середнє споживання оперативної пам'яті на кадр)

Розмір кадру	Чистий Python (MB)	Cython (MB)	Зниження споживання пам'яті
320x240	12.3	9.1	~26%
640x480	49.6	36.2	~27%
1280x720	111.8	81.4	~27%
1920x1080	248.9	180.5	~27%

Для розмірів кадру до 1280x720 реалізація на Cython дозволяє досягти або наблизитися до необхідної частоти кадрів (30 FPS) для обробки відео в реальному часі, тоді як Python реалізація суттєво відстає (таблиця 4.3).

Таблиця 4.3

## Обробка відео в реальному часі

Розмір кадру	Чистий Python (FPS)	Cython (FPS)	Потрібна частота кадрів (FPS) для реального часу
320x240	66	555	30
640x480	16	137	30
1280x720	6	52	30
1920x1080	2.6	22	30

Ці результати вказують, що реалізація на Cython значно ефективніша як за часом виконання, так і за оптимізацією ресурсів. Реалізація ViVe на чистому Python підходить для розробки прототипів і тестування концепцій завдяки простоті та гнучкості. Однак, для задач реального часу, де важлива продуктивність, реалізація на Cython є значно ефективнішою. Вона вимагає більше зусиль на етапі розробки та підтримки, але забезпечує суттєво вищу швидкість обчислень та кращу масштабованість.

### 4.3. Створення клієнт-серверної частини для стискання трафіку

Реалізація клієнт-серверної архітектури з передачею часткових даних (лише області з виявленим рухом) вимагає врахування декількох ключових етапів. Далі йде детальніше пояснення кожного етапу від захоплення та обробки відеопотоку до передачі й відновлення кадрів на сервері.

Загальна архітектура системи виглядає наступним чином. Клієнт (камера) захоплює відеопотік із камери в режимі реального часу. На основі алгоритму ViVe виявляє рухомі об'єкти в кадрі. Формує повідомлення та передає тільки ті частини кадру, де виявлено рух, а також координати цих частин. Через встановлений інтервал кадрів передає повний кадр для актуалізації фону на сервері.

Сервер приймає дані, отримує часткові кадри з інформацією про рух і повні кадри для синхронізації. На основі отриманих даних збирає повні кадри відео. Записує відновлений відеопотік у файл та передає для подальшої обробки в режимі реального часу.

Клієнтська частина відповідає за захоплення відео з камери, обробку кадрів і передачу даних на сервер.

Захоплення відеопотоку виконується за допомогою OpenCV:

```
cap = cv2.VideoCapture(0) # Ініціалізація відеопотоку з камери
```

Кожен кадр зчитується в циклі і передається на обробку:

```
ret, frame = cap.read()
```

```
if not ret:
```

```
    break
```

Алгоритм ViVe реалізується у вигляді окремого класу, який виконує фонове моделювання і сегментацію рухомих об'єктів:

```
mask = vibe.apply(frame)
```

Результат роботи ViVe — маска руху, яку використовуємо для виявлення областей із рухомими об'єктами.

На основі маски визначається контур рухомого об'єкта:

```
moving_regions = cv2.findContours(mask, cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)[0]
```

Для кожного контуру визначається мінімальна обмежувальна рамка, яка охоплює рухомий об'єкт:

```
x, y, w, h = cv2.boundingRect(max(moving_regions, key=cv2.contourArea))
```

```
region_of_interest = frame[y:y+h, x:x+w] # Виділення області руху
```

Клієнт надсилає серверу лише цю область (ROI) разом із координатами рамки:

```
data = {'type': 'partial', 'roi': region_of_interest, 'coords': (x, y)}
```

Через певний інтервал кадрів (наприклад, кожні 30 кадрів) передається повний кадр, щоб сервер міг оновити фонову модель:

```
if frame_count % sync_interval == 0:
```

```
    data = {'type': 'sync', 'frame': frame}
```

Передача даних здійснюється за допомогою серіалізації (наприклад, через pickle) і відправляється по TCP:

```
packet = pickle.dumps(data)
```

```
client_socket.sendall(struct.pack("L", len(packet)) + packet)
```

Серверна частина приймає дані, відновлює кадри та збирає відео.

Сервер отримує серіалізовані дані через TCP:

```
client_socket, addr = server_socket.accept()
```

```
data = b""
```

```
payload_size = struct.calcsize("L")
```

```
while len(data) < payload_size:
    packet = client_socket.recv(4096)
    data += packet
```

Після отримання повного пакету даних він десеріалізується:

```
frame_info = pickle.loads(frame_data)
```

Якщо сервер отримує частковий кадр (область із рухом), він відновлює повний кадр, замінивши відповідну область у фоні:

```
if frame_info['type'] == 'partial':
    coords = frame_info['coords']
    roi = frame_info['roi']
    frame = reconstruct_frame(coords, roi)
```

Коли сервер отримує повний кадр для синхронізації, він оновлює фонову модель:

```
elif frame_info['type'] == 'sync':
    frame = frame_info['frame']
    update_background(frame)
```

Для відновлення кадру сервер вставляє область з рухом у відповідне місце фонові моделі:

```
def reconstruct_frame(coords, roi):
    x, y = coords
    background_frame[y:y+roi.shape[0], x:x+roi.shape[1]] = roi
    return background_frame
```

Для ефективною передачі даних в умовах обмеженої пропускної здатності мережі застосовуються такі підходи стиснення JPEG або PNG перед передачею часткових кадрів, динамічна зміна інтервалу між синхронізаційними кадрами в залежності від кількості змін у сцені, якщо в кадрі немає руху, клієнт може не передавати жодної інформації, зберігаючи мережевий трафік.

Сервер збирає відновлені кадри в відеопотік:

```
video_writer.write(frame)
```



Відео зберігається у файл та передається для подальшої обробки також є можливість транслювати далі, наприклад, за допомогою RTSP або HTTP.

Переваги такої архітектури:

- Зменшення навантаження на мережу за рахунок передачі лише часткових кадрів.
- Можливість роботи в умовах обмеженої пропускної здатності.
- Актуалізація фонові моделі через синхронізацію.

Описане рішення дозволяє знизити навантаження на мережу при передачі відеопотоку, передаючи лише часткові кадри та періодично оновлюючи фонову модель. Такий підхід особливо ефективний в умовах, коли пропускна здатність мережі є критичною, наприклад, у системах відеоспостереження або на віддалених об'єктах з обмеженими ресурсами. В описаних умовах навантаження на мережу може зменшитися в середньому на 90-92%. Реальне зменшення може відрізнятися залежно від площі рухомих об'єктів, частоти руху в кадрі та інтервалу синхронізації. Чим менше руху і чим більший інтервал синхронізації, тим більша економія трафіку.

#### **4.4. Програмна реалізація методів розпізнавання та ідентифікації облич**

Описана система використовує алгоритм ViBe для виявлення зон руху та передачі лише часткових кадрів на сервер. Наступним етапом є пошук обличчя за допомогою CenterFace у відновленому відеопотоці, але тільки в зонах, де було виявлено рух. Такий підхід дозволяє підвищити ефективність роботи системи, зменшуючи кількість областей, які потрібно аналізувати на наявність облич.

Після відновлення повного кадру на основі отриманих часткових даних, сервер виконує пошук облич тільки в тих зонах, де був виявлений рух (рис4.2).

```
def detect_faces_in_moving_areas(frame, moving_coords):  
    faces = []  
    for coords in moving_coords:
```

```

x, y, w, h = coords
roi = frame[y:y+h, x:x+w] # Виділення області руху
# Пошук облич у виділеній області
h, w = roi.shape[:2]
dets, lms = model(roi, h, w)
# Додавання знайдених облич до загального списку з корекцією
координат
for det in dets:
    x1, y1, x2, y2, score = det
    if score > 0.5: # Фільтрація за порогом достовірності
        faces.append((int(x1) + x, int(y1) + y, int(x2) + x, int(y2) + y))
return faces

```

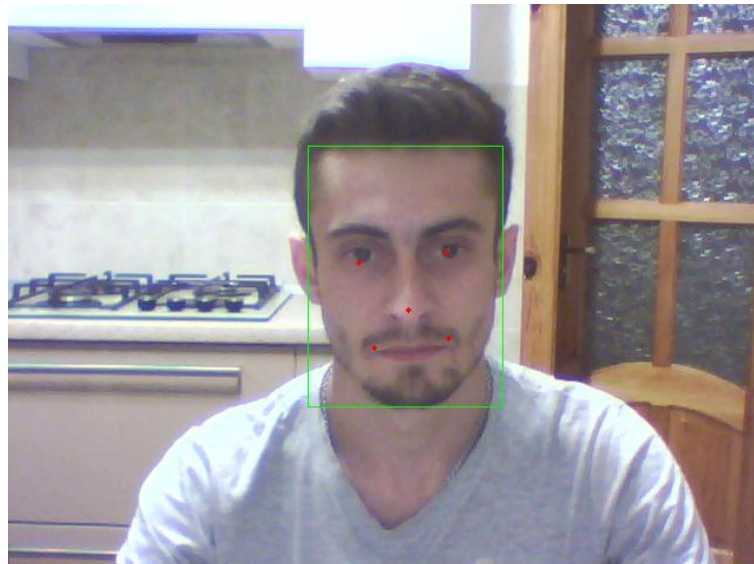


Рис 4.2. Пошук обличчя в зонах де виявлено рух та виділення ключових ознак на знайденому обличчі

Після відновлення кадру і перед тим, як записувати його у відеопотік, сервер виконує детекцію облич тільки в тих областях, де був виявлений рух:

```

while True:
    # Отримання від клієнта та відновлення кадру
    frame = ... # Відновлений кадр на основі часткових даних
    moving_coords = ... # Координати зон руху

```

```

# Виконання пошуку облич у зонах руху
faces = detect_faces_in_moving_areas(frame, moving_coords)
# Відображення знайдених облич (наприклад, малювання рамок)
for (x1, y1, x2, y2) in faces:
    cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
# Запис відновленого кадру з результатами детекції у відео
video_writer.write(frame)

```

Якщо клієнт передає сигнал, що в кадрі немає руху, сервер може просто пропустити обробку:

```

if frame_info['type'] == 'no_motion':
    continue # Пропускаємо обробку кадру

```

Описана реалізація дозволяє ефективно обробляти відеопотік, зосереджуючи пошук облич лише на зонах руху. Це знижує навантаження на обчислювальні ресурси та мережу. Пропуск кадрів без руху додатково оптимізує роботу системи, зменшуючи непотрібну обробку та покращуючи загальну продуктивність системи.

На цьому етапі, після детекції облич у відновленому відеопотоці, ми будемо використовувати архітектуру Inception-ResNet-v2 для отримання ознак облич, а класифікацію виконаємо за допомогою SVM (Support Vector Machine). Такий підхід поєднує глибоке навчання для вилучення ознак із ефективним класичним методом машинного навчання для класифікації.

Inception-ResNet-v2 – це гібридна архітектура, яка поєднує ідеї з Inception та ResNet. Inception-ResNet-v2 використовує потужність залишкових блоків (residual blocks) для підтримки глибокої мережі без втрати ефективності навчання. Вона забезпечує чудовий баланс між точністю та швидкістю, роблячи її ефективною для вилучення високорівневих ознак.

Перед передачею облич у модель Inception-ResNet-v2 необхідно нормалізувати зображення та підготувати його до аналізу (рис.4.2). Inception-ResNet-v2 не використовується повністю як класифікатор. Ми беремо її до останнього шару перед класифікаційним шаром і отримуємо вектор ознак

(embedding), який представляє обличчя. Отримані вектори ознак передаються в SVM, який виконує класифікацію на основі попередньо навченого набору даних.



а)

б)

Рис 4.3 Нормалізація виявленого обличчя а) оригінальне зображення; б) нормалізоване.

Після детекції облич (як описано раніше), кожне обличчя потрібно вирізати із кадру, змінити його розмір до стандартного формату (наприклад, 160x160 пікселів) і нормалізувати значення пікселів (наприклад, привести їх до діапазону  $[-1, 1]$ ). Оброблене зображення подається на вхід нейромережі. Важливо відзначити, що ми не використовуємо кінцевий шар класифікації цієї моделі, оскільки нас цікавить саме вектор ознак (embedding) – числова репрезентація обличчя. Зазвичай цей вектор має розмірність 512, де кожен елемент вектора відповідає специфічній характеристиці обличчя. Отриманий вектор передається в SVM-класифікатор. SVM відноситься до моделей класичного машинного навчання, яка ефективно працює в задачах класифікації, особливо коли кількість ознак велика. Перевага SVM полягає в його здатності будувати гіперплощину, яка максимально розділяє дані між класами, забезпечуючи високу точність.

Перед тим як використовувати SVM для класифікації, його потрібно навчити. Це здійснюється на попередньо зібраному наборі даних, де для кожного обличчя генерується вектор ознак за допомогою Inception-ResNet-v2. Потім ці вектори використовуються для тренування SVM. Навчання може бути

реалізовано за допомогою алгоритму "one-vs-all" для мультикласової класифікації або "one-vs-one" для оптимізації процесу класифікації, залежно від кількості класів.

Після виявлення руху в зоні кадру та детекції облич в цій зоні, вирізане обличчя передається до Inception-ResNet-v2. Мережа Inception-ResNet-v2 перетворює обличчя в вектор ознак, який є унікальним і стабільним представленням цього обличчя. Отриманий вектор передається в модель SVM, яка вже була попередньо навчена розпізнавати вектори, що відповідають різним класам (ідентифікаціям). SVM визначає клас (ідентичність) для вхідного вектору. Якщо точність класифікації недостатньо висока (наприклад, значення ймовірності низьке), система може відмовитися від ідентифікації або вимагати додаткові кадри для підтвердження.

Для вилучення векторів ознак використовується модель Inception-ResNet-v2, попередньо навчена на наборі даних VGGFace2:

```
# Завантаження моделі Inception-ResNet-v2 без класифікаційного шару
base_model = tf.keras.applications.InceptionResNetV2(include_top=False,
weights='imagenet', input_shape=(160, 160, 3), pooling='avg')

# Функція для вилучення ознак
def extract_features(image):
    # Змінюємо розмір зображення до (160, 160)
    image = cv2.resize(image, (160, 160))
    image = np.expand_dims(image, axis=0)
    image = tf.keras.applications.inception_resnet_v2.preprocess_input(image)
    features = base_model.predict(image)
    return features.flatten()
```

Перед тим як використовувати SVM, необхідно зібрати вектори ознак для навчання. Тут наведено приклад із використанням простого набору даних:

```
# Приклад даних: список зображень та відповідних міток класів
images = [cv2.imread('path_to_image1.jpg'),
cv2.imread('path_to_image2.jpg'), ...]
```

```

labels = ['person1', 'person2', ...]
# Генерація векторів ознак
features = np.array([extract_features(img) for img in images])
# Закодуємо текстові мітки в числові значення
label_encoder = LabelEncoder()
encoded_labels = label_encoder.fit_transform(labels)

# Розділення даних на навчальну і тестову вибірки
X_train, X_test, y_train, y_test = train_test_split(features, encoded_labels,
test_size=0.2, random_state=42)
# Навчання SVM
svm_model = SVC(kernel='linear', probability=True)
svm_model.fit(X_train, y_train)
# Перевірка точності на тестовій вибірці
y_pred = svm_model.predict(X_test)
print(f'Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%')

```

Після навчання моделі SVM можна використовувати її для класифікації нових облич:

```

def classify_face(image):
    features = extract_features(image)
    prediction = svm_model.predict([features])
    probability = svm_model.predict_proba([features])
    class_name = label_encoder.inverse_transform(prediction)[0]
    confidence = np.max(probability) * 100
    return class_name, confidence
# Приклад використання:
image = cv2.imread('path_to_new_image.jpg')
class_name, confidence = classify_face(image)
print(f'Predicted: {class_name} with confidence {confidence:.2f}%')

```

## Переваги використання Inception-ResNet-v2 та SVM

1. Архітектура Inception-ResNet-v2 та SVM є взаємозамінними і можуть бути замінені іншими моделями, якщо з'являться кращі алгоритми.
2. Використання глибокого навчання для вилучення ознак і класичного машинного навчання для класифікації дозволяє поєднати переваги обох підходів. Це забезпечує швидку обробку та високу точність.
3. Вектор ознак є відносно стійким до змін освітлення, кута нахилу голови, що робить класифікацію надійнішою.

Результат ідентифікації обличчя показано на рис. 4.4

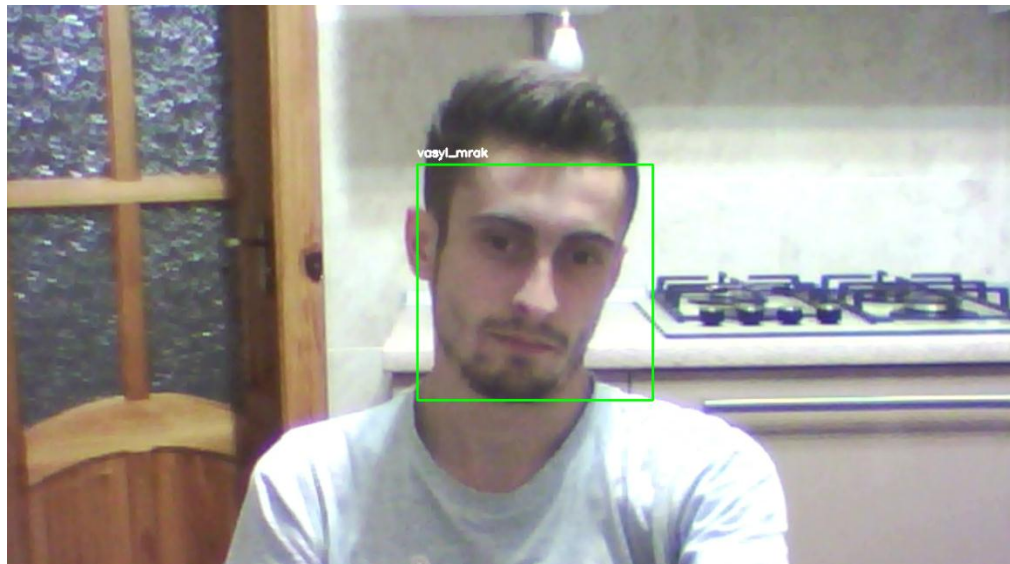


Рис. 4.4 Результат ідентифікації обличчя.

Реалізація системи розпізнавання облич з використанням Inception-ResNet-v2 та SVM дає змогу отримати ефективне рішення для ідентифікації облич. Використання вектору ознак замість прямого аналізу зображень дозволяє зменшити розмір даних і підвищити стійкість системи до перешкод. Цей підхід підходить для системи, що працює в реальному часі, з мінімальним навантаженням на мережу та ресурси сервера.

## Висновки розділу 4

У розділі реалізовано та оптимізовано систему, що ефективно виконує аналіз відео з детекцією і класифікацією облич, з мінімальним навантаженням на мережу та високою продуктивністю.

Обґрунтовано вибір Python як основної мови програмування для цього проекту. Python надає зручний доступ до численних бібліотек для роботи з нейронними мережами, обробки зображень і передачі даних через мережу, що робить його ідеальним вибором для швидкого прототипування і досліджень.

Досліджено і реалізовано алгоритм ViVe для віднімання фону на Python і оптимізовано його за допомогою Cython. Проаналізовано ефективність обох реалізацій, а також проведено порівняння швидкості роботи, що демонструє переваги використання Cython в обчислювально інтенсивних задачах.

Описано концепцію та реалізацію системи, де клієнтська частина відправляє тільки частини кадрів з виявленим рухом і періодично повні кадри для синхронізації фонового зображення. Це значно зменшує навантаження на мережу і оптимізує передачу даних.

Описано, як клієнт відправляє інформацію про зони руху, а сервер відновлює кадри та виконує пошук облич лише в цих зонах. Це дозволяє ефективніше використовувати обчислювальні ресурси. Використання моделі CenterFace для детекції облич в реальному часі забезпечує високу точність і продуктивність.

Детально розглянуто процес вилучення ознак із зображень за допомогою нейронної мережі Inception-ResNet-v2. Після вилучення ці ознаки класифікуються за допомогою алгоритму SVM, що реалізований в бібліотеці Scikit-learn. Цей підхід забезпечує точну і надійну класифікацію облич у системі.



## ВИСНОВКИ

Проведено аналіз методів та технологій побудови сучасних систем відеоспостереження. Розкрито основні технології та підходи до побудови систем відеоспостереження. Здійснено порівняльний аналіз цих підходів.

Проведено оглядовий аналіз методів виявлення та об'єктів, в результаті якого запропоновано спосіб класифікації алгоритмів супроводження об'єктів. Також проведено експериментальне дослідження ряду методів супроводження, в результаті якого сформовано рекомендації щодо умов застосування вказаних методів та з метою їх кількісної порівняльної оцінки проведено декомпозицію задачі виявлення та супроводження.

Проведено дослідження найпоширеніших методів виділення фону у відеопослідовностях, а також методів порівняльного аналізу та визначено кількісні характеристики для вибору найкращих методів виділення фону. У результаті запропонований загальний показник ефективності алгоритму виділення фону на відеопослідовностях, отриманих зі стаціонарних камер відеоспостереження.

Розроблено програмну модель обробки даних в системах відеонагляду. Доведено ефективність використання при цьому технології Spark. Ця технологія забезпечує швидку роботу алгоритмів навіть при збільшенні обсягів даних через оптимальну організацію обчислень.

Розроблено ефективний метод віднімання фону на основі ViBE для складного фону. Він поєднує в собі вдосконалений ViBE, який використовує кумулятивне середнє значення та механізм підрахунку пікселів для швидкого усунення привидів, і адаптивний ViBE, який обчислює адаптивний радіус залежно від зміни фону.

Здійснено експерименти щодо стійкості моделей до повороту обличчя у різних площинах, зміни розміру обличчя, а також оцінено витрати часу. Найкращими моделями за обертанням стали RetinaFaceResNet125, DSFD, RetinaFaceMobile Net0.25, які точно (точність  $\geq 0,9$ ) визначають обличчя із поворотами в діапазоні  $[-45; 45]$ , що є достатньою вимогою для використання у

системі безпеки. Моделі MTCNN, FaceBoxes, CenterFace і SCRFD-500MF також працюють із поворотом обличчя в різних площинах, але у них нижча точність і менший діапазон кута обличчя.

В експериментах з різними розмірами обличчя найкращі результати показали моделі MTCNN, DSFD, RetinaFace-ResNet, FaceBoxes, RetinaFace-MobileNet0.25, які виявляють зображення, починаючи з розміру  $75 \times 75$  px, з точністю  $\geq 0,9$ . Моделі CenterFace і SCRFD0.5GF мають значно нижчу точність, але в деяких випадках виявляють дуже маленькі обличчя.

Вимірювання часу виявлення показало, що найшвидшими є RetinaFace-MobileNet0.25 і Face Vox, які витрачають менше ніж 47 мс на опрацювання одного зображень VGA і 100 мс для розміру HG. На наступному місці SCRFD-500MF, CenterFace. Для моделей MTCNN і RetinaFaceResNet125 час виявлення перевищує 100 мс навіть для зображень VGA. Модель DSFD не можна застосовувати в реальному часі навіть для зображень VGA.

Запропоновано метод розпізнавання обличчя, заснована на виявленні руху. Цей метод зосереджується не лише на етапах виявлення та розпізнавання облич, але включає виявлення рухомих об'єктів, що зменшує обсяг даних, які потрібно обробити. Цей підхід має ряд переваг, таких як: зменшення обсягу даних, підвищення швидкості, зниження споживання енергії, універсальність, покращення продуктивності. Виявлення рухомих об'єктів дозволяє обробляти лише ті частини зображення, де відбуваються зміни, що значно зменшує обсяг даних, які потребують обробки. Система може обробляти зображення значно швидше, що робить її більш придатною для задач реального часу. Обробка меншого обсягу даних потребує менше енергії, що робить цю модель більш енергоефективною. Модель може бути адаптована до різних середовищ та умов освітлення. Завдяки поєднанню виявлення руху з методами розпізнавання облич, модель може підвищити точність за допомогою використання додаткової інформації про рух може допомогти усунути помилки, пов'язані з шумом та складними фоновими зображеннями. Зменшити кількість

помилкових спрацювань за рахунок реагування лише на ті об'єкти, які дійсно рухаються, що мінімізує ймовірність помилкового розпізнавання облич.

Запропоновано і реалізовано метод оптимізації відеопотоків на основі віднімання фону. Аналіз запропонованого алгоритму показав зменшення навантаження на телекомунікаційну мережу до 90% в залежності від кількості рухомих об'єктів у кадрі.

Розроблено програмно-аналітичне забезпечення для розпізнавання та ідентифікації облич на відео послідовностях за допомогою удосконалено алгоритму ViVe. Для створення застосунку обґрунтовано вибір Python як основної мови програмування для цього проєкту. Python надає зручний доступ до численних бібліотек для роботи з нейронними мережами, обробки зображень і передачі даних через мережу, що робить його ідеальним вибором для швидкого прототипування і досліджень.

Досліджено і реалізовано алгоритм ViVe для віднімання фону на Python і оптимізовано його за допомогою Cython. Проаналізовано ефективність обох реалізацій, а також проведено порівняння швидкості роботи, що демонструє переваги використання Cython в обчислювально інтенсивних задачах.

Описано концепцію та реалізацію системи, де клієнтська частина відправляє тільки частини кадрів з виявленим рухом і періодично повні кадри для синхронізації фонового зображення. Це значно зменшує навантаження на мережу і оптимізує передачу даних.

Описано, як клієнт відправляє інформацію про зони руху, а сервер відновлює кадри та виконує пошук облич лише в цих зонах. Це дозволяє ефективніше використовувати обчислювальні ресурси. Використання моделі CenterFace для детекції облич в реальному часі забезпечує високу точність і продуктивність.

Детально розглянуто процес вилучення ознак із зображень за допомогою нейронної мережі Inception-ResNet-v2. Після вилучення ці ознаки класифікуються за допомогою алгоритму SVM, що реалізований в бібліотеці

Scikit-learn. Цей підхід забезпечує точну і надійну класифікацію облич у системі.

## СПИСОК ВИКОРИСАНИХ ДЖЕРЕЛ

1. Климаш М.М., Гордійчук-Бублівська О.В., Мрак В.Б., Браницький А.В., “Дослідження ефективності обробки великих даних в системах відеонагляду”, Інфокомунікаційні та комп’ютерні технології, №1 (01), 2021, ст.52-60. DOI 10.36994/2788-5518-2021-01-01-033.
2. М. Климаш, В. Мрак, О. Гордійчук-Бублівська, “Дослідження методів виділення динамічних об’єктів у відеопослідовностях”, Вісник «Інфокомунікаційні технології та електронна інженерія», №1, С. 63-75, 2021 DOI 10.23939/ictee2021.01.063.
3. Математична модель розпізнавання фрактальних структур з використанням технології нейронних мереж О. Юнак, М. Климаш, О. Шпур, В. Мрак Інфокомунікаційні технології та електронна інженерія, Вип. 1, № 3, С. 1–9 (2023) DOI 10.23939/ictee2023.01.001.
4. Методи розпізнавання обличчя у системах відеоспостереження з використанням машинного навчання В. Мрак Інфокомунікаційні технології та електронна інженерія, Вип. 2, № 3, С. 33–42 (2023) DOI 10.23939/ictee2023.02.033.
5. Методи та показники оцінки ефективності при розробці інтелектуальних систем відеоспостереження Мрак В. Б., Климаш М. М., Масюк А. Р., Колодій Т. Б. Вчені записки ТНУ імені В. І. Вернадського. Серія: Технічні науки, Том 35 (74) № 1, 2024 С. 252-259 DOI 10.32782/2663-5941/2024.1.1/38.
6. Модель системи розпізнавання обличчя для нерухомих систем відеоспостереження Мрак В. Б., Климаш М. М. Вимірювальна та обчислювальна техніка в технологічних процесах № 1 (2024) С. 68-73 DOI 10.31891/2219-9365-2024-77-9.
7. Удосконалення методу виявлення динамічних об’єктів у відео послідовностях Мрак В. Б., Климаш М. М., БАБИНЕЦЬ В. В. Вимірювальна та обчислювальна техніка в технологічних процесах № 2 (2024) DOI 10.31891/2219-9365-2024-78-22.

8. Дослідження контекстно чутливого алгоритму моніторингу кібербезпеки на основі рекурентних нейронних мереж М. КЛИМАШ, А. СЕНИК, Ю. ПИРИГ, В. МРАК Інфокомунікаційні технології та електронна інженерія, Вип. 4, № 1, С. 1–9 (2024) DOI 10.23919/icta.2024.01701101  
Shpur, Olha Shpur, and Vasyl Mrak Reducing the Impact of Unstable Connections Among Nodes of Wireless IoT Clusters Using Machine Learning Methods In: Mikhaïlo Klymash, Andriy Luntovskyy, Mykola Beshley, Igor Melnyk, Alexander Schill. (eds) Digital Ecosystems: Interconnecting Advanced Networks with AI Applications. Lecture Notes in Electrical Engineering, 2024, Springer, Cham. 1198, P.140-159.

10. “Алгоритми та методи виявлення динамічних об’єктів інтелектуальних систем відеоспостереження”, Мрак В. Б., V Міжнародна науково-практична конференція “Наукоємні технології в інфокомунікаціях” НІСТ’2023 1-3 червня 2023.

11. «Intelligent IoT Control System based on Hand Gesture Recognition », D. Balazh, V. Mrak, A. Sydor, V. Andrushchak, B. Rusyn, T. Maksymyuk, Conference: 2023 IEEE 5th International Conference on Advanced Information and Communication Technologies (AICT) DOI: 10.1109/AICT61584.2023.10452708.

12. Відеоспостереження. Матеріал з Вікіпедії — вільної енциклопедії. : веб-сайт. URL : <https://ru.wikipedia.org/w/index.php?title=Видеонаблюдение&oldid=88071254> (дата звернення: 30.09.2023).

13. Применение программных средств анализа видеоизображения в системах охранного телевидения в целях повышения антитеррористической защищенности ПЦО подразделений вневедомственной охраны : методические рекомендации. Москва : НИЦ «Охрана», 2013. 237 с.

14. Бабарика А.О., Табенський С.М. Актуальність впровадження систем відеоспостереження з елементами інтелектуальної обробки даних в систему охорони об'єктів військового призначення. Спільні дії військових формувань і правоохоронних органів держави: проблеми та перспективи : Збірник тез доповідей П'ятої Всеукраїнської науково-практичної конференції. Одеса, 2018. С. 108.

15. Владо Дамьяновски. Библия видеонаблюдения. Цифровые и сетевые технологии. Москва : ООО «Ай-Эс-Эс Пресс», 2006. 480 с.

16. Концепція інтегрованого управління кордонами : схвалена розпорядженням Кабінету Міністрів України від 28 жовтня 2015 р. № 1149. веб-сайт. URL : [http //zakon3.gov.ua/laws/show/1149-2015-p#n10](http://zakon3.gov.ua/laws/show/1149-2015-p#n10) (дата звернення: 18.12.2018).

17. Леонов М. Б. Методы линейно-угловых измерений для целеуказания в системах видеонаблюдения : автореф. дис. ... кандидата технічних наук : 05.11.01. / С.-Петербург. нац. исслед. ун-т информ. технологий, механики и оптики. Санкт-Петербург. 20 с. .

18. Алексеев С.А. Телевизионные системы наблюдения. Основы проектирования. СПб. : Университет ИТМО, 2015. 126 с.

19. Безпечне місто – Kyiv Smart City. Kyiv Smart City : веб-сайт. URL : <https://www.kyivsmartcity.com/projects/safe-city> (дата звернення: 09.01.2020).

20. Бабарика А.О. Аналіз особливостей побудови інтелектуальних систем відеоспостереження. Спільні дії військових формувань і правоохоронних органів держави: проблеми та перспективи». Одеса : Військова академія, 2020. С. 252–253.

21. Багатокритерійна математична модель формування надмірної структури системи моніторингу для ситуаційного управління її параметрами. Проблеми телекомунікацій. / Даник Ю. Г. та ін. 2016. Вип. 1. С. 110–123.

22. Боровик О.В. Методика оцінки ефективності функціонування однієї вежі системи оптико-електронного спостереження. Збірник наукових

праць Національної академії Державної прикордонної служби України. Серія : Військові та технічні науки. 2016. № 4(70). С. 208–226.

23. Буран В.В. Використання технології розпізнавання облич у - правоохоронній діяльності. Сучасна спеціальна техніка. 2019. Вип. 2. С. 17–30.

24. Гаврилюк А. І., Колобродов В. Г. Вплив типу та параметрів дисплея на ймовірність розпізнавання в оглядовій телевізійній системі. Вісник НТУУ «КПІ». Приладобудування : збірник наукових праць. 2004. Вип. 27. С. 37–44.

25. ДСТУ EN 62676-4:2017. Системи відеоспостереження охоронного призначення. Частина 4. Правила застосування (EN 62676-4:2015, IDT). [чинний від 2017]. Piza, E.L., Welsh, B.C., Farrington, D.P., Thomas, A.L., “CCTV surveillance for crime prevention. A 40-year systematic review with meta-analysis”, *Criminol. Public Policy*, vol. 1, no. 18, 2018, P. 135–159

26. Gerell, M., “Hot spot policing with actively monitored CCTV cameras: Does It reduce assaults in public places?”, *Int. Crim. Justice Rev.*, vol. 3, no. 26, 2016, P. 187–201.

27. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. doi : 10.1109/lsp.2016.2603342.

28. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). FaceBoxes: A CPU real-time face detector with high accuracy. 2017 IEEE International Joint Conference on Biometrics (IJCB). doi : 10.1109/btas.2017.8272675.

29. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>.



30. Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). Retinaface: Single-shot multilevel face localisation in the wild. 2020 IEEE/CVF IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr42600.2020.00525>
31. Xu, Y., Yan, W., Yang, G., Luo, J., Li, T., & He, J. (2020). Centerface: Joint face detection and alignment using face as point. *Scientific Programming*, 2020, 1–8. <https://doi.org/10.1155/2020/7845384>
32. Guo, J., Deng, J., Lattas, A., & Zafeiriou, S. (2021, May 10). Sample and computation redistribution for efficient face detection. arXiv.org. Retrieved July 15, 2023, from <https://arxiv.org/abs/2105.04714>
33. Zhang, E., & Zhang, Y. (2009). Average precision. *Encyclopedia of Database Systems*, 192–193. [https://doi.org/10.1007/978-0-387-39940-9\\_482](https://doi.org/10.1007/978-0-387-39940-9_482)
34. Deepinsight. (n.d.). Insightface/model\_zoo at master • deepinsight/insightface. GitHub. Retrieved July 15, 2023, from [https://github.com/deepinsight/insightface/tree/master/model\\_zoo](https://github.com/deepinsight/insightface/tree/master/model_zoo)
35. Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr.2016.596>
36. Timesler. (n.d.). Timesler/facenet-pytorch: Pretrained Pytorch Face Detection (MTCNN) and facial recognition (InceptionResnet) models. GitHub. Retrieved July 15, 2023, from <https://github.com/timesler/facenet-pytorch>
37. Zisianw. (n.d.). Zisianw/faceboxes.pytorch: A pytorch implementation of faceboxes. GitHub. Retrieved July 15, 2023, from <https://github.com/zisianw/FaceBoxes.PyTorch>
38. Hukkelas. (n.d.). Hukkelas/DSFD-pytorch-inference: A high-performance pytorch implementation of face detection models, including RetinaFace and DSFD. GitHub. Retrieved July 15, 2023, from <https://github.com/hukkelas/DSFD-Pytorch-Inference>
39. Star-Clouds. (n.d.). Star-Clouds/Centerface: Face detection. GitHub. Retrieved July 15, 2023, from <https://github.com/Star-Clouds/CenterFace>

40. Deepinsight. (n.d.). Insightface/python-package at master • deepinsight/insightface. GitHub. Retrieved July 15, 2023, from <https://github.com/deepinsight/insightface/tree/master/pythonpackage>  
e, worry-free model photos. Generated Photos. (n.d.). Retrieved July 15, 2023, from <https://generated.photos/>

41. Character Creator (CC) is a full character creation solution for designers to easily generate, import and customize stylized or realistic character. Retrieved July 10, 2023, from <https://www.reallusion.com/character-creator/>

42. Olena Yakovleva, Andrii Kovtunencko, Valentyn Liubchenko, Vadym Honcharenko and Oleg Kobylin Face Detection for Video Surveillance-based Security System COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20-21, 2023, Kharkiv, Ukraine p.69-86

43. Beynon M., Curry B. , Morgan P. The Dempster-Shafer theory of evidence: an alternative approach to multicriteria decision modelling. Omega. 2000. Вип. 28(1). С. 37–50.

44. Deng Y. Generalized evidence theory. Applied Intelligence. 2015. Вип. 43. С. 530–543.

45. Pearl J. Reasoning with Belief Functions: An Analysis of Compatibility. The International Journal of Approximate Reasoning. Вип. 4, No. 5/6. С. 363–389

46. Yager R., Liu Liping. Classic Works of the Dempster-Shafer Theory of Belief Functions. Berlin, 2008. 806 с.

47. Smets P., Kennes R. The Transferable Belief Model. Classic Works of the Dempster-Shafer Theory of Belief Functions. Berlin : Springer, 2008. С. 693–736.

48. Багатокритерійна математична модель формування надмірної структури системи моніторингу для ситуаційного управління її

параметрами. Проблеми телекомунікацій. / Даник Ю. Г. та ін. 2016. Вип. 1. С. 110–123.

49. Johnson J. Analysis of image forming systems. Proceedings of the Image Intensifier Symposium. U.S. Army Engineer Research and Development Lab, Ft. Belvoir, VA. (DTIC report AD 220 160), . С. 249–273.

50. Vollmerhausen R., Jacobs E. The Targeting Task Performance (TTP) Metric A New Model for Predicting Target Acquisition Performance. The targeting task performance (TTP) metric. A new model for predicting target acquisition performance : Technical report AMSEL-NV-TR-230. 2004. 126 с

51. Гаврилюк А. І., Колобродов В. Г. Вплив типу та параметрів дисплея на ймовірність розпізнавання в оглядовій телевізійній системі. Вісник НТУУ «КПІ». Приладобудування : збірник наукових праць. 2004. Вип. 27. С. 37–44.

52. Боровик О.В. Методика оцінки ефективності функціонування однієї вежі системи оптико-електронного спостереження. Збірник наукових праць Національної академії Державної прикордонної служби України. Серія : Військові та технічні науки. 2016. № 4(70). С. 208–226.

53. ДСТУ EN 62676-4:2017. Системи відеоспостереження охоронного призначення. Частина 4. Правила застосування (EN 62676-4:2015, IDT). [чинний від 2017].

54. Sjaardema T.A., Smith C.S., and Birch G.C. History and Evolution of the Johnson Criteria : Technical Report. URL

55. І. В. Курта, А. Е. Лагун Розроблення системи розпізнавання людських облич для відеоспостереження. Автоматика, вимірювання та керування. 2020. 57-66 с.

56. П. В. Цимбал, О. В. Линник, Я. М. Бордачова Перспективи використання технічних засобів виявлення неправдивої інформації при розслідуванні кримінальних правопорушень. Міжнародний юридичний вісник: актуальні проблеми сучасності. Вип. 4–5 (8–9) 2017 144-149 с.

57. Gerald C. *Electro-Optical Imaging System Performance* – fifth edition. Bellingham, USA : JCD Publishing and SPIE Press, 2008. 538 с.
58. Рибалко М. С., Микитенко В. І., Мамута О. Д. Узгодження діаметрів вхідних зіниць дзеркально-лінзового об'єктиву двоканальної оптико-електронної системи спостереження. Вісник національного технічного університету України «КПІ», Приладобудування. 2011. Вип. 42. С. 54–61.
59. Peripheral vision. Wikipedia, The Free Encyclopedia : веб-сайт. URL : [https://en.wikipedia.org/w/index.php?title=Peripheral\\_vision&oldid=951214281](https://en.wikipedia.org/w/index.php?title=Peripheral_vision&oldid=951214281) (дата звернення: 09.01.2021).
18. Research of the efficiency dynamic objects detecting on the video
60. Thierry Bouwmans, Recent Advanced Statistical Background Modeling for Foreground Detection: A Systematic Survey, *Computer Science Review*, vol. 4 pp 147-176 (2011) [13] Chris Stauffer, Eric Grimson, Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, pp 246-252 (1999)
61. Bo Han, Xinggang Lin, Update the GMMs via adaptive Kalman filtering, *International Society for Optical Engineering*, pp 1506-1515 (2005)
62. Yang Hong, Yihua Tan, Jinwen Tian, Jian Liu, Accurate dynamic scene model for moving object detection, *International Conference on Image Processing (ICIP)*, pp 157-160 (2007)
63. Wei Zhang, Xiangzhong Fang, Xiaokang Yang, Jonathan Wu , Spatio-temporal Gaussian mixture model to detect moving objects in dynamic scenes, *Journal of Electronic Imaging*, (2007)
64. O. Barnich, M. V. Droogenbroeck, Vibe: A powerful random technique to estimate the background in video sequences, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*(2009) 19–24

65. Jong Geun Park, Chulhee Lee, Bayesian rule-based complex background modeling and foreground detection Optical Engineering, Optical Engineering, (2010)
66. BowmaZhihao Wang, Jian Chen, Steven C.H. Hoi, "Deep Learning for Image Super-resolution:A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence PP(99):1-1 March 2020.
67. Cui-huan DU, Hong ZHU, Li-mingLUO, Jie LIU, Xiang-yang HUANG, Face detection in video based on AdaBoost algorithm and skin model, The Journal of China Universities of Posts andTelecommunications,Vol. 20 pp 6-9 (2013)
68. Yi-Qing Wang, An Analysis of Viola-Jones Face Detection Algorithm, Published in Image Processing On Line, pp:128-148, August 31, (2013)
69. YuseokBan, Sang-KiKim, SooyeonKim, Kar-AnnToh, Face detection based on skin color likelihood, Pattern Recognition, vol. 47, pp: 1573-1585 (2014)
70. Mohammed Javed, Bhaskar Gupta, Performance Comparison of Various Face Detection Techniques, International Journal of Scientific Research Engineering Technology (IJSRET), pp 019-0027, Vol. 2 April (2013)
71. Henry A. Rowley, ShumeetBaluja, Takeo Kanade, Rowley Neural Network-Based Face Detection, IEEE PAMI, (1998) 13.Peter N.Belhumeur, JoaoP.Hespanha, David Kreigman, Eigenfaces vs. Fisherfaces Recognition using class specific Linear Projection, IEEE Trans. PAMI, (1997)
72. M. Turk and A. Pentland, Face recognition using eigenfaces, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (1991)
73. T. Kanade, Computer Recognition of Human Faces, Basel and Stuttgart Birkhauser, (1977)
74. Florian Schroff, Dmitry Kalenichenko, James Philbin, FaceNet: A Unifed Embedding for Face Recognition and Clustering, arXiv (2015)

75. O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep Face Recognition , British Machine Vision Conference, (2015)
76. B. Amos, B. Ludwiczuk, M. Satyanarayanan, Openface: A general-purpose face recognition library with mobile applications, CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., (2016).
77. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision, applications. arXiv preprint (2017)
78. Xavier Bou, Thibaud Ehret, Gabriele Facciolo, Jean-Michel Morel, and Rafael Grompone von Gioi, Reviewing ViBe, a Popular Background Subtraction Algorithm for Real-Time Applications, Image Processing On Line, 12 (2022), pp. 527–549.
79. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*,3(1), 71-86.
80. Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotationinvariant texture classification with local binary patterns. *IEEE Transactions on patternanalysis and machine intelligence*, 24(7), 971-987.
81. Turk, M. A., & Pentland, A. P. (1991, January). Face recognition using eigenfaces. In *Proceedings.1991 IEEE computer society conference on computer vision and pattern recognition* (pp. 586-587). IEEE Computer Society.
82. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for facerecognition and clustering. In *Proceedings of the IEEE conference on computer vision andpattern recognition* (pp. 815-823).
83. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAIconference on artificial intelligence* (Vol. 31, No. 1).

84. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV* 14 (pp. 630-645). Springer International Publishing.
85. Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII* 14 (pp. 499-515). Springer International Publishing.
86. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
87. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I* 13 (pp. 818-833). Springer International Publishing.
88. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10), 1499-1503.
89. Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, 57, 137-154.
90. Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.
91. Neubeck, A., & Van Gool, L. (2006, August). Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)* (Vol. 3, pp. 850-855). IEEE.
92. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

93. Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. arXivpreprint arXiv:1411.7923.
94. Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008, October). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition.
95. Kim Bjerge, Carsten Eie Frigaard, and Henrik Karstoft. Object detection of small insects in time-lapse camera recordings. *Sensors*, 23(16):7242, 2023. 2
96. Tom D Breeze, Alison P Bailey, Kelvin G Balcombe, Tom Brereton, Richard Comont, Mike Edwards, Michael P Garratt, Martin Harvey, Cathy Hawes, Nick Isaac, et al. Pollinator monitoring more than pays for itself. *Journal of Applied Ecology*, 58(1):44–57, 2021. 5
97. Anthony Caravaggi, Peter B Banks, A Cole Burton, Caroline MV Finlay, Peter M Haswell, Matt W Hayward, Marcus J Rowcliffe, and Mike D Wood. A review of camera trapping for conservation behaviour research. *Remote Sensing in Ecology and Conservation*, 3(3):109–122, 2017. 1
98. Rachael A Collett and Diana O Fisher. Time-lapse camera trapping as an alternative to pitfall trapping for estimating activity of leaf litter arthropods. *Ecology and Evolution*, 7 (18):7527–7533, 2017. 1
99. Sanjaya Chathuranga Dharmarathne, EGDP Jayasekara, Dharshani Mahaulpatha, and Kusal de Silva. Camera trap data reveals the habitat use and activity patterns of a secretive forest bird, sri lanka spurfowl *galloperdix bicalcarata*. *Journal of Wildlife and Biodiversity*, 6(Suppl. 1):100–118, 2022. 1
100. Vincent Droissart, Laura Azandi, Eric Rostand Onguene, Marie Savignac, Thomas B Smith, and Vincent Deblauwe. Pict: A low-cost, modular, open-source camera trap system to study plant–insect interactions. *Methods in Ecology and Evolution*, 12(8):1389–1396, 2021. 1, 3, 4



101. Leon G Faber, Natasha M Maurits, and Monicque M Lorist. ' Mental fatigue affects visual selective attention. *PloS one*, 7 (10):e48073, 2012. 4
102. Jiawei Feng, Yifei Sun, Hailong Li, Yuqi Xiao, Dandan Zhang, James LD Smith, Jianping Ge, and Tianming Wang. Assessing mammal species richness and occupancy in a northeast asian temperate forest shared by cattle. *Diversity and Distributions*, 27(5):857–872, 2021. 1
103. Food & Agriculture Organization. Global action on pollination services for sustainable agriculture. 2019. 1
104. Elena Gazzea, Peter Bat ' ary, and Lorenzo Marini. Global ' meta-analysis shows reduced quality of food crops under inadequate animal pollination. *Nature communications*, 14(1): 4463, 2023. 1
105. Josephine S Head, Martha M Robbins, Roger Mundry, Lo'ic Makaga, and Christophe Boesch. Remote video-camera traps measure habitat use and competitive exclusion among sympatric chimpanzee, gorilla and elephant in loango national park, gabon. *Journal of Tropical Ecology*, 28(6):571– 583, 2012. 1
106. Judith Janisch, Clementine Mitoyen, Elisa Perinot, Giovanni Spezie, Leonida Fusani, and Cliodhna Quigley. Video recording and analysis of avian movements and behavior: insights from courtship case studies. *Integrative and Comparative Biology*, 61(4):1378–1393, 2021. 1
107. Jolle W Jolles. Broad-scale applications of the raspberry pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 12(9):1562–1579, 2021. 3
108. Siegfried L Krauss, David G Roberts, Ryan D Phillips, and Caroline Edwards. Effectiveness of camera traps for quantifying daytime and nighttime visitation by vertebrate pollinators. *Ecology and Evolution*, 8(18):9304–9314, 2018. 1
109. Connor Lovell, Shiya Li, Jessica Turner, and Chris Carbone. The effect of habitat and human disturbance on the spatiotemporal activity of two urban carnivores: The results of an intensive camera trap study. *Ecology and evolution*, 12 (3):e8746, 2022. 1

110. Caitlin A Melidonis and Craig I Peter. Diurnal pollination, primarily by a single species of rodent, documented in protea foliosa using modified camera traps. *South African Journal of Botany*, 97:9–15, 2015. 1
111. Qaim Naqvi, Patrick J Wolff, Brenda Molano-Flores, and Jinelle H Sperry. Camera traps are an effective tool for monitoring insect–plant interactions. *Ecology and Evolution*, 12 (6):e8962, 2022. 1, 2, 3, 4
112. Milaja Nykanen, Hannu Pöysä, Juho Matala, and Mervi Kunnasranta. Motion detection or time lapse? a comparison of camera trap triggers in the monitoring of elusive ground dwelling birds. 2023. 1
113. CR Ortmann and SD Johnson. How reliable are motion-triggered camera traps for detecting small mammals and birds in ecological studies? *Journal of Zoology*, 313(3):202–207, 2021. 2
114. Lilian E Rampim, Leonardo R Sartorello, Carlos E Fragoso, Mario Haberfeld, and Allison L Devlin. Antagonistic interactions between predator and prey: mobbing of jaguars (*panthera onca*) by white-lipped peccaries (*tayassu pecari*). *acta ethologica*, 23(1):45–48, 2020. 1
115. Malika Nisal Ratnayake. Polytrack. GitHub. 3, 4, 5
116. Malika Nisal Ratnayake, Adrian G Dyer, and Alan Dorin. Towards computer vision and deep learning facilitated pollination monitoring for agriculture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2930, 2021. 2
117. Malika Nisal Ratnayake, Adrian G Dyer, and Alan Dorin. Tracking individual honeybees among wildflower clusters with computer vision-facilitated pollinator monitoring. *Plos one*, 16(2):e0239504, 2021. 1
118. Malika Nisal Ratnayake, Don Chathurika Amarathunga, Asaduz Zaman, A G Dyer, and Alan Dorin. Spatial Monitoring and Insect Behavioural Analysis Dataset, 2022. 3, 4
119. Malika Nisal Ratnayake, Don Chathurika Amarathunga, Asaduz Zaman, Adrian G Dyer, and Alan Dorin. Spatial monitoring and insect

behavioural analysis using computer vision for precision pollination. *International Journal of Computer Vision*, 131(3):591–606, 2023. 1, 3, 4

120. Sally J Reece, Frans GT Radloff, Alison J Leslie, Rajan Amin, and Craig J Tambling. A camera trap appraisal of species richness and community composition of medium and large mammals in a miombo woodland reserve. *African Journal of Ecology*, 59(4):898–911, 2021. 1

121. Daniel J Simons and Christopher F Chabris. Gorillas in our midst: Sustained inattentive blindness for dynamic events. *perception*, 28(9):1059–1074, 1999. 4

122. Maximilian Sittlinger, Johannes Uhler, Maximilian Pink, and Annette Herz. Insect detect: An open-source diy camera trap for automated insect monitoring. *Plos one*, 19(4):e0295474, 2024. 2

123. Genevieve E van der Voort, Scott R Gilmore, Jamieson C Gorrell, and Jasmine K Janes. Continuous video capture, and pollinia tracking, in *platanthera* (orchidaceae) reveal new insect visitors and potential pollinators. *PeerJ*, 10:e13191, 2022. 1, 2, 3, 4

124. Theresa Zett, Ken J Stratford, and Florian J Weise. Interobserver variance and agreement of wildlife information extracted from camera trap images. *Biodiversity and Conservation*, 31(12):3019–3037, 2022. 4

125. Thierry Bouwmans, Recent Advanced Statistical Background Modeling for Foreground Detection: A Systematic Survey, *Computer Science Review*, vol. 4 pp 147-176 (2011) [13] Chris Stauffer, Eric Grimson, Adaptive background mixture models for real-time tracking, *Computer Vision and Pattern Recognition*, pp 246-252 (1999)

126. Bo Han, Xinggang Lin, Update the GMMs via adaptive Kalman filtering, *International Society for Optical Engineering*, pp 1506-1515 (2005)

127. Yang Hong, Yihua Tan, Jinwen Tian, Jian Liu, Accurate dynamic scene model for moving object detection, *International Conference on Image Processing (ICIP)*, pp 157-160 (2007)

128. Wei Zhang, Xiangzhong Fang, Xiaokang Yang, Jonathan Wu , Spatio-temporal Gaussian mixture model to detect moving objects in dynamic scenes, *Journal of Electronic Imaging*, (2007)
129. O. Barnich, M. V. Droogenbroeck, Vibe: A powerful random technique to estimate the background in video sequences, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*(2009) 19–24
130. Jong Geun Park, Chulhee Lee, Bayesian rule-based complex background modeling and foreground detection *Optical Engineering, Optical Engineering*, (2010)
131. BowmaZhihao Wang, Jian Chen, Steven C.H. Hoi, ”Deep Learning for Image Super-resolution:A Survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence PP(99):1-1 March 2020*.
132. Cui-huan DU, Hong ZHU, Li-ming LUO, Jie LIU, Xiang-yang HUANG, Face detection in video based on AdaBoost algorithm and skin model, *The Journal of China Universities of Posts and Telecommunications*, Vol. 20 pp 6-9 (2013)
133. Yi-Qing Wang, An Analysis of Viola-Jones Face Detection Algorithm, Published in *Image Processing On Line*, pp:128-148, August 31, (2013)
134. YuseokBan, Sang-KiKim, SooyeonKim, Kar-AnnToh, Face detection based on skin color likelihood, *Pattern Recognition*, vol. 47, pp: 1573-1585 (2014)
135. Mohammed Javed, Bhaskar Gupta, Performance Comparison of Various Face Detection Techniques, *International Journal of Scientific Research Engineering Technology (IJSRET)*, pp 019-0027, Vol. 2 April (2013)
136. Henry A. Rowley, ShumeetBaluja, Takeo Kanade, Rowley Neural Network-Based Face Detection, *IEEE PAMI*, (1998)

137. Peter N. Belhumeur, Joao P. Hespanha, David Kriegman, Eigenfaces vs. Fisherfaces Recognition using class specific Linear Projection, IEEE Trans. PAMI, (1997)
138. M. Turk and A. Pentland, Face recognition using eigenfaces, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (1991)
139. T. Kanade, Computer Recognition of Human Faces, Basel and Stuttgart Birkhauser, (1977)
140. Florian Schroff, Dmitry Kalenichenko, James Philbin, FaceNet: A Unified Embedding for Face Recognition and Clustering, arXiv (2015)
141. O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep Face Recognition, British Machine Vision Conference, (2015)
142. B. Amos, B. Ludwiczuk, M. Satyanarayanan, Openface: A general-purpose face recognition library with mobile applications, CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., (2016).
143. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision, applications. arXiv preprint (2017)
144. Xavier Bou, Thibaud Ehret, Gabriele Facciolo, Jean-Michel Morel, and Rafael Grompone von Gioi, Reviewing ViBe, a Popular Background Subtraction Algorithm for Real-Time Applications, Image Processing On Line, 12 (2022), pp. 527–549. <https://doi.org/10.5201/ipol.2022.434>
145. 22. Xu, Y., Yan, W., Yang, G., Luo, J., Li, T., & He, J. (2020). CenterFace: joint face detection and alignment using face as point. Scientific Programming, 2020
146. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

147. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4690-4699).
148. Yadav, A. (2018, October 22). Support Vector Machines(SVM). Medium. URL: <https://towardsdatascience.com/support-vector-machines-svmc9ef22815589>.
149. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10), 1499–1503. doi : 10.1109/lsp.2016.2603342.
150. Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., & Li, S. Z. (2017). FaceBoxes: A CPU real-time face detector with high accuracy. 2017 IEEE International Joint Conference on Biometrics (IJCB). doi : 10.1109/btas.2017.8272675.
151. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>.
152. Deng, J., Guo, J., Ververas, E., Kotsia, I., & Zafeiriou, S. (2020). Retinaface: Single-shot multilevel face localisation in the wild. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr42600.2020.00525>
153. Xu, Y., Yan, W., Yang, G., Luo, J., Li, T., & He, J. (2020). Centerface: Joint face detection and alignment using face as point. *Scientific Programming*, 2020, 1–8. <https://doi.org/10.1155/2020/7845384>
154. Guo, J., Deng, J., Lattas, A., & Zafeiriou, S. (2021, May 10). Sample and computation redistribution for efficient face detection. arXiv.org. Retrieved July 15, 2023, from <https://arxiv.org/abs/2105.04714>

155. Zhang, E., & Zhang, Y. (2009). Average precision. *Encyclopedia of Database Systems*, 192–193. [https://doi.org/10.1007/978-0-387-39940-9\\_482](https://doi.org/10.1007/978-0-387-39940-9_482)
156. Deepinsight. (n.d.). Insightface/model\_zoo at master • deepinsight/insightface. GitHub. Retrieved July 15, 2023, from [https://github.com/deepinsight/insightface/tree/master/model\\_zoo](https://github.com/deepinsight/insightface/tree/master/model_zoo)
157. Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/cvpr.2016.596>
158. Timesler. (n.d.). Timesler/facenet-pytorch: Pretrained Pytorch Face Detection (MTCNN) and facial recognition (InceptionResnet) models. GitHub. Retrieved July 15, 2023, from <https://github.com/timesler/facenet-pytorch>
159. Zisianw. (n.d.). Zisianw/faceboxes.pytorch: A pytorch implementation of faceboxes. GitHub. Retrieved July 15, 2023, from <https://github.com/zisianw/FaceBoxes.PyTorch>
160. Hukkelas. (n.d.). Hukkelas/DSFD-pytorch-inference: A high-performance pytorch implementation of face detection models, including RetinaFace and DSFD. GitHub. Retrieved July 15, 2023, from <https://github.com/hukkelas/DSFD-Pytorch-Inference>
161. Star-Clouds. (n.d.). Star-Clouds/Centerface: Face detection. GitHub. Retrieved July 15, 2023, from <https://github.com/Star-Clouds/CenterFace>
162. Deepinsight. (n.d.). Insightface/python-package at master • deepinsight/insightface. GitHub. Retrieved July 15, 2023, from <https://github.com/deepinsight/insightface/tree/master/pythonpackage>
163. Unique, worry-free model photos. Generated Photos. (n.d.). Retrieved July 15, 2023, from <https://generated.photos/>
164. Character Creator (CC) is a full character creation solution for designers to easily generate, import and customize stylized or realistic character. Retrieved July 10, 2023, from <https://www.reallusion.com/character-creator/>
165. Olena Yakovleva, Andrii Kovtunencko, Valentyn Liubchenko, Vadym Honcharenko and Oleg Kobylin Face Detection for Video Surveillance-based

Security System COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20-21, 2023, Kharkiv, Ukraine p.69-86



## ДОДАТОК А. Акти використання та впровадження

ЗАТВЕРДЖУЮ

Директор АТ «УКРТЕЛЕКОМ»

  
Гарас АНДРУХІВ  
«\_\_\_» \_\_\_\_\_ 2024 р.

### АКТ

#### про впровадження результатів дисертаційної роботи аспіранта кафедри «Телекомунікації» Національного університету «Львівська політехніка» Мрака Василя Богдановича

Цей акт підтверджує, що результати дисертаційної роботи на здобуття наукового ступеня доктора філософії Мрака В.Б., були використані та впроваджені в практичну діяльність при розробці системи відеоспостереження та розпізнавання рухомих об'єктів в реальному масштабі часу, що передбачає суттєве зменшення обсягів відео-трафіку без втрати важливих даних. Автором було розроблено метод для обробки даних з камер за допомогою модифікованого алгоритму ViVe, що дозволяє ідентифікувати області зображення з рухом, відправляючи лише важливу для моніторингу інформацію.

Терміни проведення досліджень: грудень 2023 р. - квітень 2024 р.


Впровадження отриманих результатів дисертаційного дослідження Василем Мраком полягає у наступному:

- аналіз ефективності роботи існуючих методів розпізнавання об'єктів;
- розробка методів обробки даних у системах розпізнавання рухомих об'єктів;
- реалізація та апробація розроблених методів.

Використання розроблених у дисертаційному дослідженні методів та засобів дало змогу побудувати та реалізувати систему розпізнавання рухомих об'єктів в реальному масштабі часу.

Члени комісії:

Начальник регіонального  
центру ядра мережі  
ЛФ АТ «УКРТЕЛЕКОМ»

  
\_\_\_\_\_

Роман МАЧИНСЬКИЙ

Начальник  
Відділу правового забезпечення  
ЛФ АТ «УКРТЕЛЕКОМ»

  
\_\_\_\_\_

Наталія ФІЛЯРОВСЬКА

ЗАТВЕРДЖУЮ



Директор ТзОВ „ МАКСІТЕХ”

Сергій ЗАБЛОЦЬКИЙ

« 16 » 04 2024 р.

**АКТ**

**про впровадження результатів  
дисертаційної роботи аспіранта кафедри «Телекомунікації»  
Національного університету «Львівська політехніка»  
Мрака Василя Богдановича**

Цей акт підтверджує, що результати дисертаційної роботи на здобуття наукового ступеня доктора філософії Мрака В. Б., були використані та впроваджені в практичну діяльність підприємства при розробці системи відеоспостереження для розпізнавання рухомих об'єктів в реальному масштабі часу, при цьому зменшуючи обсяги відео-трафіку та навантаження на мережу.

Автором розроблено методи обробки даних, що дозволяють ідентифікувати області зображення з рухом, використовуючи лише важливу для інформацію, таким чином зменшуючи навантаження на телекомунікаційну мережу.

Терміни проведення досліджень: листопад 2023 р. - лютий 2024 р.

Впровадження отриманих результатів дисертаційного дослідження Мраком В. Б. полягає у наступному:

- аналіз основних проблем та існуючих підходів до їх вирішення в даній галузі;
- розробка методів оптимального збору даних в системах відеоспостереження;
- впровадження в роботу існуючої системи відеоспостереження розроблених методів, аналіз їх ефективності та пропозиції щодо вдосконалення.

Використання запропонованих у дисертаційному дослідженні методів та засобів дало змогу реалізувати систему відеоспостереження для ефективного виявлення рухомих об'єктів.

Члени комісії:

д.т.н., доцент кафедри ТК

д.т.н., доцент кафедри ТК

Тарас МАКСИМЮК

Микола БЕШЛЕЙ



ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи

Національного університету

«Львівська політехніка»



Олег ДАВИДЧАК

«15» 04 2024 р.

**АКТ**

**про впровадження в навчальний процес результатів  
дисертаційної роботи на здобуття наукового ступеня доктора філософії  
Мрака Василя Богдановича**

Цей акт складений про те, що результати дисертаційної роботи Мрака В. Б. на тему «Методи та засоби розпізнавання рухомих об'єктів в телекомунікаційних системах в реальному масштабі часу», представленої на здобуття наукового ступеня доктора філософії, використовуються у навчальному процесі кафедри «Телекомунікації» Національного університету «Львівська політехніка».

Зокрема, у навчальному процесі використовуються такі методи, запропоновані Мраком В. Б.:

- Моделі аналізу об'єктів в телекомунікаційних системах в реальному масштабі часу (дисципліна «Програмні платформи інформаційно-комунікаційних систем» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за спеціальністю 126 «Інформаційні системи та технології», з галузі знань 12 «Інформаційні технології»);

- Методи та засоби розпізнавання рухомих об'єктів в телекомунікаційних системах (дисципліна «Інтернет речей та повсюдний комп'ютинг» для студентів освітньо-кваліфікаційного рівня «магістр», що навчаються за спеціальністю 172 «Електронні комунікації та радіотехніка», з галузі знань 17 «Електроніка, автоматизація та електронні комунікації»).

Завідувач кафедри ТК,  
д.т.н., професор

Михайло КЛИМАШ

Професор кафедри ТК,  
д.т.н., професор

Мар'ян КИРИК



**ЗАТВЕРДЖУЮ**  
Проректор з наукової роботи  
Національного університету  
"Львівська політехніка"  
Іван ДЕМІДОВ  
2024 р.

**А К Т**

використання наукових результатів дисертаційної роботи

**Мрака Василя Богдановича**

«Методи та засоби розпізнавання рухомих об'єктів в телекомунікаційних системах в реальному масштабі часу», представленої на здобуття наукового ступеня доктора філософії за спеціальністю 172 – телекомунікації та радіотехніка при виконанні держбюджетної НДР

Комісія в складі: голови комісії – начальника науково-дослідної частини, д.т.н., ст. досл. Небесного Р. В. та членів комісії – завідувача кафедри телекомунікацій, д.т.н., професора Климаша М.М., завідувача відділу науково-організаційного супроводу наукових досліджень, к.т.н. Лазько Г. В. і в. о. заступника начальника планово-фінансового відділу Фаст І. І., цим актом підтверджують, що результати дисертаційної роботи Мрака В.Б., використовувалися при виконанні держбюджетної НДР: «Розроблення інноваційних методів та засобів розгортання інтелектуальної інформаційної інфраструктури для подвійного використання в умовах цифрової трансформації України» (ДБ/Metaverse), 2023-2025 рр, №0123U100232, в якій Мраком В.Б. було обґрунтовано підходи до вдосконалення існуючих та створення нових методів і підходів до обробки даних з камер відеоспостереження, щоб ідентифікувати області зображення з рухом, відправляючи лише інформацію, важливу для моніторингу рухомих об'єктів. Мраком В.Б. розроблено алгоритм для стиснення відео, знятих камерами з обмеженим ресурсом, який використовує аналіз руху для видалення небажаних пікселів і кадрів зображення, що дозволяє досягнути середнього стиснення тестових даних на 30%, зберігаючи при цьому всю інформацію, важливу для аналізу поведінки рухомих об'єктів. Це значне зменшення розміру файлу значно розширює можливості моніторингу за рахунок мінімізації вимог до пропускної здатності та полегшення ефективної передачі даних із пристроїв з обмеженими ресурсами, в порівнянні з існуючими підходами.

**Голова комісії:**

Начальник науково-дослідної частини,  
д.т.н., ст. досл.

Роман НЕБЕСНИЙ

**Члени комісії:**

Зав.відділу науково-організаційного  
супроводу наукових досліджень

Галина ЛАЗЬКО

В.о. заступника начальника  
планово-фінансового відділу

Ірина ФАСТ

Завідувач кафедри  
телекомунікацій, д.т.н., проф.

Михайло КЛИМАШ





ЗАТВЕРДЖУЮ

Директор ТОВ „ҒІГАЦЕНТР”

Назарій КУРОЧКО

2024 р.

**АКТ**  
**про впровадження результатів**  
**дисертаційної роботи аспіранта кафедри «Телекомунікації»**  
**Національного університету «Львівська політехніка»**  
**Мрака Василя Богдановича**

Цей акт підтверджує, що результати дисертаційної роботи на здобуття наукового ступеня доктора філософії Мрака В. Б., були використані та впроваджені в практичну діяльність підприємства при аналізі даних в системі відеоспостереження та розпізнавання рухомих об'єктів в реальному масштабі час. Автором запропоновано методи для обробки даних з камер, що дозволяють ідентифікувати області зображення рухомих об'єктів, використовуючи тільки найважливішу інформацію.

Терміни проведення досліджень: січень 2024 р. - квітень 2024 р.

Впровадження отриманих результатів дисертаційного дослідження Василем Мраком полягає у наступному:

- аналіз ефективності роботи існуючих методів оброблення даних в системах відеонагляду;
- розробка нових методів ефективного розпізнавання рухомих об'єктів;
- реалізація та оцінка ефективності роботи запропонованих методів.

Використання розроблених у дисертаційному дослідженні методів дало змогу покращити ефективність виявлення рухомих об'єктів в системах відеонагляду.

Члени комісії:

Головний інженер

Керівник відділу техпідтримки ЦОД

Костянтин ПАНЧЕНКО

Віктор ЦИМБАЛ

**ДОДАТОК Б. Список публікацій здобувача за темою дисертації та  
відомості про апробацію результатів дисертації**

***Публікації, у яких опубліковані основні наукові результати дисертації:***

1. Климаш М. М., Гордійчук-бублівська О. В., Мрак В. Б., Браницький А. В. Дослідження ефективності обробки великих даних в системах відеонагляду // Інфокомунікаційні та комп'ютерні технології. 2022. № 1 (01). С. 52–62. (Належить до фахових видань України. технічні науки ISSN 2788-5518)

<https://doi.org/10.36994/2788-5518-2021-01-01-033>

*Особистий внесок здобувача: опрацювання результатів експериментальних досліджень, проведення аналізу даних в системах відеонагляду, обговорення одержаних результатів, підготовка статті до друку.*

2. Климаш М.М., Мрак В.Б., Гордійчук-Бублівська О.В., Дослідження методів виділення динамічних об'єктів у відеопослідовностях // Вісник «Інфокомунікаційні технології та електронна інженерія», №1, С. 63-75, 2021

<https://doi.org/10.23939/ictee2021.01.063>

*Особистий внесок здобувача: опрацювання результатів експериментальних досліджень виділення динамічних об'єктів у відеопослідовностях, обговорення одержаних результатів, підготовка статті до друку.*

3. Юнак О. М., Климаш М. М., Шпур О. М., Мрак В. Б. Математична модель розпізнавання фрактальних структур з використанням технології нейронних мереж // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2023. Vol. 3, № 1. Р. 1–9. (Належить до фахових видань України. технічні науки ISSN 2786-4553)

<https://doi.org/10.23939/ictee2023.01.001>

*Особистий внесок здобувача: участь у дослідженнях розпізнавання фрактальних структур з використанням технології нейронних мереж, опрацювання результатів досліджень, обговорення одержаних результатів, підготовка статті до друку.*

4. Мрак В. Б. Методи розпізнавання обличчя у системах відеоспостереження з використанням машинного навчання // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2023. Vol. 3, № 2. Р. 33–42. (Належить до фахових видань України. технічні науки ISSN 2786-4553)

<https://doi.org/10.23939/ictee2023.02.033>

*Особистий внесок здобувача: дослідження методів розпізнавання обличчя у системах відеоспостереження, інтерпретація результатів, написання статті.*

5. Мрак В. Б., Климаш М. М., Масюк А. Р., Колодій Т. Б. Методи та показники оцінки ефективності при розробці інтелектуальних систем відеоспостереження // Вчені записки Таврійського національного університету імені В. І. Вернадського. Серія: Технічні науки. 2024. Т. 35 (74), № 1. С. 252–259. (Належить до фахових видань України. технічні науки ISSN 2663-5941 E-ISSN 2663-595X)

<https://doi.org/10.32782/2663-5941/2024.1.1/38>

*Особистий внесок здобувача: опрацювання результатів оцінки ефективності при розробці інтелектуальних систем відеоспостереження, участь у інтерпретації результатів, написання статті.*

6. Мрак В. Б., Климаш М. М. Модель системи розпізнавання обличчя для нерухомих систем відеоспостереження // Вимірювальна та обчислювальна техніка в технологічних процесах. 2024. № 1. С. 68–73. (Належить до фахових видань України. технічні науки ISSN 2219-9365)

<https://doi.org/10.31891/2219-9365-2024-77-9>

*Особистий внесок здобувача: системи розпізнавання обличчя для нерухомих систем відеоспостереження, реалізація моделювання дослідження, написання статті.*

7. Мрак В. Б., Климаш М. М., Бабинець В. В. Удосконалення методу виявлення динамічних об'єктів у відео послідовностях // Вимірювальна та

обчислювальна техніка в технологічних процесах. 2024. № 2. С. 195–204.  
(Належить до фахових видань України. технічні науки ISSN 2219-9365)

<https://doi.org/10.31891/2219-9365-2024-78-22>

*Особистий внесок здобувача: дослідження удосконаленого методу виявлення динамічних об'єктів у відео послідовностях, обговорення одержаних результатів, підготовка статті до друку.*

8. Климаш М. М., Пиріг Ю. В., Сеник А. Д., Мрак В. Б. Дослідження контекстно-чутливого алгоритму моніторингу кібербезпеки на основі рекурентних нейронних мереж // Infocommunication Technologies and Electronic Engineering = Інфокомунікаційні технології та електронна інженерія. 2024. Vol. 4, № 1. Р. 1–9. (Належить до фахових видань України. технічні науки ISSN 2786-4553)

<https://doi.org/10.23939/ictee2024.01.001>

*Особистий внесок здобувача: аналіз результатів дослідження моніторингу кібербезпеки, інтерпретація результатів, написання статті.*

9. Zhuravel S., Klymash M., Shpur O., Mrak V. Reducing the impact of unstable connections among nodes of wireless IIoT clusters using machine learning methods // Lecture Notes in Electrical Engineering. 2024. Vol. 1198 : Digital ecosystems: interconnecting advanced networks with AI applications. Р. 144–159. (Наукове серійне видання Netherlands, ISSN: 1876-1100, E-ISSN: 1876-1119)

*Особистий внесок здобувача: дослідження даних в IIoT за допомогою методів машинного навчання, інтерпретація результатів, написання статті.*

### **Публікації, які засвідчують апробацію матеріалів дисертації:**

1. “Алгоритми та методи виявлення динамічних об'єктів інтелектуальних систем відеоспостереження”, Мрак В. Б., V Міжнародна науково-практична конференція “Наукоємні технології в інфокомуніаціях” НІСТ'2023 1-3 червня 2023

<http://repositsc.nuczu.edu.ua/bitstream/123456789/18013/1/%D0%9D%D0%86%D0%A1%D0%A2%202023.pdf>.



*Особистий внесок здобувача: опрацювання результатів досліджень методів виявлення динамічних об'єктів інтелектуальних систем відеоспостереження, аналіз та інтерпретація результатів, обговорення одержаних результатів, апробація матеріалів на конференції*

2. Intelligent IoT control system based on hand gesture recognition / D. Balazh, V. Mrak, A. Sydor, V. Andrushchak, B. Rusyn, T. Maksymyuk // Advanced information and communication technologies : proceedings of the 5th IEEE International conference (Lviv, Ukraine, November 21–25, 2023). 2023. P. 102–104.

<https://doi.org/10.1109/AICT61584.2023.10452708>

*Особистий внесок здобувача: опрацювання результатів досліджень аналізу даних в IoT, обговорення одержаних результатів, апробація матеріалів на конференції.*