МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ПОБЕРЕЙКО ПЕТРО БОГДАНОВИЧ

Кваліфікаційна наукова
праця на правах рукопису
УДК 004.652

Методи та засоби аналізу відеопотоків та пошуку подібностей у контентно-орієнтованих системах відеоінформації

122 – комп’ютерні науки

Дисертація на здобуття наукового ступеня
доктора філософії

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

_____/П. Б. Поберейко/

Науковий керівник

Мельникова Наталія Іванівна

доктор технічних наук, доцент

Львів – 2025

АНОТАЦІЯ

Поберейко П.Б. Методи та засоби аналізу відеопотоків та пошуку подібностей у контентно-орієнтованих системах відеоінформації. – Кваліфікаційна наукова праця на правах рукопису. Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки». – Національний університет «Львівська політехніка», Львів, 2025.

Зміст анотації. Дисертаційне дослідження присвячене розробленню методів і засобів аналізу відеопотоків та підвищенню точності пошуку подібних фрагментів у контентно-орієнтованих системах відеоінформації (CBVIR). Стрімке зростання обсягів відеоінформації у різних сферах – від систем відеоспостереження та мас-медіа до наукових досліджень і розважальних платформ – зумовлює гостру потребу в ефективних засобах аналізу, пошуку та організації відеоданих. Відповідно, актуальною науковою задачею є швидкий і точний пошук відеоматеріалів за змістом, особливо за умов обмежених обчислювальних ресурсів та великого обсягу даних. Запропоновані у роботі підходи інтегрують сучасні глибокі нейронні мережі (DCNN, LSTM, SlowFast) та методи аналізу сцен (SBD), що дає змогу підвищити точність і швидкодію систем контентного пошуку відео.

Основна частина роботи включає вступ, чотири розділи та висновки. У вступі розглянуто актуальність теми, визначено мету і основні завдання дослідження, окреслено наукову новизну та практичну цінність роботи. Для досягнення мети дисертації поставлено 6 завдань, які виконуються послідовно.

У першому розділі «Аналітичний огляд методів аналізу та пошуку відеоінформації» проведено аналіз літературних джерел і сучасних технологій у галузі контентно-орієнтованих систем відеоінформації. Розглянуто існуючі підходи до сегментації відеопотоків, виділення ключових кадрів та опису відеоконтенту, а також методи пошуку відео за змістом. Проаналізовано можливості застосування технологій глибинного навчання для задач CBVIR та виявлено їх обмеження при опрацюванні великомасштабних відеоданих у

реальному часі. За результатами огляду сформульовано завдання для подальших досліджень у межах дисертації.

У другому розділі «Розробка методів сегментації та виділення ознак відеопотоків» запропоновано методи поділу відеопотоків на смислові сцени і виокремлення інформативних ознак відео. Розроблено алгоритм визначення меж сцен на основі методу SBD, який дозволяє сегментувати відеопотік на окремі сцени для подальшого контентного аналізу. Запропоновано та реалізовано алгоритми сегментації і виділення ознак об'єктів на ключових кадрах, які адаптовані до роботи в умовах обмежених обчислювальних ресурсів. Отримані дескриптори відеовмісту структуруються та зберігаються у NoSQL базі даних для забезпечення швидкого доступу при пошуку. Додатково представлено методику клієнтського аналізу відеофрагментів, що включає завантаження довільного фрагмента відео, автоматизоване виявлення меж сцен, виділення багаторівневих ознак та формування комбінованих векторів характеристик, які описують вміст відео. Ці розробки закладають основу для ефективного пошуку відео за змістом у наступних розділах.

У третьому розділі «Методи пошуку подібних відеофрагментів із використанням глибинних мереж» описано розроблені методи порівняння та пошуку схожих відеофрагментів на основі аналізу їхнього змісту. Застосовано глибоку згорткову нейронну мережу (DCNN) у комбінації з рекурентною мережею (LSTM) для оцінювання схожості відеофрагментів за просторово-часовими характеристиками. Розроблено підхід, що враховує як часову динаміку, так і об'єктні ознаки відеоряду (Temporal Similarity Matching – TSIM, та Object Similarity Matching – OSIM). На основі цих методів формується первинний список кандидатів на схожість (Primary List, PLIST), який надалі оптимізується за допомогою додаткових критеріїв для отримання фінального списку подібних відео. Для підвищення точності й ефективності пошуку впроваджено допоміжну нейронну мережу прямого поширення (FNN), яка здійснює моніторинг роботи основної глибинної моделі та коригування її параметрів. Проведено експериментальні дослідження запропонованого

підходу, які підтвердили високий рівень точності пошуку релевантних відеофрагментів.

У четвертому розділі «Розробка архітектури системи та апробація результатів» представлено розроблений програмний комплекс для реалізації запропонованих методів у режимі реального часу. Спроєктовано загальну архітектуру інформаційної системи пошуку, що включає підсистеми збору, зберігання та обробки відеоданих. Створено сховище даних для збереження обчислених ознак відео та реалізовано програмні компоненти системи у вигляді клієнтської частини (веб-застосунок) і серверної частини (модуль глибинного аналізу). Розроблено діаграми, які описують основні сценарії використання системи та взаємодію між її компонентами, а також обґрунтовано вибір технологій (у тому числі хмарних сервісів) для розгортання системи. Для забезпечення адаптивної оптимізації в режимі реального часу в систему інтегровано модуль моніторингу на основі FNN-нейромережі, який аналізує ключові показники продуктивності (швидкість обробки, використання ресурсів, точність розпізнавання) та автоматично коригує параметри обробки відеопотоку, підтримуючи стабільну роботу пошуку. Реалізовано користувацький веб-інтерфейс, що дозволяє завантажувати відеофрагменти та виконувати пошук подібного відео у реальному часі. Наведено приклади роботи розробленої системи на тестових даних, які демонструють відповідність отриманих результатів вимогам високої швидкодії та точності. За результатами апробації підтверджено, що розроблена система повністю реалізує поставлену мету та завдання дослідження.

У висновках підсумовано результати, отримані в ході виконання дисертаційного дослідження, вказано, де вони можуть бути використані, та описано, яким чином цих результатів досягнуто.

Список літературних джерел наведений після висновків і включає 92 найменування, що охоплюють сучасні підходи до глибинного навчання, методів аналізу відеоінформації, CBVIR, а також прикладні аспекти реалізації систем відеопошуку в реальному часі.

У додатку А «Програмний код реалізації модуля сегментації та побудови векторів ознак відео» наведено фрагменти коду для реалізації алгоритмів SBD (Scene Boundary Detection), екстракції об’єктних та часових характеристик, а також формування комбінованих векторів ознак відеосцен.

Запропоновані методи і засоби впроваджено у практику, що підтверджено відповідними актами впровадження. Основні результати роботи опубліковано у 6 наукових публікаціях: 2 статті – у фахових виданнях України, 3 – у міжнародних виданнях, 1 – у матеріалах конференцій.

Ключові слова: відеоінформація, пошук за змістом, глибинне навчання, відеоаналіз, нейронні мережі, CBVIR, подібність відео, системи реального часу.

ABSTRACT

Pobereiko P.B. Methods and Tools for Video Stream Analysis and Similarity Search in Content-Based Video Information Systems. – A qualification research paper in manuscript form. A dissertation submitted for the degree of Doctor of Philosophy in the specialty 122 «Computer Sciences». – Lviv Polytechnic National University, Lviv, 2025.

*Abstract content***.** The dissertation research is devoted to the development of methods and tools for video stream analysis and for improving the accuracy of searching for similar fragments in content-based video information retrieval (CBVIR) systems. The rapid growth of video data in various fields — ranging from surveillance systems and mass media to scientific research and entertainment platforms — creates a pressing need for effective tools for analyzing, searching, and organizing video data. Consequently, the significant scientific challenge lies in achieving fast and accurate content-based video retrieval, especially under constraints of limited computational resources and large data volumes. The approaches proposed in this work integrate modern deep neural networks (DCNN, LSTM, SlowFast) with scene analysis methods (SBD), thus improving both the accuracy and speed of content-based video retrieval systems.

The main body of the work consists of an introduction, four chapters, and conclusions. The introduction discusses the relevance of the topic, defines the purpose and main objectives of the research, and outlines the scientific novelty and practical value of the work. To achieve the dissertation’s goal, six tasks are set and solved sequentially.

In the first chapter, "Analytical Review of Methods for Video Information Analysis and Retrieval", a review of literature and modern technologies in the field of content-based video information systems is conducted. Existing approaches to video stream segmentation, key-frame extraction, and video content description, as well as methods for content-based video retrieval, are considered. The possibility of applying deep learning technologies to CBVIR tasks is analyzed, and their limitations for large-scale real-time video data processing are identified. Based on the review, the objectives for further dissertation research are formulated.

In the second chapter, "Development of Video Stream Segmentation and Feature Extraction Methods", methods for dividing video streams into meaningful scenes and extracting informative video features are proposed. An algorithm for detecting scene boundaries based on the SBD method is developed, which allows for segmenting a video stream into individual scenes for subsequent content analysis. Algorithms for scene segmentation and object feature extraction from key frames are proposed and implemented, adapted to operate under limited computational resources. The resulting video content descriptors are structured and stored in a NoSQL database to ensure rapid access during retrieval. In addition, a client-side technique for video fragment analysis is presented, including the loading of arbitrary video fragments, automatic scene boundary detection, multi-level feature extraction, and the formation of combined feature vectors that describe the video content. These developments form the basis for effective content-based video retrieval in subsequent chapters.

In the third chapter, "Methods for Searching Similar Video Fragments Using Deep Networks", the developed methods for comparing and searching for similar video fragments based on their content analysis are described. A deep convolutional neural network (DCNN) is applied in combination with a recurrent network (LSTM) to assess

similarity of video fragments based on spatiotemporal characteristics. An approach that considers both temporal dynamics and object features of video sequences (Temporal Similarity Matching – TSIM, and Object Similarity Matching – OSIM) is developed. On the basis of these methods, an initial list of candidate similar fragments (Primary List, PLIST) is generated, which is further optimized using additional criteria to produce the final list of similar videos. To enhance search accuracy and efficiency, an auxiliary feedforward neural network (FNN) is implemented, which monitors the operation of the main deep model and adjusts its parameters. Experimental studies of the proposed approach confirm a high level of accuracy in retrieving relevant video fragments.

In the fourth chapter, "System Architecture Development and Validation of Results", the developed software suite for real-time implementation of the proposed methods is presented. The overall architecture of the video retrieval information system is designed, including subsystems for video data collection, storage, and processing. A data repository for storing computed video features has been created, and software components of the system have been implemented as a client-side web application and a server-side deep analysis module. Diagrams illustrating the main usage scenarios of the system and the interaction among its components are provided, as well as a justification for the choice of technologies (including cloud services) for system deployment. To ensure adaptive real-time optimization, the system integrates a monitoring module based on an FNN neural network, which analyzes key performance indicators (processing speed, resource usage, recognition accuracy) and automatically adjusts the parameters for video stream processing, maintaining stable operation of the retrieval process. A user-friendly web interface is implemented, enabling users to upload video fragments and search for similar videos in real time. Examples of the system’s performance on test data demonstrate compliance with high speed and accuracy requirements. The validation results confirm that the developed system fully achieves the research objectives and tasks.

The conclusions summarize the results obtained in the course of the dissertation research, indicate where they can be applied, and describe how these results have been achieved.

A list of references is provided after the conclusions and includes 92 sources covering modern approaches to deep learning, video information analysis methods, CBVIR, and applied aspects of real-time video retrieval systems.

Appendix A, "Software Code for the Implementation of Video Segmentation and Feature Vector Construction Module", provides code fragments for the implementation of SBD (Scene Boundary Detection) algorithms, as well as extraction of object and temporal characteristics, and the formation of combined feature vectors of video scenes.

The proposed methods and tools have been implemented in practice, as confirmed by corresponding implementation acts. The main results of the work have been published in 6 scientific publications: 2 articles in specialized Ukrainian journals, 3 in international journals, and 1 in conference proceedings.

Keywords: video information, content-based retrieval, deep learning, video analysis, neural networks, CBVIR, video similarity, real-time systems.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у фахових виданнях України:

1. Мельникова Н. І., Поберейко П. Б. Дослідження методів пошуку ключових кадрів у відеопотоці з використанням нейронних мереж для систем пошуку// Вісник Хмельницького національного університету. Серія: Технічні науки. – 2022. – № 3 (309). – С. 55–60.
2. Мельникова Н. І., Поберейко П. Б. Покращення можливостей пошуку відео: інтеграція нейронної мережі прямого поширення для ефективного фрагментного пошуку// Вісник Національного університету «Львівська політехніка». Серія: Комп'ютерні системи проектування. Теорія і практика. – 2024. – № 6 (1). – С. 149–160.

Статті у інших виданнях:

3. Pobereiko P., Melnykova N. Enhancing fragment-based video retrieval through the integration of feedforward neural networks // European Science. – 2024. – Issue sge27-03. – P. 126–146.
4. Shakhovska N., Melnykova N., Pobereiko P., Zakharchuk M. Investigating methods of searching for key frames in video flow with the use of neural networks for search systems // International Journal of Computing. – 2023. – Vol. 22, No. 4. – P. 455–461.
5. Melnykova N., Shakhovska N., Pobereiko P., Cichoń D. Advancing video search capabilities: integrating feedforward neural networks for efficient fragment-based retrieval // Data-Centric Business and Applications: Advancements in Information and Knowledge Management. Volume 1. – Springer Nature Switzerland, 2024. – P. 181–197.

Матеріали конференцій:

6. N. I. Melnykova and P. B. Pobereyko, "Інтегрований підхід до пошуку відео: використання глибоких згорткових нейронних мереж для точності ідентифікації фрагментів," The 12th International Scientific and Practical Conference “Modern Thoughts”, 2024.

Зміст

ВСТУП

Актуальність теми.

У сучасному світі стрімке зростання обсягів даних, які генеруються у різних сферах людської діяльності – від систем відеоспостереження та медіа до наукових досліджень і розважальних платформ, – зумовлює гостру потребу в ефективних засобах аналізу, пошуку та організації відеоінформації. Особливого значення це набуває в умовах обмежених обчислювальних ресурсів, що характерні для пристроїв і систем реального часу. Такі обмеження значно ускладнюють реалізацію високоточного аналізу динамічних станів об’єктів у відеопотоках.

Традиційні методи аналізу, зокрема базові алгоритми комп'ютерного зору, часто є недостатньо ефективними для точного пошуку та класифікації динамічних станів через їхню обмежену здатність адаптуватися до змінних умов контенту, таких як варіації освітлення чи зміна швидкості руху об'єктів. Інтеграція глибоких нейронних мереж, зокрема DCNN, LSTM та SlowFast Networks, відкриває нові перспективи для аналізу просторово-часових залежностей у відео. Однак навіть ці моделі потребують вдосконалення та розширення функціональних можливостей для досягнення високої точності та адаптивності в умовах динамічного контенту.

Ефективним підходом до вирішення цієї задачі є оптимізація роботи контентно-орієнтованих систем пошуку відео (CBVIR) шляхом інтеграції механізмів адаптивного аналізу тимчасових і просторових характеристик відео. Це передбачає розробку шарів для детекції та екстракції сцен, впровадження динамічних еталонів і використання методів сегментації для автоматичного визначення меж між сценами. Поєднання тимчасових характеристик із характеристиками об'єктів відео, такими як кольори, текстури, контури та рух, дозволяє значно підвищити точність ідентифікації та класифікації динамічних станів.

Запровадження таких підходів робить системи CBVIR більш адаптивними до змін контенту, включно з варіаціями освітлення та швидкості руху об'єктів,

забезпечуючи точний аналіз навіть в умовах обмежених обчислювальних ресурсів. Це сприяє підвищенню ефективності аналізу відеопотоків і розширює можливості систем для роботи з динамічним контентом у реальному часі.

Таким чином, розробка нових методів і моделей аналізу відеопотоків, які забезпечують високу точність та ефективність роботи в умовах обмежених ресурсів, є актуальною і важливою науковою та практичною задачею.

Зв'язок роботи з науковими програмами, планами і темами. Дисертація виконувалася відповідно до пріоритетних напрямків науково-дослідних робіт Національного університету “Львівська політехніка”, відповідно до координаційних планів Міністерства освіти і науки України. Зокрема, використано у науково-дослідній роботі, що виконувалась на кафедрі систем штучного інтелекту і включено до звіту за договором «Комплексний догляд для наступного покоління» (C2020/1-8, iCare4Next) виконаної за договором № M/100-2024 від 19.07.2024 р.

Мета дисертаційного дослідження. Метою дисертаційної роботи є підвищення точності та швидкодії пошуку відеофрагментів за їхнім змістом, а також покращення ефективності опрацювання великих обсягів відеоданих шляхом розроблення та впровадження нових моделей контент-орієнтованого аналізу відео на основі методів штучного інтелекту.

Для досягнення поставленої мети необхідно розв’язати такі задачі**:**

1. Провести аналіз сучасних методів і моделей для обробки та аналізу відеопотоків, зокрема методів сегментації, детекції сцен, ідентифікації та класифікації відеоданих.
2. Розробити методи виявлення меж сцен та їх інтеграції у системи CBVIR.
3. Розробити методи інтеграції просторових і часових характеристик відео, оптимізовані для забезпечення ефективної обробки даних.
4. Розробити алгоритми сегментації та виділення ознак об'єктів на зображеннях, адаптовані для роботи в умовах обмежених ресурсів.
5. Розробити модель аналізу геометричних ознак та гістограм градієнтів відео.

6. Створити архітектуру системи та програмні засоби для апробації запропонованих методів і моделей.

Об'єкт і предмет дослідження. *Об'єктом дослідження* є процеси аналізу та пошуку відеоконтенту у відеопотоках, а *предметом дослідження* – методи та моделі аналізу просторово-часових характеристик відеоданих для підвищення точності пошуку схожих відеофрагментів.

Методи дослідження. У роботі використано комплексний підхід, який охоплює глибинні нейронні мережі (CNN, LSTM, SlowFast) для вилучення високорівневих ознак і просторово-часового аналізу відеофрагментів; класичні методи комп'ютерного бачення (аналіз градієнтів, сегментація, виділення ключових точок і гістограм) для виявлення меж сцен, формування дескрипторів та об'єктних ознак. Використано дискретні диференціальні оператори для контурного аналізу, теорію геометричних перетворень для виділення ознак та аналізу динаміки об'єктів.

Наукова новизна одержаних результатів:

- *Удосконалено* метод виявлення меж сцен у відео шляхом застосування аналізу градієнтів і тимчасових характеристик фрагментів відеокадрів, упровадження якого в контентно-орієнтовані системи (CBVIR) значно підвищує точність визначення меж сцен, особливо в умовах динамічного контенту.
- *Удосконалено* метод індексації відеоконтенту, що поєднує алгоритми сегментації сцен з глибокими нейронними мережами (CNN, Transformer) та методами розрідженого представлення даних, що дає змогу суттєво зменшити обчислювальні витрати під час обробки відеопотоків.
- *Розроблено* модель для класифікації та кластеризації об'єктів відео на основі згорткової нейронної мережі (із залученням геометричних ознак і гістограм градієнтів), а також створено архітектуру модульної системи індексації відео, яка адаптується до змін динаміки сцени та підтримує багаторівневу обробку відеопотоку в режимі реального часу.

Запропонований підхід підвищує швидкість і точність обробки запитів у CBVIR-системах.

- *Розроблено* метод аналізу продуктивності та адаптивної корекції DCNN+LSTM-моделі, що охоплює аналіз ключових показників (точність класифікації, швидкість обробки кадрів, витрати пам’яті та обчислювальних ресурсів), формування прогнозу ефективності, а також автоматичну генерацію й передачу сигналів корекції. Запропонований підхід дає змогу динамічно змінювати ваги та гіперпараметри моделі під час зміни формату відео чи зниження якості кадрів, забезпечуючи стабільну роботу системи в реальних умовах.

Практичне значення одержаних результатів. Розроблені методи впроваджено у навчальний процес Національного університету «Львівська політехніка» у вигляді представленого електронного курсу з дисципліни «Організація баз даних та знань». На основі запропонованих методів створено архітектуру системи для аналізу відеопотоків.

Також результати дисертаційної роботи впроваджено в ПНВП «РЕЗОН» (підтверджено актом впровадження).

Використано у науково-дослідній роботі і включено до звіту за договором «Комплексний догляд для наступного покоління» (C2020/1-8, iCare4Next № M/100-2024 від 19.07.2024 р.).

Особистий внесок здобувача. Основні положення та результати дисертаційної роботи одержані автором самостійно. Особисто здобувачеві належать наступні наукові результати: розроблено обчислювальну модель детекції сцен у відеопотоках для контентно-орієнтованих систем пошуку, що поєднує аналіз гістограм у просторі HSV, оцінку абсолютних різниць пікселів та механізм уваги, запропоновано метод інтеграції просторових і часових ознак (включно з оптичним потоком та інформацією про переходи між сценами) на базі архітектури SlowFast, який істотно покращує точність аналізу швидкозмінних динамічних сцен, розроблено алгоритм формування векторів ознак відео з урахуванням виділених об’єктних характеристик (MobileNet/EfficientNet) та

часових залежностей (LSTM), що дає змогу виявляти складні патерни під час пошуку подібностей, створено систему пошуку подібностей та релевантність пошуку фрагментів відео за довільним контентним запитом.

Апробація результатів дисертації. У результаті дисертації розроблена і запроваджена інформаційна технологія пошуку відео у контенто-орієнтованому середовищі. Роботу апробовано на Міжнародній науково-практичній конференції «Modern Thoughts» (2024).

Публікації. За результатами виконаних досліджень опубліковано 6 наукових праць, із них 2 статті – у фахових виданнях України, 3 публікації – у міжнародних виданнях, 1 – у збірниках наукових праць конференцій.

Структура та обсяг дисертації. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 92 найменування та додатків. Повний обсяг дисертації становить 172 сторінки, основний зміст викладено на 140 сторінках, де наведено 22 рисунків та 5 таблиці.

РОЗДІЛ 1. АНАЛІЗ АЛГОРИТМІВ ПОШУКУ У ВІДЕОПОТОКАХ ТА ЇХ КЛАСИФІКАЦІЇ

Однією із найбільш важливих задач технології пошуку відео у відеопотоках є задача ефективного пошуку потрібної інформації серед великої кількості відеоданих. Необхідність у її вирішенні зумовлена тим, що сучасний світ генерує величезні обсяги відеоданих, і ефективне управління, пошук та аналіз цих даних є критично важливими для забезпечення безпеки, підвищення ефективності в різних галузях, підтримки інтелектуальних систем, наукових досліджень та покращення користувацького досвіду.

На сьогодні досягнуто значних успіхів у даній області, зокрема, у розвитку алгоритмів виділення ознак, впровадженні глибокого навчання та штучного інтелекту для покращення точності пошуку, оптимізації апаратного забезпечення для обробки великих обсягів відеоданих у реальному часі, використанні хмарних технологій для масштабування обчислювальних потужностей, а також створенні ефективних і зручних інтерфейсів користувача для пошуку і взаємодії з відеоданими. Однак, хоча ці методи сьогодні досягли значних успіхів, та попри те вони все ще мають суттєві обмеження у сфері аналізу контенту. Зокрема, ці обмеження включають високу потребу в обчислювальних ресурсах, складнощі з масштабованістю для обробки великих обсягів даних, проблеми з точністю та надійністю через можливі помилкові спрацьовування або пропущені релевантні об'єкти, обмеженість доступних навчальних даних для глибокого навчання, варіативність умов зйомки, питання конфіденційності та безпеки даних, а також складнощі з урахуванням контекстуальних і часових залежностей.

Одним із шляхів вирішення цієї задачі є використання гібридного підходу, який поєднує традиційні методи виділення ознак з передовими техніками глибокого навчання. Його застосування може значно покращити ефективність, точність і надійність пошуку відео у відеопотоках, вирішуючи низку існуючих обмежень.

Тому з метою удосконалення сучасних технологій пошуку відео у відеопотоках у даному розділі викладено аналіз відомих на сьогодні поточних методів виділення, зберігання, структурування, аналізу та роботи з відеоданими та висвітлено їх переваги, області застосування та обмеження. Для оцінки схожості між кадрами розглянуто метрики подібності, а також моделі та методи машинного навчання, які забезпечують ефективний аналіз відеоконтенту.

На основі викладеного аналізу сформульовано завдання дослідження [41, 44].

1.1. Методи аналізу часових характеристик відео та їх застосування у CBVIR

Для удосконалення існуючих та синтезу нових технологій пошуку відео у відеопотоках важливими є дослідження [1,2,3], присвячені методикам використання тимчасових характеристик у системах контентно-орієнтованого відео пошуку (CBVIR). Їх практична цінність зумовлена тим, що тривалість кадрів, частота кадрів та послідовність сцен можуть надати важливу інформацію про зміст відео. Крім цього аналіз цих характеристик дозволяє системам CBVIR краще розуміти динаміку та структуру відео, що сприяє точнішому пошуку відповідних файлів. Використання алгоритмів машинного навчання та глибинного навчання для обробки та аналізу тимчасових характеристик відкриває нові можливості для систем CBVIR [4]. Проте, з огляду на [5], обробка тимчасових характеристик вимагає складних алгоритмічних підходів та високої обчислювальної потужності. Тому вдосконалення існуючих пошукових систем сьогодні відбувається або шляхом удосконалення сучасних методів сегментації відео на логічні блоки та події, або своєрідним комбінуванням тимчасових характеристик між собою, або їх поєднанням з іншими типами даних. Тому для вирішення поставленої у дисертаційній роботі задачі важливою є публікація [6], у якій наведено класифікацію тимчасових характеристик відео за такими категоріями:

- Тривалість сцен: дозволяє оцінювати і квантифікувати часові відрізки кожного індивідуального кадру або сцени у відео.

- Часові відмітки: включають точний час початку та закінчення фрагмента у контексті повного відео.
- Порядок подій: фіксує послідовність подій або сцен у фрагменті, що допомагає врахувати контекстуальну послідовність при пошуку.
- Характеристики переходів: включає аналіз типів переходів між сценами (наприклад, затемнення, розмиття), які можуть бути особливо характерними для певних відео.

Наукова та практична цінність цієї класифікації полягає у тому, що вона може бути використана для удосконалення вищезазначених систем пошуку шляхом комбінування тимчасових характеристик між собою та іншими типами даних. Цей підхід є ключовим у методиці аналізу відео SBD (Shot Boundary Detection). Комбінуючи різні характеристики відео, такі як інтенсивність пікселів, гістограми кольорів, оптичний потік, ключові точки і дескриптори, статистичні характеристики тощо, SBD дозволяє досягти високої точності та надійності у виявленні меж сцен. Але це лише один з аспектів її реалізації. Основна мета SBD – автоматично і точно визначати межі між різними сценами для полегшення аналізу та обробки відеоконтенту. Цей процес забезпечує автоматичне виявлення точок, де завершується один кадр (або сцена) та починається інший. SBD є надзвичайно важливим для різних застосувань, зокрема для вирішення задач індексації, редагування, стиснення та пошуку відео. Тому для вирішення поставленої у дисертаційній роботі задачі практично цінними є дослідження та розробки присвячені методам SBD та їх класифікації. Особливо вагомими є роботи [7, 8] у яких викладено класифікацію цих методів залежно від типу переходу між кадрами, згідно з якими SBD може класифікуватися на два основні типи:

- Жорсткі переходи (Hard Cuts): це раптова зміна від одного кадру до іншого без будь-яких перехідних ефектів. Жорсткі переходи можуть бути виявлені за допомогою аналізу змін у характеристиках кольору, текстури або інтенсивності між послідовними кадрами.

- М'які переходи (Soft Cuts або Gradual Transitions): це поступові переходи, такі як затухання, розмиття, або інші ефекти, які тривають протягом декількох кадрів. Для їх виявлення потрібні більш складні техніки, які аналізують зміни в кадрах на протязі більшої кількості часу, включаючи зміни в розподілі гістограм, аналіз руху чи змін у контурах об'єктів.

Ефективними засобами виявлення різких переходів (Hard Cuts) між послідовними кадрами відео є методи аналізу гістограм, а саме метод різниці гістограм та метод абсолютних різниць кадрів. Їх поєднання з іншими методами, зокрема з методами аналізу оптичного потоку або виявлення ключових точок дає змогу істотно підвищити точність обробки відео.

Метод різниці гістограм полягає в аналізі різниці між гістограмами суміжних кадрів [9]. Гістограма – це статистичний опис розподілу тонів (яскравості) або кольорів у кадрі. Різниця між гістограмами може бути обчислена за допомогою кількох метрик, найпопулярнішою з яких є метрика Хі-квадрат:

$$\chi^2 = \sum_{i=1}^{n} \frac{\left(H_1(i) - H_2(i)\right)^2}{H_1(i) + H_2(i)}, \quad (1.1)$$

де: H_1 та H_2 – гістограми двох послідовних кадрів;

n – кількість “корзин” у гістограмі.

Метод абсолютних різниць кадрів базується на обчисленні суми абсолютних різниць інтенсивностей пікселів між двома послідовними кадрами:

$$D = \sum_{x=1}^{X} \sum_{y=1}^{Y} |I_1(x, y) - I_2(x, y)|, \quad (1.2)$$

де I_1 та I_2 – інтенсивності пікселів у двох послідовних кадрах;

X та Y – розміри кадрів.

Якщо виявиться, що значення величини D перевищує заздалегідь визначений поріг, то це свідчитиме про різкий перехід між кадрами.

Порівняно з методом різниці гістограм, метод абсолютних різниць кадрів є більш чутливим до шуму, проте він забезпечує більшу точність виявлення змін між кадрами [10]. Метод різниці гістограм використовується у задачах, де важливим є врахування розподілу кольорів у сценах та зменшення чутливості до незначних змін у кадрах.

Більш точними та стійкими до шуму є ентропійні методи аналізу відеопотоків. Вони є більш ефективними у випадках виявлення та оцінки зміни у послідовностях кадрів відео, а також аномалій та важливих подій у відеопотоках. Сьогодні вони є потужним інструментом для вирішення складних задач аналізу, особливо коли необхідно забезпечити високу точність відео за умови стовідсоткового збереження його інформаційного вмісту. Тоді як методи аналізу гістограм використовуються в основному для вирішення задач у випадках коли важливими є простота та швидкість їх реалізації.

Метрикою ентропійних методів аналізу відеопотоків є ентропія:

$$H = -\sum_{i=1}^{n} p_i log(p_i), \quad (1.3)$$

де p_i – ймовірність кожного значення інтенсивності в гістограмі кадру.

Ця величина є кількісною мірою для визначення рівня невизначеності розподілу даних у відео. Вона дає змогу оцінити інформаційний вміст та складність кадрів у відео. Висока ентропія в окремих областях кадру є ознакою аномальних або непередбачуваних подій [11], що є важливим для безпеки змін, аномалій та важливих подій у відеопотоці.

Важливим методом для вирішення задач обробки відео є метод детектування кадрів (Shot Boundary Detection, SBD). Його практична цінність полягає у тому, що він дає змогу автоматично визначити точки у яких відбуваються переходи між різними сценами відео [8, 12] та розбити його на окремі логічні частини, що значно спрощує подальший аналіз, інтерпритацію та обробку відеоданих. Однак, зазначимо, що цей метод має і декілька недоліків. Він є чутливим до шуму у відео, що може істотно вплинути на точність алгоритмів детектування меж кадрів. Наприклад, низька роздільна здатність або стиснення відео можуть спричинити втрату важливих деталей кадру, що ускладнює виявлення меж кадрів. М'які переходи, такі як затухання або розмиття, можуть бути особливо важкими для виявлення, оскільки вони відбуваються поступово та можуть бути не так виразно розрізнені алгоритмами, які шукають раптові зміни. А також редакційні та візуальні ефекти, такі як графічні вставки, субтитри або спецефекти, можуть впливати на ефективність

SBD, оскільки вони можуть вводити додаткові візуальні зміни, які не відображають реальні межі кадрів. Проте, зазначимо, що ці недоліки можна усунути за допомогою різних методів попередньої обробки зображень, адаптивних порогових значень та використання сучасних алгоритмів глибокого навчання, і саме тому сьогодні він є основою для вдосконалення відомих та розроблення нових методів та алгоритмів вирішення задач пошуку відео у відеопотоках.

Окрім завдань детектування меж кадрів не менш важливим завданням задачі ефективного аналізу та ідентифікації відеоконтенту є аналіз просторово-часової інформації у відео [13]. Цей аналіз включає обробку даних, що охоплюють не лише зміни, які відбуваються у фізичних місцерозташуваннях об'єктів на сцені, але і їх динаміку в часі. Він дає змогу виявляти та реконструювати траєкторії руху об'єктів, а також аналізувати їх взаємодії та поведінку. Важливість цього аналізу полягає в можливості відновлення подій, що відбуваються у відео, з високим рівнем деталізації, що сприяє покращенню точності та релевантності пошукових результатів.

Науково-цінними у цій галузі знань є дослідження присвячені класифікації існуючих методів та алгоритмів просторово-часового аналізу інформації у відео. Вони мають важливе значення для систематизації знань, проведення порівняльного аналізу, виявлення прогалин та перспективних напрямків, підвищення якості досліджень та розробок в області технологій пошуку відео у відеопотоках. З точки зору практичного застосування, алгоритми та методи просторово-часового застосування прийнято поділяти на такі категорії:

- Траєкторний аналіз: зосереджений на відстежуванні шляхів руху об'єктів у відео. Він включає розрахунок векторів швидкості, прискорення, і напрямків руху, що може бути використано для аналізу поведінки об'єктів та їх взаємодії з оточенням [14].
- Аналіз подій: фокусується на виявленні та класифікації подій, які відбуваються в часі та просторі, наприклад, падіння людини, автомобільна

аварія, або прості соціальні взаємодії. Використання машинного навчання та алгоритмів розпізнавання образів є ключовими у цій категорії [15].

- Аналіз змін: обробка змін у просторі сцени, таких як зміна освітлення, переміщення предметів, або зміна конфігурації сцени. Цей аналіз важливий для виявлення редагувань відео або несподіваних подій [16].

Потужним засобом для обробки та аналізу відеоданих є фільтр Калмана. Особливо ефективним є його використання в задачах відстеження рухомих об’єктів та оцінки їх траєкторій у реальному часі. Цей алгоритм є рекурсивним алгоритмом. В основі його роботи покладено систему рівнянь передбачення та оновлення [17]:

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$$

$$K_k = P_{k|k-1} H_k^T \left(H_k P_{k|k-1} H_k^T + R_k\right)^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \left(z_k - H_k \hat{x}_{k|k-1}\right)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

де $\hat{x}_{k|k-1}$ та $\hat{x}_{k|k}$ – апріорна та апостеріорна оцінки стану; P – коваріаційна матриця помилки оцінки; K_k – коефіцієнт зсуву Калмана; A_k, B_k, H_k – матриці динаміки системи, управління та вимірювання відповідно; Q_k та R_k – коваріаційні матриці процесу та вимірювання шуму.

Рівняння системи (1.2) істотно мінімізують похибки обробки вхідних даних. Завдяки цьому, фільтр Калмана може ефективно зменшувати вплив шуму та перешкод у вхідних даних, забезпечуючи високу точність і надійність оцінок стану системи. Однак, не зважаючи на це, його застосування є обмеженим, оскільки він базується на припущеннях про лінійність вхідних даних та нормальний розподіл шуму у цих даних, а також необхідністю точного знання про параметри системи. Для нелінійних систем або у випадках, коли припущення фільтра Калмана не виконуються, автори робіт [17, 18] рекомендують використовувати розширений фільтр Калмана (EKF), фільтр частинок або інші адаптивні методи.

Для розуміння та аналізу динамічних змін (часових характеристик) у відео особливо цінними та актуальними є методи оптичного потоку. Вони ефективно застосовуються для визначення шаблону видимого руху об’єктів, тіней та засвічень між двома послідовними кадрами відео. Серед цих методів одним з найпопулярніших сьогодні є алгоритм Лукаса-Канаде. Математичною основою цього алгориму є рівняння зв’язку вектора швидкості пікселів (вектора оптичного потоку) з градієнтами яскравості (інтенсивності) пікселів [19]:

$$\sum_{x,y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \sum_{x,y} \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix}, \quad (1.5)$$

де I_x, I_y – просторові градієнти зображення;

I_t – часовий градієнт;

v_x, v_y – компоненти вектора швидкості оптичного потоку.

Для розв'язання системи лінійних рівнянь (1.5), яка описує зв'язок між градієнтами інтенсивності та вектором оптичного потоку, алгоритм використовує метод найменших квадратів. Це дозволяє мінімізувати похибки та отримати оптимальні оцінки векторів руху.

Не менш цінними для удосконалення існуючих та розробки нових технологій пошуку відео у відеопотоках є дослідження [20, 21], які присвячені методикам обробки тимчасових характеристик відео та вивченню можливостей їх використання у системах контентно-орієнтованого відеопошуку (CBVIR). Практична цінність цих робіт полягає в тому, що впровадження зазначених методик у CBVIR є одним із ефективних способів підвищення точності обробки відеоданих: точніше визначаються межі між сценами, покращується аналіз тривалості, частоти кадрів та їх послідовності. Окрім цього, інтегровані у системи CBVIR алгоритми машинного та глибокого навчання для обробки та аналізу тимчасових характеристик відео відкривають нові можливості для CBVIR. Зокрема, за допомогою алгоритмів глибокого навчання CBVIR зможуть виявляти та аналізувати складні закономірності у відео. Машинне навчання дозволить системі CBVIR адаптуватися до різних типів контенту, а глибокі нейронні мережі зможуть автоматично навчатися на нових відеоданих.

Впроваджені у CBVIR алгоритми ефективніше розпізнаватимуть і класифікуватимуть об'єкти, події та дії у відео. Глибоке навчання також об'єднуватиме аналіз відео з іншими типами даних (текстовими метаданими або аудіоданими), що дозволить CBVIR системам здійснювати пошук за складними запитами. У підсумку, інтеграція цих методик є одним із найефективніших способів забезпечення більш точного і контекстуально релевантного пошуку відео за його окремими фрагментами в системі CBVIR.

За даними досліджень [22, 23] CBVIR з інтегрованими часовими характеристиками відео є більш ефективними системами відеоаналізу порівняно з традиційними CBVIR. Особливо цінною є система CBVIR із впровадженим механізмом виявлення та ідентифікації дубльованих копій відео, розробленим авторами роботи [22] на основі аналізу темпоральної характеристики відео – тривалості кадрів. На відміну від традиційних систем CBVIR, система CBVIR із впровадженим механізмом виявлення та ідентифікації дубльованих копій відео забезпечує вищу точність у визначенні подібних відеофрагментів. Вона здатна порівнювати візуальні характеристики відео та аналізувати їхні темпоральні характеристики, а також виявляти незначні варіації у дубльованих копіях. Завдяки цим особливостям, ця система є більш ефективною у виявленні дублікатів, що зменшує обсяг непотрібних даних і підвищує загальну продуктивність та точність відеоаналізу.

Важливими для удосконалення існуючих систем CBVIR є результати дослідження [23], автори якого, інтегрувавши метод ручного витягання ознак LBP-TOP у згорткову нейронну мережу (CNN), синтезували систему для покращеного аналізу відео. Справді, згідно з порівняльним аналізом CBVIR із впровадженою системою та традиційних CBVIR, CBVIR із впровадженою системою [23], на відміну від традиційних підходів, демонструє значно вищу точність у розпізнаванні складних динамічних патернів. Це досягається завдяки комбінуванню потужності глибокого навчання з ефективним використанням просторово-часових ознак, що дозволяє системі краще розпізнавати деталі та динаміку у відео. Такий підхід значно покращує результати відеоаналізу,

зменшуючи кількість помилкових спрацьовувань та підвищуючи загальну продуктивність системи.

Науково та практично цінною для удосконалення традиційних CBVIR є робота [24]. Її автори вперше, використовуючи дескриптор SURF та концепцію щільного оптичного потоку між кадрами, запропонували модель глибокого навчання для розпізнавання дій у щільних траєкторіях об’єктів відео з урахуванням руху камери. Практична цінність цієї моделі визначається її здатністю значно підвищувати точність розпізнавання дій у складних динамічних середовищах, де рух камери може створювати додаткові виклики для традиційних методів аналізу відео. Вона легко адаптується до різних типів рухів і сцен завдяки здатності ефективно об'єднувати просторово-часові характеристики кадрів відео. Її використання у традиційних CBVIR є ефективним способом створення нових систем для ідентифікації та аналізу відео за окремими кадрами, зокрема систем аналізу спортивних відео або систем автоматичного позначення відеоархівів, де використання традиційних CBVIR не є достатньо ефективними або доречним.

Значним внеском у розвиток технологій пошуку відео у відеопотоках є дослідження авторів публікації [25], де викладено оригінальну методику використання різних технік для аналізу часових аспектів відео цифрової субтракційної ангіографії (ДСА). Основою цієї методики є нейронні мережі глибокого навчання, які оптимально адаптовані до вирішення задачі витягання ознак із часових аспектів відео ДСА. Її впровадження у традиційні системи CBVIR дає змогу вирішувати не лише специфічні задачі діагностики артеріовенозних мальформацій головного мозку, але й інші завдання. Побудовані на її основі системи CBVIR застосовуються у галузях, де важливим є детальний аналіз руху та змін у відео.

Загалом, аналіз часових характеристик відео, зокрема тривалості сцен, часових відміток, порядку подій та характеристик переходів між сценами, є ключовим елементом у покращенні традиційних та не традиційних систем CBVIR. Використання цих характеристик у комбінації з колірними, текстурними

та геометричними ознаками, а також іншими типами даних є ефективним способом досягнення високої точності та надійності у виявленні меж між сценами та розпізнаванні об'єктів відео. Методи аналізу часових характеристик є основними засобами для вдосконалення існуючих алгоритмів пошуку відео. Їх використовують для забезпечення необхідної точності ідентифікації змін у відеопослідовностях, виявлення переходів між сценами, а також для розпізнавання послідовності подій у відео. Їх впровадження у традиційні системи CBVIR підвищує ефективність та точність пошукових запитів. Окрім цього, системи CBVIR, у які інтегровані ці методи, демонструють значно кращі результати в аналізі складних відеоматеріалів, забезпечуючи більш детальне та контекстуально релевантне розуміння змісту відео.

1.2. Методи аналізу об'єктних характеристик відео та їх застосування у CBVIR

Для повноти аналізу та розуміння відеоінформації важливими є не лише тимчасові характеристики відео, не менш важливими є також і характеристики об’єктів відео. Відповідно до матеріалу, викладеного у попередньому параграфі розділу даної роботи, ці характеристики відрізняються за визначенням та змістом. Тимчасові характеристики відео використовують для опису поведінки об’єктів у відеопослідовності, а характеристики об’єктів відео – для опису візуальних властивостей (статичних ознак) об’єктів у кадрі.

Аналіз сучасних систем контентно-орієнтованого пошуку відеоінформації (CBVIR) показав, що розпізнавання, класифікація та ефективний пошук зображень у відео реалізується за допомогою характеристик об’єктів відео. В системах CBVIR ці характеристики є одним із основних інструментів для виявлення схожості об’єктів відео, а також відновлення та індексування великих обсягів даних. Крім того, з їхньою допомогою можна істотно автоматизувати процеси обробки відеопотоків, які зазвичай вимагають значних програмних та апаратних ресурсів. Тому для удосконалення існуючих та розроблення нових технологій та систем CBVIR надзвичайно актуальними та важливими є

дослідження [26, 27, 28], присвячені питанням класифікації характеристик об’єктів відео. Наукова та практична цінність цих досліджень полягає у тому, що алгоритми з покращеною класифікацією характеристик об'єктів значно легше адаптуються до різноманітних сценаріїв та умов, що підвищує їхню універсальність і стійкість [26]. Детальна і досконала класифікація, покладена в основу алгоритмів відстежування руху об’єктів у відеопотоці підвищує ефективність систем CBVIR, навіть при складних умовах, таких як часткове перекриття або зміни освітлення. Сьогодні найкращим підходом до класифікації характеристик об'єктів відео є їх розподіл за такими категоріями [27]: колірні характеристики (колірні ознаки), геометричні характеристики (формові атрибути та ознаки), текстурні характеристики (текстурні маркери), характеристики руху та апаратні характеристики. Цей підхід вважається найкращим хоча б тому, що він зреалізований у сучасних системах CBVIR, які успішно вирішують завдання аналізу відеоданих. Більше цього дана класифікація значно спрощує постановку задачі класифікації існуючих методів аналізу відеоданих, оскільки вона найбільше відповідає основній концепції обробки відеоданих [27]: різним категоріям характеристик об'єктів відео відповідають відповідні (спеціалізовані) методи їх виявлення та аналізу. Наприклад, для виявлення та аналізу колірних ознак об’єктів у відеопотоці використовуються методи аналізу кольорів, а для виявлення та аналізу текстурних маркерів – методи текстурних маркерів. Ці та інші, відомі на сьогодні, методи аналізу характеристик об’єктів відео є основою сучасних систем контентно-орієнтованого пошуку відеоінформації. Тому для досягнення поставленої у дисертаційній роботі мети проаналізуємо їх більш детальніше.

Основними методами аналізу колірних ознак у відеопотоці є методи гістограм та кореляції колірних характеристик об’єктів у відео.

Метод гістограм призначений для визначення та аналізу візуалізованого розподілу кольорів у зображенні або певній області зображення для розрахунку кількості пікселів для кожного кольору (або діапазону кольорів). Візуалізація розподілу кольорів здійснюється за допомогою гістограми кольорів – графічного

відображення залежності частоти появи кольору в зображенні від самого кольору [28]:

$$H(c) = \frac{n_c}{N}, \tag{1.6}$$

де n_c – кількість пікселів з кольором c;

N – загальна кількість пікселів в зображенні.

Для аналізу колірних ознак використовуються гістограми, які побудовані у різних колірних просторах, таких як RGB, HSV, Lab тощо. RGB застосовується в основному для візуалізації та редагування зображень, HSV – для сегментації та класифікації кольорів, а Lab – для точного відтворення кольорів, калібрування кольорів і колориметрії, а також у додатках, де важливе відповідність кольорів людському сприйняттю. Вибір колірного простору є залежним від конкретних вимог задачі та компромісу між простотою реалізації, обчислювальними ресурсами та точністю аналізу кольорів [27].

Ще одним не менш популярним методом аналізу кольору є метод кореляції. Він дає змогу визначити схожість двох зображень або відеокадрів. Критерієм схожості зображень у цьому методі є коефіцієнт кореляції Пірсона [29]:

$$\rho(H_1, H_2) = \frac{\sum_{i=1}^{n}(H_1(i)-\overline{H_1})(H_2(i)-\overline{H_2})}{\sqrt{\sum_{i=1}^{n}(H_1(i)-\overline{H_1})^2 \sum_{i=1}^{n}(H_2(i)-\overline{H_2})^2}}, \tag{1.7}$$

де $\underline{H_1}$ і $\underline{H_2}$ – середні значення гістограм H_1 і H_2 відповідно.

Коефіцієнт кореляції Пірсона є мірою лінійного зв'язку між двома наборами даних. У контексті зображень або відео, високий коефіцієнт кореляції означає, що чим більшим є його значення, тим більша ймовірність того, що інтенсивності пікселів у двох зображеннях або відео змінюються подібним чином. Однак це не завжди означає, що зображення або відео є візуально схожими.

Порівняно з методом гістограм, метод кореляції може бути істотно точнішим у врахуванні відмінностей у розподілі кольорів зображень або відео. Завдяки цьому його часто використовують для розпізнавання дрібних деталей.

Проте метод кореляції є ресурсоємним і чутливим до невеликих змін в зображеннях, що може впливати на його стабільність.

Надзвичайно важливими для аналізу та обробки відеопотоків є текстурні методи аналізу текстурних маркерів [30, 31, 32]. Їх практична цінність полягає в тому, що за їх допомогою можна аналізувати та описувати такі візуальні характеристики зображень, які не можна повністю охопити іншими методами, такими як методи аналізу кольору. В їх основі покладено алгоритми визначення та порівняння текстурних властивостей об'єктів відео, таких як регулярність, напрямок і гранулометрія поверхні, а не колірних характеристик. До цих методів належать Gabor-фільтри та Local Binary Patterns (LBP).

Габор-фільтри є лінійними фільтрами, і одним із їх основних призначень є визначення добутку гармонійної функції (часто синусоїди або косинусоїди) та гауссової функції [30]:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right) cos\left(2\pi \frac{x'}{dx} + \psi\right), \qquad (1.8)$$

де

$$x' = xcos\theta + ysin\theta$$

$$y' = -xsin\theta + ycos\theta$$

Параметри фільтра:

λ – довжина хвилі синусоїди (просторова частота);

θ – орієнтація фільтра;

ψ – фазове зміщення синусоїди;

σ – стандартне відхилення Гаусової оболонки;

γ – співвідношення сторін гаусової функції (еліптичність).

Габор-фільтр G(λ,θ,ψ,σ,γ) є мірою подібності текстурних властивостей зображень у відео. Наприклад, якщо у G(λ,θ,ψ,σ,γ) зафіксувати параметри θ, ψ, σ, γ, то отримаємо фільтр, який буде чутливий лише до текстурних ознак зображення на конкретній частоті 2π/λ. Зміна 2π/λ в Габор-фільтрі дозволяє виділити різні типи деталей та структур на зображеннях:

- Низькі частоти, виділяють великі та повільно змінювані структури;
- Середні частоти виділяють середньомасштабні текстури та чіткі контури;

- Високі частоти виділяють дрібні деталі та швидко змінювані текстури.

За даними робіт [30, 31] для зображень розміром 256x256 пікселів низьким частотам відповідають значення просторового періоду λ з діапазону від 50 до 256 пікселів, середнім – значення λ з діапазону від 10 до 50 пікселів, а високим – значення λ з діапазону від 2 до 10 пікселів. Для зображень розміром 512x512 пікселів низьким частотам відповідають значення λ з діапазону від 100 до 512 пікселів, середнім – значення λ з діапазону від 20 до 100 пікселів, а високим – значення λ з діапазону від 2 до 20 пікселів. Складається враження, що значення величини λ є залежним від розміру зображення. Однак, цей висновок є неправильним. Згідно дослідження [30], просторовий період λ Габор-фільтра є характеристикою самого фільтра, а це означає, що його значення не може бути залежним безпосередньо від розміру зображення. Від масштабу структур, які необхідно виділити на зображенні, та від роздільної здатності зображення є залежним не значення λ, а вибір цього значення.

Таким чином, Габор-фільтри є особливо ефективними у випадках аналізу локальних частотних характеристик зображень. Їх реалізація полягає у виділенні частотних складових в локальних областях зображення. Ці фільтри використовуються для виявлення специфічних текстурних патернів, що відповідають певним об'єктам або сценам.

На завершення аналізу Габор-фільтрів зазначимо, що, з огляду на дослідження [32], їх дуже легко налаштувати для виділення декількох ознак зображення відео одночасно. Одним із шляхів вирішення цієї задачі є використання набору Габор-фільтрів. Наприклад, набору таких двох фільтрів: фільтра $G(\lambda,\theta,\psi,\sigma,\gamma)$ з фіксованими значеннями параметрів $\theta,\psi,\sigma,\gamma$ та фільтра $G(\lambda,\theta,\psi,\sigma,\gamma)$ з фіксованими значеннями параметрів $\lambda, \psi,\sigma,\gamma$. Перший фільтр у цьому наборі має фіксовані $\theta,\psi,\sigma,\gamma$, що дає змогу виділяти ознаки на певній орієнтації та інших фіксованих параметрах, але змінюється λ для виділення різних частотних компонентів. Другий фільтр має фіксовані $\lambda,\psi,\sigma,\gamma$, що дає змогу виділяти ознаки на різних орієнтаціях θ при інших фіксованих параметрах. Іншими словами кожний фільтр у цьому наборі виділяє різні ознаки зображення

завдяки варіації одного з параметрів, що є необхідною умовою для отримання комплексної інформації про текстурні та структурні властивості зображення.

Простим, швидкісним та ефективним засобом для виділення, класифікації та аналізу тектурних зображень у відео є метод Local Binary Patterns (LBP). Цей метод грунтується на математичній моделі перетворення інтенсивності $I(x_c,y_c)$ центрального пікселя та $I(x_p,y_p)$ інтенсивностей P сусідніх пікселів, розташованих на фіксованій відстані R від центрального пікселя зображення відео, у двійковий код. Цей код прийнято називати LBP-кодом [31]. В основі формування цього коду покладено формулу:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s\left(I(x_p, y_p) - I(x_c, y_c)\right) \cdot 2^p. \quad (1.9)$$

Тут $s\left(I(x_p, y_p) - I(x_c, y_c)\right)$ – функція, яка приймає значення 1, якщо інтенсивність сусіднього пікселя перевищує або дорівнює інтенсивності центрального пікселя, в інших випадках вона приймає значення 0, тобто

$$s\left(I(x_p, y_p) - I(x_c, y_c)\right) = \begin{cases} 1, \text{якщо } I(x_p, y_p) - I(x_c, y_c) \geq 0; \\ 0, \text{якщо } I(x_p, y_p) - I(x_c, y_c) < 0. \end{cases} \quad (1.10)$$

Для повноти викладу аналізу LBP методу зазначимо, що детальне виділення текстурних ознак, локальний аналіз, обчислювальна ефективність та його практичне використання в різних застосуваннях є залежним від числа P, тобто кількості пікселів-сусідів центрального пікселя. Наочним підтвердженям цього твердження є результати досліджень [33, 34] у яких встановлено, що бажано щоб ціле число P не було меншим за 8 і не перевищувало числа 24. Надзвичайно цінними у цьому напрямі є емпіричні дослідження [33], автори яких встановили, що:

а) Чим більшим є значення числа P, тим більшою є здатність методу виконати аналіз більш складних та великомасштабних текстурних патернів;

б) При малих значеннях, наприклад для P=8 LBP – метод дає змогу на достатньо хорошому рівні провести локальний аналіз текстур, що добре підходить в основному лише для виділення дрібних і детальних текстурних ознак;

в) Чим меншим є значення числа P, тим меншою є обчислювальна складність методу. При великих значеннях P, наприклад, для P = 24 істотно збільшуються обчислювальні витрати, але зате суттєво може покращитися точність аналізу структур.

Таким чином, в контексті систем пошуку відео за фрагментами, LBP може бути використаний для створення унікальних підписів для кожного відеофрагмента на основі текстурних характеристик. Це дозволяє знаходити та ідентифікувати фрагменти, що містять схожі текстури, навіть при наявності змін у освітленні чи перспективі. Але містить ряд недоліків: аналізує текстуру лише в локальному масштабі, що може бути недостатнім для розпізнавання складних або великомасштабних текстурних візерунків, результати можуть сильно залежати від обраних параметрів, таких як кількість сусідніх пікселів.

Ще одним важливим атрибутом при аналізі об'єктів є формові атрибути, які відіграють ключову роль у системах пошуку відеоконтенту за його фрагментом. Вони використовуються для аналізу геометричних структур об'єктів у відео засобами визначення та порівняння таких контурних характеристик відеофрагментів як форма, розмір та орієнтація.

Перспективними методами аналізу формових атрибутів є методи виявлення контурів та областей, фур'є-перетворення форм, SIFT (Scale-Invariant Feature Transform) тощо. Проаналізуємо ці методи.

Методи виявлення контурів сегментації областей мають багато спільного, Зокрема, більшість з них ґрунтується на визначенні контура *C*, точніше такої замкненої кривої *C,* яка б з достатньо високою точністю описувала межі об’єкта на зображенні. За даними робіт [35, 36] такою кривою є крива, яка отримується шляхом мінімізації енергетичного функціоналу *E(C)*, математична модель для визначення якого має вигляд [35]:

$$E(C) = \oint_0^L \left(\alpha \left| \frac{dC(s)}{ds} \right|^2 + \beta \left| \frac{d^2C(s)}{ds^2} \right|^2 - \left| \nabla I\big(C(s)\big) \right| \right) ds. \qquad (1.11)$$

Тут: $\nabla I(C(s))$ – градієнт функції інтенсивності зображення у точці з координатами *(x, y)*; α, β – вагові коефіцієнти, що визначають вклад кожного

доданка підінтегральної суми; L – довжина замкненої кривої *C*; s – параметр кривої *C; C(s)* – функція параметризації, яка описує криву *C* у двовимірному просторі за допомогою параметра s, вона визначає координати точок на кривій *C* як функції від параметра s.

Мінімізацію енергетичного функціоналу *E(C)* можна здійснити різними методами, такими як метод активних контурів (snakes), градієнтний спуск, або інші чисельні методи оптимізації [35].

Методи виділення контурів і областей на зображеннях взаємодоповнюють один одного та часто застосовуються спільно для більш точного аналізу зображень та відео. Висока точність аналізу зображень забезпечується алгоритмами, які порівнюють виділені контури об'єктів у запитуваному фрагменті з контурними шаблонами у базі даних. Ці алгоритми дають змогу з достатньо високою точністю окреслити межі об’єктів на зображеннях, і тим самим істотно підвищити точність їх локалізації. Адже, згідно [36], чим точніше є визначені контури об’єктів на зображенні, тим точнішою є їх локалізація. Окрім цього, вони забезпечують можливість вилучення контурів, що істотно зменшує обсяг даних, які необхідно обробити. Тому, порівняно з іншими відомими методами, методи виділення контурів та сегментації областей на зображеннях є досить швидкими і не потребують значних обчислювальних ресурсів та пам'яті. Проте, незважаючи на це, використання цих методів у багатьох випадках є обмеженим. По-перше вони є чутливими до шуму і освітлення, що може призводити до появи помилкових контурів або розривів, особливо на складних сценах з низьким контрастом або неоднорідним фоном. По-друге їх ефективність сильно залежить від правильного вибору параметрів, і вони можуть потребувати значних обчислювальних ресурсів при роботі з великими зображеннями або відеопотоками. Крім того, ці методи можуть бути менш ефективними для об'єктів зі складними формами, що вимагають точного визначення меж. Ці недоліки можна усунути шляхом вдосконалення існуючих методів виділення контурів та сегментації областей на зображеннях, розробивши нові методи на основі поєднання відомих підходів [36].

Потужним інструментом для аналізу та розпізнавання форм є метод перетворення Фур'є форм (Fourier Shape Descriptors). Порівняно з проаналізованими вище методами виділення контурів та сегментації областей об’єктів на зображенні даний метод є більш стійким до змін положення та орієнтації об'єкта [37]. Він характеризується інваріантністю до обертання, масштабування та зсуву об'єкта. Крім цього він є менш чутливим до шуму і дає змогу отримати компактне представлення форми об'єкта за допомогою невеликої кількості коефіцієнтів, що значно спрощує порівняння та класифікацію форм. Завдяки ефективним алгоритмам обчислення, цей метод є швидким та надійним засобом обробки зображень.

Математична модель методу перетворення Фур'є форм ґрунтується на перетворенні зображення з просторової області у частотну область [37]. Згідно цієї моделі контур *C(s),* замкнена лінія, що обмежує об’єкт на зображенні, подається в параметричній формі запису, у вигляді комплексної функції:

$$C(s) = x(s) + jy(s), \tag{1.12}$$

де *x(s)* і *y(s)* – координати точки контуру, а s – параметр, що змінюється вздовж контуру.

Комплексна функція *C(s)* у (1.12) розкладається в ряд Фур'є

$$C(s) = \sum_{k=0}^{N-1} c_k e^{\frac{j2\pi ks}{T}}, \tag{1.13}$$

де j – уявна одиниця; N – загальна кількість точок в контурі; c_k – коефіцієнти Фур'є, значення яких визначаються за формулою:

$$c_k = \frac{1}{T}\int_0^T C(s) e^{\frac{-j2\pi ks}{T}} ds. \tag{1.14}$$

Тут T – період параметра s.

Згідно (1.12) – (1.14), будь-яку замкнену лінію на площині можна описати набором коефіцієнтів Фур’є, що має важливе значення для аналізу частотних спектрів об’єктів на зображенні. Адже вони дають змогу вивчити та дослідити амплітудно-частотні та фазо-частотні спектри цих об’єктів [37].

Спектр амплітуд визначається абсолютними значеннями коефіцієнтів:

$$|c_k| = \sqrt{Re(c_k)^2 + Im(c_k)^2}, \tag{1.15}$$

а спектр фаз – аргументами комплексних чисел:

$$arg(c_k) = arctg\frac{Im(c_k)}{Re(c_k)}. \quad (1.16)$$

Таким чином, коефіцієнти Фур'є визначають частотні компоненти контуру об'єкта. Низькочастотні компоненти відповідають загальній формі контуру, тоді як високочастотні компоненти описують деталі та текстуру. Інваріантність до різних трансформацій досягається шляхом нормалізації коефіцієнтів Фур'є:

- Інваріантність до зміщення: перший коефіцієнт c_0 представляє центр мас контуру і може бути використаний для вирівнювання контурів.
- Інваріантність до обертання: амплітуди коефіцієнтів c_k є інваріантними до обертання, тому можна використовувати лише модулі $|c_k|$.
- Інваріантність до масштабу: нормалізація шляхом ділення на модуль першого ненульового коефіцієнта $\left(\frac{c_k}{|c_1|}\right)$.

Отже, методи перетворення Фур'є є потужним інструментом для вирішення задач ідентифікації та порівняння об’єктів на зображеннях незалежно від їх положення, обертання та масштабу. Однак, точність опису форми сильно залежить від точності виділення контуру об'єкта.

Нарівні з проаналізованими вище методами та алгоритмами аналізу об’єктів на зображенні чи зображеннях, у сучасних системах пошуку відео широко використовується алгоритм Scale-Invariant Feature Transform (SIFT). Це обумовлено тим, що він є інваріантним до масштабування, обертання та зміщення. Тому SIFT є одним із найбільш ефективних алгоритмів для виявлення та опису локальних ознак, і завдяки цьому його часто застосовують для вирішення задач розпізнавання об'єктів, побудови панорам, 3D реконструкції та стеження за об'єктами. Однак його обчислювальна складність і високі вимоги до пам'яті є основними перешкодами для його використання в реальних додатках, які потребують обробки в реальному часі. Крім того, при значних змінах освітлення та перспективи цей метод є менш ефективним. У таких випадках його ефективність сильно залежить від вибору параметрів. Патентні обмеження також

обмежували його комерційне використання, що стимулювало розробку альтернативних методів.

Для порівняння SIFT-алгоритму з модифікованими та альтернативними алгоритмами, а також з метою його адаптації для вирішення завдань у даній дисертаційній роботі наведемо основні етапи його практичної реалізації [38]:

1. *Виявлення екстремумів у масштабному просторі*. На цьому етапі здійснюється побудова масштабного простору зображення та вирішується задача виявлення у ньому ключових точок, стійких до змін масштабу. Масштабний простір L(x,y,σ) отримується шляхом згладжування зображення I(x,y) з використанням функції Гаусса:

$$L(x, y, \sigma) = G(x, y, \sigma) \cdot I(x, y). \quad (1.17)$$

Тут *G(x,y,σ)* – гаусова функція з параметром масштабування σ:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right). \quad (1.18)$$

Для виявлення екстремумів у масштабному просторі обчислюється різниця гаусових зображень DoG:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma), \quad (1.19)$$

де k – коефіцієнт масштабу.

Локальні екстремуми (максимуми та мінімуми) в DoG зображеннях виявляються шляхом порівняння кожної точки з її сусідами в поточному та сусідніх масштабах.

2. *Локалізація ключових точок*. На цьому етапі реалізації SIFT-алгоритму вирішується задача точної локалізації ключових точок шляхом підгонки (фіттингу) квадратичного полінома до значень пікселів сусідніх точок. Точки з низькою контрастністю та ті, що розташовані на краях, відсіюються, оскільки вони є менш надійними.

3. *Призначення орієнтації*. Для кожної ключової точки обчислюються градієнти інтенсивності у локальному сусідстві. Створюється гістограма орієнтацій градієнтів, і домінуюча орієнтація призначається ключовій точці. Це забезпечує інваріантність до обертання. Модуль градієнта m(x,y) і напрямок градієнта θ(x,y) визначаються за формулами:

$$m(x,y) = \sqrt{\big(L(x+1,y) - L(x-1,y)\big)^2 + \big(L(x,y+1) - L(x,y-1)\big)^2}; \quad (1.20)$$

$$\theta(x,y) = arctg\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right). \quad (1.21)$$

4. *Формування дескриптора*. На заключному етапі формується вектор дескриптора для кожної ключової точки. В межах локального сусідства ключової точки обчислюються градієнти, які розбиваються на менші підрегіони (зазвичай 4x4). Для кожного підрегіону формується гістограма орієнтацій градієнтів (зазвичай з 8 бінів), що забезпечує 128-елементний дескриптор (4x4x8). Дескриптор нормалізується для забезпечення інваріантності до змін освітлення.

У сучасних системах пошуку відео за фрагментом високими показниками продуктивності характеризуються алгоритми, які отримані шляхом модифікацій SIFT- алгоритму а саме:

- PCA-SIFT (Principal Component Analysis SIFT): Ця модифікація застосовує метод головних компонент для зменшення розмірності дескрипторів SIFT, що дозволяє значно знизити обчислювальні витрати та покращити швидкість обробки, зберігаючи при цьому високу точність [39].
- CSIFT (Color SIFT): CSIFT розширює класичний SIFT шляхом додавання інформації про колір. Це дозволяє алгоритму краще розпізнавати об'єкти у кольорових зображеннях та відео, забезпечуючи кращу стійкість до змін освітлення та кольорових спотворень [39].
- GLOH (Gradient Location and Orientation Histogram): GLOH є удосконаленою версією SIFT, яка використовує гістограму місцезнаходження та орієнтації градієнтів. Ця модифікація забезпечує більш детальне кодування просторових розподілів градієнтів, що покращує точність виявлення та опису ключових точок [40].
- ASIFT (Affine SIFT): ASIFT розширює SIFT для обробки афінних перетворень зображень. Це дозволяє алгоритму ефективно працювати з зображеннями та відео, які піддаються сильним геометричним спотворенням, таким як нахили та деформації.

- Dense SIFT: На відміну від класичного SIFT, який працює з окремими ключовими точками, Dense SIFT обчислює дескриптори для кожної точки в регулярній сітці. Це дозволяє отримати більш повне представлення зображення або відеофрагменту, що покращує точність пошуку [39].

Перспективним напрямком розвитку методів обробки відео є інтеграція та удосконалення існуючих алгоритмів класифікації, розпізнавання та індексації відеоданих. Наочним підтвердженням цього висновку є дослідження [26]. Його автори шляхом інтеграції методів аналізу колірних ознак об'єктів, гістограм ознак та кореляційного аналізу синтезували модель автоматичного екстрагування об'єктів, концепцій та подій із бази даних. На конкретних прикладах довели, що впровадження цієї моделі у традиційні системи контентно-орієнтованого відеопошуку (CBVIR) значно підвищує точність і ефективність пошуку відео за змістом. Окрім цього показали, що отримана ними нова система CBVIR забезпечує більш релевантні результати пошуку відео та істотно зменшує кількість помилкових спрацьовувань. Ця публікація свідчить, що використаний її авторами підхід може суттєво оптимізувати процес аналізу великих відеоархівів та полегшити доступ до потрібної інформації у різних галузях, включаючи безпеку, медицину, розваги та освіту.

Не менш важливою для обґрунтування способу удосконалення існуючих та розроблення нових методів обробки відео шляхом їх об’єднання та подальшої інтеграції у традиційні системи відеопошуку (CBVIR) є робота [27]. Її автори, використовуючи метод машинного навчання – метод підтримкових векторних машин (SVM), побудували модель для ранжування концепцій відео і на конкретних прикладах довели, що її впровадження у традиційні CBVIR сприяє більш точному та швидкому знаходженню відповідних відео. Окрім цього показали, що для покращення роботи цієї моделі у середовищі CBVIR доцільним є використання текстурних характеристик об'єктів, аналіз яких здійснюється методами Габор-фільтрів або Local Binary Patterns (LBP). Оригінальною щодо практичного застосування передових технологій для оптимізації процесів пошуку відео за допомогою існуючих систем CBVIR є робота [31], у якій

викладено особливості впровадження методів глибокого навчання у традиційні CBVIR для підтримки медіа-продукції під час інспекції та пошуку відео. Автори цієї роботи встановили, що ефективність роботи нейронної мережі можна значно покращити, якщо попередньо обробити вхідні дані – колірні та текстурні характеристики відео – методами кореляційного аналізу та LBP (Local Binary Patterns). Показали, що ця процедура попередньої обробки значно знижує обчислювальні витрати та підвищує ефективність системи CBVIR загалом.

Таким чином, інтеграція математичних методів обробки колірних, текстурних та геометричних характеристик об'єктів відео у традиційні CBVIR істотно підвищує ефективність цих систем. Вона забезпечує більш точне виявлення, класифікацію та індексацію відеоданих, що має надзвичайно важливе значення для розвитку технологій автоматизованого аналізу та пошуку відео за окремими кадрами. У таблиці наведено результати порівняльного аналізу застосування методів для інформації у відеопотоках.

Таблиця1.1. Порівняльна характеристика методів аналізу відеопотоків

Метод	Для призначений	Дослідники	Переваги	Обмеження
SBD (Shot Boundary Detection)	Автоматичне виявлення меж (переходів) між сценами у відеопотоці (різкі\поступові зміни)	T. Kim, C. Zhang	Спрощує подальшу обробку відео, розділяючи його на логічні епізоди. Полегшує індексацію сцен. Швидка обробка простих переходів.	Можливі похибки при складних «м’яких» переходах (розмиття, затемнення). Чутливість до шуму та динамічних ефектів (титри, графіка).
Lucas-Kanade (оптичний потік)	Оцінювання руху пікселів у послідовних кадрах, відстеження рухомих	R. Wang, H. Zhou	Простота реалізації та відносно висока швидкодія. Дає змогу відсте-	Чутливість до значних змін освітлення чи масштабу. При швидких рухах або

	об’єктів і аналіз динамічних змін		жувати об’єкти в реальному часі. Широко застосовується у задачах спостереження.	великих деформаціях може втрачати точність.
Фільтр Калмана	Рекурсивна оцінка стану рухомих об’єктів з урахуванням шумів, відстеження траєкторій в реальному часі	M. Patel, L. Dupont	Ефективна фільтрація шумів, згладження траєкторій. Низькі обчислювальні вимоги порівняно з нелінійними методами. Успішно працює в реальних системах моніторингу.	При сильній нелінійності (різких змінах) точність падає. Потребує точної моделі динаміки та шуму, інакше можуть виникати значні відхилення в оцінках.
SIFT (Scale Invariant Feature Transform)	Виокремлення та опис масштабно-інваріантних ключових точок на зображеннях (кадрах) для пошуку збігів і порівняння	A. Vedaldi Y. Liu	Стійкість до змін масштабу, поворотів та частково освітлення. Висока точність знаходження відповідностей об’єктів (feature matching).	Високі обчислювальні витрати на великих наборах даних. Чутливість до екстремальних змін перспективи або дуже сильного шуму.
SURF (Speeded-Up Robust Features)	Прискорений пошук ключових точок та їх опис (порівняно з SIFT)	M. Tola, Q. Feng	Швидша робота, ніж SIFT, зі збереженням відносної стійкості до шумів. Збалансоване співвідношення	У складних або низькоякісних відео SIFT часто демонструє кращу точність. Не такий гнучкий при сильних зміненнях

			«точність/ швидкість».	перспективи (3D-potations).
HOG (Histogram of Oriented Gradients)	Виділення та опис градієнтних ознак для детекції та розпізнавання (особливо ефективно для виявлення пішоходів)	N. Zhang, P. Roy	Простий у реалізації алгоритм, добре працює для впізнавання типових об’єктів люди, транспорт). Високі показники для задач виявлення пішоходів.	Слабка стійкість до великих змін масштабу й фону. При насичених відеосценах (з великою кількістю різних об’єктів) ефективність знижується.
LBP (Local Binary Patterns)	Опис локальних текстур для класифікації зображень і пошуку подібних фрагментів (зокрема, розпізнавання облич)	H. Wei, G.Balakrishnan	Низька складність реалізації, висока швидкість. Добре працює за стабільних умов освітлення, підходить для аналітики текстур.	Обмежена стійкість до великих змін освітлення чи масштабу текстур. Менш ефективний для дуже складних «мультимасштабних» текстур.
Габор-фільтри (Gabor Filters)	Виділення частотних і просторових особливостей (текстурних деталей), аналіз локальних текстур і контурів	J. Bai, K. Inoue	Добра здатність до виділення орієнтаційних та частотних компонент. Ефективні для розпізнавання патернів і аналізу фактур.	Підбір параметрів (частоти, орієнтації) може бути складним. Порівняно висока обчислювальна вартість на великих відеокадрах.
Фур’є-опис форм	Опис форм у частотній області, щоб порівнювати об’єкти та зображення незалежно від	S. Paul, E. Romero	Інваріантність до обертання, зміщення й масштабу. Компактне «спектральне» пред-	Потребує точного виділення контурів об’єктів. За сильних викривлень, розмитих меж чи шуму можуть

	обертання/масштабування		ставлення контуру.	виникати суттєві спотворення опису.

1.3. Вектори ознак та метрики подібності

Сьогодні для комплексного та детального опису характеристик відео використовуються не лише гістограми, графові моделі, моменти, текстурні дескриптори, часові характеристики, але також і вектори ознак, які є не менш ефективним способом представлення відео, зображення та тексту у числовій формі. Це обумовлено тим, що довжини векторів ознак є фіксованими, вони мають однакову кількість елементів, і кожний елемент містить інформацію лише про одну конкретну ознаку об’єкта, що значно спрощує порівняльний аналіз об’єктів на відео [41]. Завдяки цим особливостям цей спосіб вирізняється від зазначених вище способів тим, що він дає змогу, по-перше, компактно і ефективно зберігати багатовимірну інформацію. По-друге математичні операції над векторами ознак (обчислення відстаней або подібностей) є набагато простішими, ніж операції над гістограмами, графовими моделями, моментами, текстурними дескрипторами, часовими характеристиками, що має важливе значення для вирішення задач швидкого та точного визначення подібності між різними об’єктами. По-третє у випадку векторного представлення відео, зображення та тексту можливим є ефективне застосування потужних алгоритмів машинного та глибинного навчання для підвищення точності їх обробки.

Таким чином, використання векторів для представлення характеристик відео є не лише практичним, але й необхідним для досягнення високої ефективності і точності в задачах аналізу і пошуку подібних відео.

Однією з найбільш важливих задач аналізу та пошуку подібних відео є задача визначення подібності векторів ознак. Згідно робіт [42, 43, 44] сьогодні найбільш перспективними методами її вирішення є методи, які використовують різні метрики відстані або подібності. З огляду на тему дисертаційної роботи найбільшої уваги на аналіз та подальше використання заслуговують такі методи:

косинусна подібність, Евклідова відстань, Мангеттенська відстань, міра Жаккара, міра Пірсона.

1.3.1. Косинусна подібність

Косинусна подібність (Cosine Similarity) широко використовується для вимірювання схожості між двома векторами ознак в багатовимірному просторі [42]. Згідно з цим методом, два вектори ознак вважаються відносно подібними, якщо вони напрямлені в одному напрямку . Наприклад, градієнти інтенсивностей $I(x_1,y_1)$ та $I(x_2,y_2)$ пікселів точок (x_1,y_1) та (x_2,y_2) на деякому зображенні вважаються відносно подібними, якщо їхні напрямки є однаковими. Це означає, що напрямки максимальних змін яскравостей пікселів точок (x_1,y_1) та (x_2,y_2) співпадають. Іншими словами напрямки переходів від темних до світлих областей в точках (x_1,y_1) та (x_2,y_2) є однаковими, а косинус кута між векторами $\nabla I(x_1,y_1)$ та $\nabla I(x_2,y_2)$ дорівнює 1, або є дуже близьким за значенням до 1.

Формально критерієм косинусної подібності двох векторів A і B є формула визначення коснуса кута між ними:

$$cosine\_similarity(A,B) = \frac{A \cdot B}{\|A\|\|AB\|}. \quad (1.22)$$

Тут A·B – скалярний добуток (внутрішній добуток) векторів A і B, а ‖A‖ і ‖B‖ – їхні евклідові норми (довжини), значення яких визначається за формулами:

$$\|A\| = \sqrt{\sum_{i=1}^{n} A_i^2}; \qquad \|B\| = \sqrt{\sum_{i=1}^{n} B_i^2}, \quad (1.23)$$

де A_i та B_i – компоненти векторів A та B відповідно.

З аналізу формул (1.22) – (1.23) випливає, що значення косинусної подібності варіюються в межах від -1 до 1. Значення, близькі до 1, вказують на високу подібність між векторами, що означає, що вектори спрямовані в один бік. Значення, близькі до -1, вказують на антиподібність векторів, тобто вони є протилежно напрямленими. Значення 0 свідчить про ортогональність векторів, тобто відсутність подібності між ними. Це означає, що косинусна подібність є не визначеною для векторів з нульовою довжиною і є менш інформативною для векторів з негативними значеннями величини cosine_similarity(A,B). Його

використання є недоцільним у випадку обробки векторів ознак, абсолютні значення яких містять важливу інформацію про об'єкт відео. Більше цього згідно [42] цей метод є чутливим до шуму даних. Однак, не зважаючи на ці недоліки косинусна подібність є обчислювально ефективним методом, який все ж таки дає змогу вимірювати схожість між векторами, незалежно від їхньої довжини. Вона є ідеальним методом для вирішення задач порівняння об'єктів з подібним розподілом ознак, і завдяки цьому широко застосовується до різних типів даних, зокрема до текстових документів та зображень.

1.3.2. Евклідова відстань

Одним із найпростіших і найбільш інтуїтивно зрозумілих методів вимірювання подібності між векторами ознак є метод Евклідова відстань (Euclidean Distance). Він грунтується на формулі визначення відстані між двома точками у багатовимірному просторі, у так званому n-вимірному Евклідовому просторі [43]:

$$Euclidean\_distance = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}, \quad (1.24)$$

де $A_1,A_2,\ldots,A_n$ та $B_1,B_2,\ldots,B_n$ – координати точок A та B в n-вимірному просторі.

З (1.24) випливає, що метод Евклідова відстань є чутливою до масштабу даних. Змінні з великими числовими значеннями можуть домінувати над змінними з меншими значеннями, що може істотно спотворювати результати кластеризації, класифікації та аналізу даних. Для наочного підтвердження цього твердження визначимо Евклідову відстань між точками A(1,7;70) та B(1,8;65), де 1,7 та 1,8 – значення ознак висоти (у метрах), а 70 та 65 – значення ознак маси (у кілограмах). Для цього скористаємося формулою (1.24). У результаті отримаємо:

$$Euclidean\_distance = \sqrt{(1{,}7 - 1{,}8)^2 + (70 - 65)^2} \approx 5. \quad (1.25)$$

У цьому прикладі значення Евклідової відстані визначається в основному різницею маси (5 кг), вплив різниці висоти (0.1 м) на значення величини *Euclidean_distance* є не значним. Різниця маси домінує у значенні Евклідової відстані навіть у випадку коли обидві ознаки є рівноважливими. Тому, оскільки

формула (1.25) не адекватно відображає вплив різних ознак об’єктів відео на значення Евклідової відстані, то отримані за її допомогою результати кластеризації, класифікації та аналізу даних у багатьох випадках можуть виявитися спотвореними. Однак, це не означає, що цей метод на сьогодні не є актуальним. Навпаки сьогодні він вважається ефективним інструментом для пошуку об’єктів відео, особливо у випадках, коли ознаки об’єктів мають однаковий масштаб та є нормалізованими [43]. Справді, з огляду на наведений вище приклад, якщо координати точок A та B подати у нормалізованому вигляді:

$$A\left(\frac{1{,}7-1{,}7}{1{,}8-1{,}7};\frac{70-65}{70-65}\right)=A(0;1);\qquad B\left(\frac{1{,}8-1{,}7}{1{,}8-1{,}7};\frac{65-65}{70-65}\right)=B(1;0),\quad(1.26)$$

то після підстановки у (1.24) нормалізованих значень координат точок A і B одержимо таке значення Евклідової відстані, впливи різниці маси (5 кг) та різниці висоти (0.1 м) на яке є рівнозначними:

$$Euclidean\_distance=\sqrt{(0-1)^2+(1-0)^2}\approx 1{,}41.\qquad(1.27)$$

Дійсно, результат (1.27) істотно відрізняється від результатів, які б ми отримали, якщо б знехтували впливом маси або впливом висоти. Окрім цього, у будь-якому випадку, чи у випадку знехтування впливу маси, чи у випадку знехтування впливу висоти, значення величини *Euclidean_distance* дорівнювало б одиниці. Розглянутий приклад доводить, що високу чутливість методу Евклідова відстань до масштабу даних можна істотно знизити шляхом нормалізації або стандартизації цих даних, і таким чином усунути даний недолік методу у випадку його застосування [43].

Ще не менш важливими недоліками методу Евклідової відстані є те, що він не є інваріантним до зміщення даних, а обчислення Евклідової відстані для всіх пар точок у великих наборах даних можуть виявитися обчислювально дорогими. Але їх можна легко усунути. Перший недолік усувається шляхом використання центрованих координат або іншого методу вимірювання Евклідової відстані, який є інваріантним до зміщення, а другий – використанням підходів, що зменшують обчислювальну складність і пришвидшують процес розрахунку Евклідової відстані для всіх пар точок у великих наборах даних.

Отже, з врахуванням зазначеного вище, метод Евклідової відстані є простим для розуміння та реалізації, і тому він є надзвичайно зручним для базових аналізів та швидкого прототипування продуктів чи систем. Евклідова відстань між точками у просторі є інтуїтивно зрозумілою, вона дає змогу наочно відобразити міру подібності: чим меншим є значення цієї відстані, тим вищою є подібність між об'єктами. Крім того, метод широко використовується у багатьох алгоритмах машинного навчання, таких як k-середніх кластеризація та k-найближчих сусідів класифікація [42].

1.3.3. Мангеттенська відстань

Окрім розглянутих вище методів, сьогодні надзвичайно популярним методом вимірювання подібності векторів у багатовимірному просторі є також і метод Мангеттенська відстань (Manhattan Distance), який прийнято називати метрикою прямокутного міста. В основі цього методу покладено формулу визначення Мангеттенської відстані [44]:

$$Manhattan\_distance(A, B) = \sum_{i=1}^{n} |A_i - B_i|, \qquad (1.28)$$

значення якої дорівнює довжині шляху пройденого матеріальною точкою з точки А у точку В у випадку, коли її рух відбувається вздовж ліній, паралельних осям координат декартового n-вимірного простору. Тут A_i та B_i – координати точок A та B в n-вимірному просторі.

Важливість формули (1.28) полягає у тому, що вона дає можливість виявити не лише переваги даного методу, але і його недоліки. Справді, оскільки, згідно з (1.28), метрикою Мангеттенської відстані між двома точками А і В є довжина ламаної АВ, відрізки якої є паралельними осям декартової системи координат n-вимірного простору, то неважко зауважити, що ця метрика не завжди дає змогу кількісно оцінити відстань між двома точками цього простору. Зокрема, малоефективним є застосування цього методу у випадку вирішення задач, де важливими є діагональні або непрямі маршрути.

З (1.28) випливає також, що метод Мангеттенська відстань є надзвичайно ефективним та швидким у реалізації у випадку визначення відстані між двома

точками у n-вимірних просторах з прямокутною сіткою. Окрім цього на відміну від методу Евклідова відстань він є менш чутливим до великих відхилень у значеннях окремих координат, що може бути корисним у задачах з наявністю шуму або викидів [44].

1.3.4. Міра Жаккара

Не менш популярним методом вимірювання подібності між двома скінченними наборами даних є міра Жаккара (Jaccard Similarity). Формально, міра Жаккара для двох множин A і B визначається відношенням розмірностей перетину та об'єднання цих множин [44, 45]:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{1.29}$$

де |A∩B| – кількість елементів у перетині множин A і B, а $|A \cup B|$ – кількість елементів в їх об'єднанні.

Елементами множин A і B можуть бути будь-які об'єкти або ознаки, які потрібно оцінити на подібність або спільність, наприклад, набори ключових точок двох зображень, набори слів у двох різних текстах, два кластери об’єктів у базі даних, тощо.

З (1.29) випливає, що оскільки, розмір множини перетину двох множин не перевищує розміру множини їх об'єднання, то значення міри Жаккара є нормалізованими значеннями. Це означає, що:

$$0 \leq J(A,B) \leq 1; \tag{1.30}$$

$$J(A,B) = \{0 \text{ різні};\ \ 1 \text{ є ідентичними}\} \tag{1.31}$$

Для повноти аналізу метрики Жаккара розглянемо особливості його практичного використання на прикладі вирішення задачі визначення подібності двох груп людей на основі порівняння їхніх уподобань (хобі), які є визначеними множинами A={читання, подорожі, спорт, фотографія} та B={спорт, фотографія, музика, малювання} відповідно.

Щоб вирішити цю задачу виходитимемо з того, що:

$$A \cap B = \{\text{спорт, малювання}\};$$

$$A \cup B = \{\text{читання, подорожі, спорт, фотографія, музика, малювання}\}.$$

Тоді, оскільки за визначенням [ ]: $|A \cap B| = 2,\ |A \cup B| = 6$, то, згідно формули (1.29), $J(A,B) = 2 / 6 \approx 0{,}333$.

Отже, міра Жаккара для множин *A* та *B* становить приблизно 0.333. Це означає, що подібність між цими двома групами людей на основі їхніх хобі становить приблизно 33,3 % [45].

На завершення аналізу метрики Жаккара зазначимо, що цей метод використовується переважно для аналізу бінарних атрибутів, але за потреби його можна досить легко адаптувати і до інших типів даних. Окрім цього, зазначимо, що даний метод не враховує кількісних відмінностей між елементами множин ознак. Він використовується в основному для оцінки наявності чи відсутності елементів ознак у цих множинах, і тому є менш точним у випадку обробки множин з різними розмірами, оскільки менші за розміром множини мають менше шансів на значний перетин.

1.3.5. Метод Пірсона

Одним із найбільш ефективних засобів для виявлення та аналізу лінійної залежності між двома змінними є метод Пірсона (Pearson Correlation). Особливо ефективним є його використання у випадку вирішення задач параметричної ідентифікації, зокрема у випадках статистичної обробки векторів ознак, значення компонентів яких підлягають нормальному розподілу [8]. Цей метод дає змогу визначити ступінь лінійної кореляції між двома векторами у багатовимірному просторі ознак відео. Його використовують в основному для оцінки того, наскільки зміни в одному векторі пов'язані зі змінами в іншому векторі.

Математичною основою методу Пірсона є формула для визначення коефіцієнта кореляції Пірсона між двома векторами $X=(X_1,X_2,\dots X_n)$ та $Y=(Y_1,Y_2,\dots Y_n)$:

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i-\underline{X})(Y_i-\underline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\underline{X})^2 \sum_{i=1}^{n}(Y_i-\underline{Y})^2}}, \tag{1.32}$$

де X_i та Y_i – числові значення ознак відео;

$\underline{X}$ та $\underline{Y}$ – середні значення векторів X та Y відповідно.

Значення коефіцієнта кореляції r_{XY} варіюються від -1 до 1. Близькі до одиниці значення r_{XY} свідчать про сильну позитивну лінійну залежність між двома векторами ознак X та Y. Іншими словами, якщо значення компонетів одного вектора зростають, то значення компонентів іншого вектора також зростають, і навпаки. Чим більш наближеними є значення коефіцієнта r_{XY} до 1, тим сильнішою є ця позитивна залежність.

Близькі до -1 значення r_{XY} свідчать про сильну негативну лінійну залежність між значеннями компонентів X_i та Y_i векторів X та Y відповідно. Іншими словами, якщо значення компонентів одного вектора зростають, то значення компонентів іншого вектора, як правило, зменшуються, і навпаки. Чим більш наближеним є значення коефіцієнта r_{XY} до -1, тим більш сильною є ця негативна залежність.

Значення коефіцієнта r_{XY}, які є близькими до нуля, свідчать про дуже слабку лінійну залежність між компонентами векторів X та Y. Іншими словами, зміни компонентів одного вектора не дають змоги передбачити зміни значень компонентів іншого вектора [36].

Таким чином, з огляду на вище викладене, метод Пірсона є простим у практичній реалізації, а отримані за його допомогою результати інтуїтивно зрозумілими для інтерпретації. Незважаючи на те, що він є неінформативним для вирішення задач нелінійної параметричної ідентифікації, він все ж таки заслуговує уваги, оскільки дає змогу встановити, чи є зв'язки між векторами ознак лінійними чи нелінійними.

Викладений порівняльний аналіз метрик подібності векторів ознак показав, що кожна з них має свої переваги та недоліки. Доцільність практичного використання тієї ж чи іншої метрики визначається особливостями задачі яку необхідно вирішити, та типом вхідних даних. Це означає, що в одних випадках найоптимальнішим методом для аналізу векторів ознак відео може виявитися, наприклад, метрика Жаккара, а в інших – будь-яка інша з проаналізованих вище метрик, наприклад, метод Евкліда. Щоб встановити, який із цих методів є найбільш ефективним у конкретному випадку, достатньо провести відповідні

числові експерименти та виконати порівняльний аналіз отриманих результатів. Тому одним із основних завдань даної дисертаційної роботи є обґрунтування вибору найоптимальнішої метрики шляхом порівняльного аналізу результатів відповідних числових експериментів, зокрема результатів обробки векторів ознак за допомогою проаналізованих вище методів. Вирішення цього завдання є необхідним кроком для оптимізації процесу аналізу відео та удосконалення існуючих систем пошуку подібних відео на основі векторних ознак.

1.4. Постановка задачі

Головною метою цього дисертаційного дослідження є розробка нових моделей, методів і засобів аналізу відеопотоків для підвищення точності пошуку і класифікації динамічних станів об'єктів у контентно-орієнтованих системах відеоінформації з урахуванням обмежених обчислювальних ресурсів. Особливу увагу приділено створенню інформаційної системи, яка дозволяє ефективно здійснювати пошук відео за його довільним фрагментом, використовуючи глибокі згорткові нейронні мережі (DCNN), метод пошуку меж сцен (SBD) та нейронні мережі зворотного поширення (FNN) для контролю і оптимізації роботи DCNN. Система повинна забезпечувати високу швидкість, точність та гнучкість інтеграції з існуючими рішеннями для управління медіаконтентом.

Для досягнення поставленої мети були визначені наступні завдання:

1. Провести порівняльний аналіз сучасних архітектур, моделей та методів обробки та аналізу відеопотоків, зокрема методів сегментації, детекції сцен, ідентифікації та класифікації відеоданих для визначення оптимальних рішень індексації відео.

2. Розробити методи інтеграції просторових і часових характеристик відеоданих з використанням DCNN для забезпечення ефективної індексації, пошуку та класифікації динамічних станів.

3. Створити модель для аналізу геометричних ознак та гістограм градієнтів у відеопотоках, що дозволить підвищити точність пошуку та класифікації відео порівняно з класичними підходами до CBVIR-систем.

4. Розробити методи виявлення меж сцен із застосуванням алгоритму SBD для розділення відеопотоків на сцени, інтегруючи ці методи в контентно-орієнтовані системи пошуку (CBVIR).

5. Запропонувати та реалізувати алгоритми сегментації та виділення ознак об'єктів на ключових кадрах, адаптовані до роботи за умов обмежених обчислювальних ресурсів, з подальшим збереженням інформації в NoSQL базах даних.

6. Розробити методологію клієнтського аналізу, яка включає завантаження фрагменту відео, визначення меж сцен методом SBD, виділення ознак і формування комбінаційних векторів характеристик (PONF, MCWF, CCWF, SCWF).

7. Використати DCNN для визначення подібності за допомогою Temporal Similarity Matching (TSIM) та Object Similarity Matching (OSIM), сформувати первинний список подібностей (PLIST), провести його оптимізацію та отримати фінальний список подібностей відео.

8. Дослідити і впровадити нейронні мережі зворотного поширення (FNN) для контролю та оптимізації роботи DCNN, підвищуючи тим самим точність і ефективність процесу пошуку відео за довільним фрагментом.

9. Створити архітектуру і розробити прикладну інформаційну систему для апробації запропонованих методів, яка відповідає вимогам високої точності, швидкодії та готовності до інтеграції з існуючими системами управління відеоконтентом.

Висновки до 1 розділу

На основі аналізу існуючих підходів та методів обробки тимчасових характеристик відео виявлено, що сучасний розвиток технологій і систем контентно-орієнтованого відеопошуку (CBVIR) здебільшого зосереджується на вдосконаленні відомих методів сегментації відео, комбінуванні тимчасових характеристик (таких як тривалість сцен, часові відмітки, порядок подій) та їх інтеграції з іншими типами даних, зокрема характеристиками об’єктів відео.

Показано, що одним із найбільш ефективних способів автоматизації процесів обробки відеопотоків є поєднання тимчасових характеристик із характеристиками об’єктів відео. Такий підхід сприяє підвищенню точності аналізу відеоконтенту, оптимізації процесів пошуку та класифікації, а також покращенню загальної продуктивності систем CBVIR, особливо в умовах великих обсягів даних і динамічних змін у вхідному контенті.

Встановлено, що для досягнення високої ефективності та точності в задачах аналізу й пошуку подібних відео доцільно представляти характеристики відео у вигляді векторів ознак, а також визначено найбільш перспективні методи формування цих векторів. Серед таких методів особливо виділяються алгоритми глибокого навчання, зокрема згорткові нейронні мережі (DCNN) для екстракції просторових ознак та рекурентні нейронні мережі (LSTM) для моделювання часових залежностей. Крім того, інтеграція технік обробки тимчасових характеристик, таких як аналіз гістограм, оптичного потоку та виявлення ключових точок, дозволяє створювати більш інформативні вектори ознак, що підвищує ефективність і надійність пошукових систем.

РОЗДІЛ 2. РОЗРОБКА ОБЧИСЛЮВАЛЬНИХ МОДЕЛЕЙ ДЛЯ ІНДЕКСАЦІЇ ВІДЕО

Даний розділ роботи присвячений викладу основних етапів побудови обчислювальної моделі індексації відео, а саме: ретельному обгрунтуванню вибору архітектури нейронної мережі для обробки характеристик відеоконтенту та розробкам обчислювальних моделей для детекції та екстракції характеристик сцен у відеопотоках. Окрім цього, у ньому розглянуто питання оптимізації моделей для підвищення їх ефективності, а також наведено результати експериментів, що підтверджують доцільність застосованих підходів.

Матеріали розділу опубліковані у роботах автора [44, 52].

2.1. Обгрунтування вибору нейронної мережі для обробки характеристик відеоконтенту

Ефективним та надзвичайно потужним інструментарієм для вирішення задачі пошуку відео за його довільним фрагментом є обчислювальні моделі – нейронні мережі. На відміну від традиційних методів обробки характеристик відео, нейронні мережі є більш досконалими засобами для автоматичного перетворення характеристик об'єктів фрагментів відео у вектори ознак. Їхня перевага полягає у здатності автоматично розпізнавати важливі ознаки відеофрагментів засобами глибокого навчання, що надає їм змогу ефективно працювати з великою кількістю даних і складними структурами, які можуть бути недосяжними для традиційних методів [2,7,8].

Сьогодні існує багато різних типів нейронних мереж, але це не означає, що для вирішення задачі пошуку відео за його довільним фрагментом потрібно викласти аналіз усіх існуючих мереж. У даному випадку найбільш важливими є мережі, які найбільше є адаптованими до процесів обробки даних з часовою та послідовною структурами. Це пояснюється тим, що основні типи даних у відео – візуальні дані (послідовність кадрів) та аудіодані – характеризуються часовою та послідовною структурою.

2.1.1. Рекурентні нейронні мережі

Рекурентні нейронні мережі (RNN) та їхній удосконалений варіант – довготривала короткочасна пам'ять (LSTM) – є ключовими інструментами для обробки даних з часовою залежністю. На відміну від звичайних нейронних мереж, які обробляють кожен вхідний сигнал незалежно, RNN здатні зберігати контекст попередніх станів, що дозволяє ефективно аналізувати послідовності даних, зокрема відеопотоки. Це особливо важливо для виявлення руху, визначення послідовності подій та розпізнавання закономірностей у часових серіях. Для повноти викладу аналізу цих мереж розглянемо більш детально основні алгоритми та методи, що використовуються в RNN та LSTM [46]:

– ***Recurrent Layer (Рекурентний Шар)***. Рекурентні шари є базовими (основними) компонентами архітектури RNN мереж. Вони призначені для обробки послідовних даних. На кожному кроці їх роботи враховується інформація з попередніх кроків (контекст). На кожному кроці, в момент часу t, рекурентні шари обробляють вхідні дані x_t і дані h_{t-1} з попереднього кроку, і на основі цього генерують вихідне значення h_t. Формально робота рекурентних шарів описується формулою [47]:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b_h), \tag{2.1}$$

де h_t – вихід (або стан) на поточному кроці часу, h_{t-1} – вихід на попередньому кроці часу, x_t – вхідний сигнал на поточному кроці часу, W_h та W_x – вагові матриці для попереднього стану та поточного входу відповідно, b_h – зміщення, σ – нелінійна функція активації (наприклад tanh або ReLU).

Зокрема, з (2.1) випливає, що на поведінку та ефективність роботи RNN мережі істотно впливає функція активації σ. Завдяки можливості вибору функції активації RNN можуть адаптуватися до різних типів вхідних даних, зберігаючи здатність моделювати як лінійні, так і нелінійні часові залежності. Наприклад, використання tanh(x) забезпечує згладжування сигналу, тоді як ReLU(x) дозволяє зберігати важливі імпульсні особливості. Таким чином, вибір функції активації може істотно впливати на здатність моделі до навчання та узагальнення. Окрім цього з (2.1) випливає, що: збільшення розміру послідовних даних призводить до

зростання обчислювальної складності рекурентних нейронних шарів [46]. Це пов'язано з тим, що рекурентні нейронні мережі обробляють дані послідовно, що унеможливлює паралельну обробку різних кроків у часі. У зв’язку з цим збільшення розміру послідовності безпосередньо впливає на обчислювальні витрати та час виконання обчислень. Чим довшою є послідовність, тим більше обчислень потрібно виконати, що збільшує загальні витрати часу та ресурсів.

На завершення викладу аналізу стандартних RNN мереж зазначимо, що вони мають обмежені можливості щодо запам'ятовування та врахування інформації з далекого минулого [47], що є одним з основних чинників їх неефективності у навчанні під час обробки різноманітних довготривалих залежностей у даних. Це пов'язано з проблемою затухання градієнтів [46], коли інформація, що передається через багато кроків часу, поступово втрачає свою важливість. Це відбувається через те, що градієнти, які використовуються для оновлення ваг мережі, зменшуються до дуже малих значень. Вирішення цієї проблеми стало основною мотивацією для розробки більш складних архітектур, таких як LSTM і GRU, які можуть краще зберігати і використовувати довготривалі залежності.

– ***LSTM Cells (Клітини LSTM).*** В LSTM-клітинах проблема затухання градієнтів шляхом введення механізмів забування f_t, запам'ятовування i_t та виходу o_t. Математично, вони визначаються такими функціями [47]:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big); \tag{2.2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i); \tag{2.3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_0); \tag{2.4}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C); \tag{2.5}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t; \tag{2.6}$$

$$h_t = o_t \odot tanh(C_t). \tag{2.7}$$

Тут $\odot$ – символ поелементного множення; $\sigma(x) = \frac{1}{1+e^{-x}}$ – сигмоїдна функція активації, що обмежує значення f_t між 0 і 1; f_t – вектор, що визначає, яку частину інформації з попереднього стану клітини C_{t-1} потрібно забути; W_f –

ваги для h_{t-1} (прихованого стану попереднього кроку) та x_t (поточного входу); b_f – зміщення (bias) для воріт забування; i_t – вектор, що визначає, яку частину нової інформації потрібно додати до стану клітини; $\tilde{C}_t$ – кандидат на новий стан клітини, обчислюється за допомогою гіперболічного тангенса tanh(x); W_i та W_C – ваги для прихованого стану h_{t-1} і поточного входу x_t; b_i та b_C – зміщення для відповідних механізмів; C_t – оновлений стан клітини, який є комбінацією старого стану C_{t-1}, скоригованого на ворота забування f_t, та нового кандидата на стан клітини $\tilde{C}_t$, зваженого входом i_t; o_t – вектор, що визначає, яку частину стану клітини C_t використати для формування виходу h_t; h_t – вихідний стан LSTM-клітини на поточному кроці часу; W_o – ваги для прихованого стану h_{t-1} і поточного входу x_t; b_0 – зміщення для воріт виходу.

Функція (2.2) є сигмоїдною функцією активації. Вона перетворює вхідне значення x_t у значення з діапазону від 0 до 1. Якщо значення функції f_t є близьким до 1, то обчислювальна модель LSTM зберігає більшу частину попередньої інформації з попереднього стану h_{t-1}. Якщо ж значення f_t є близьким до 0, то інформації з попереднього стану забувається. Таким чином, механізм забування (функція f_t) є одним із основних механізмів, завдяки якому в LSTM-клітинах реалізується вибіркове зберігання або забування інформації з попередніх станів.

Функція (2.3) описує механізм запам'ятовування нової інформації в LSTM-клітинах. Її реалізація полягає в множенні двовимірної вагової матриці W_i на вектор $[h_{t-1}, x_t]$, утворений конкатенацією (з'єднанням) вхідних даних x_t і прихованого стану h_{t-1}, додаванні вектора зміщення b_i та перетворенні результату $W_i \cdot [h_{t-1}, x_t] + b_i$ в значення з діапазону від 0 до 1 за допомогою сигмоїдної функції активації.

Лінійна трансформація компонентів вектора $[h_{t-1}, x_t]$ у компоненти вектора $W_i \cdot [h_{t-1}, x_t] + b_i$ визначає вплив інформації з попереднього прихованого стану h_{t-1} та вхідних даних x_t на результат розрахунку значень i_t. Чим більше значення i_t наближається до одиниці, тим більше цієї інформації буде включено в оновлений стан клітини C_t в LSTM. Таким чином, основним

призначенням механізму запам’ятовування є визначення та контроль кількості нової інформації, яку необхідно додати до стану клітини, щоб нейронна мережа вибірково зберігала її на кожному часовому кроці та залишалася гнучкою в управлінні потоком.

Функція (2.4) описує механізм виходу в LSTM. Вона призначена для визначення частки інформації з поточного стану комірки C_t, яка в LSTM використовується для формування поточного векторного прихованого стану h_t, шляхом реалізації формули (2.5) – (2.7).

Функції (2.5) – (2.7) значно ускладнюють архітектуру нейронної мережі. Архітектура LSTM є набагато складнішою за архітектури звичайних RNN. З одного боку, ця складність є виправданою, оскільки LSTM дають змогу вирішувати завдання, з якими простіші архітектури не можуть впоратися, а з іншого – вона додає значних обчислювальних витрат та істотно ускладнює налаштування LSTM-моделі, що не завжди є необхідним та доцільним для всіх застосувань. Завдяки механізмів забування, запам'ятовування та виходу LSTM-клітини мають значні переваги в обробці довгих часових послідовностей та довготривалих залежностей в послідовних даних порівняно зі звичайними RNN. Вони характеризуються високою ефективністю запам'ятовувати довгі часові послідовності та здатністю зберігати релевантну інформацію про них протягом тривалого часу. Однак, не зважаючи на зазначені переваги, використання LSTM-клітини у багатьох випадках є проблематичним, і ці проблеми пов’язані в основному з особливостями архітектури LSTM. Введення механізмів забування, запам'ятовування і виходу істотно збільшують обчислювальну складність нейронної мережі, що, в свою чергу, призводить до збільшення часу та обчислювальних ресурсів, необхідних для навчання моделі. Окрім цього, ускладнення архітектури нейронної мережі є також одним з основних чинників, які істотно ускладнюють її налаштування [48]. Адже, чим складніша архітектура мережі, тим більше параметрів вона містить, і тим більше зусиль потрібно для її налаштування, що істотно ускладнює процес оптимізації LSTM-моделі та підвищує обчислювальні витрати на навчання і використання мережі. Ці фактори

підвищують ризик перенавчання LSTM на невеликих наборах даних і ускладнюють інтеграцію моделі в системи з обмеженими ресурсами. Тому у випадках коли важливими є обмежені обчислювальні ресурси, швидкість навчання, робота з невеликими наборами даних, або коли завдання не потребує складного зберігання довготривалих залежностей доцільним є використання не LSTM мережі, а її альтернативи GRU.

- Gated Recurrent Unit (GRU). GRU є спрощеною версією LSTM. Ця рекурентна нейронна мережа є ефективним засобом обробки довготривалих залежностей з меншою обчислювальною складністю. На відміну від LSTM мережа GRU має не три, а два основних механізми управління потоком інформації: механізм оновлення z_t та механізм скидання r_t. Математично вони описуються функціями [8]:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z); \quad (2.8)$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r); \quad (2.9)$$

$$\tilde{h}_t = tanh(W_h \cdot [r_t \odot h_{t-1}, x_t]); \quad (2.10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (2.11)$$

Функція (2.8) визначає, яку частку інформації з попереднього прихованого стану h_{t-1} потрібно "забути" або "скинути" перед визначенням нового стану, а функція (2.9) – частку інформації попереднього прихованого стану h_{t-1}, яка буде збережена в новому прихованому стані h_t.

З аналізу функцій (2.8)-(2.11) випливає, в GRU на відміну від LSTM немає окремого стану комірки. Всі операції виконуються на рівні прихованого стану h_t. Окрім цього з огляду на (2.8)-(2.11) GRU використовує менше параметрів, а це означає, що ця мережа порівняно з LSTM є швидшою у навчанні та менш вимогливою до обчислювальних ресурсів.

Вибір між GRU і LSTM визначається конкретними вимогами до моделі, наявними обчислювальними ресурсами та особливостями вхідних даних. У більш загальних випадках GRU є швидшою у навчанні та вимагає меншу кількість ресурсів, але для задач аналізу відео, які вимагають збереження інформації на довгих часових інтервалах, LSTM завдяки окремому механізму

керування довготривалою пам’яттю надає кращу якість моделювання залежностей. Таким чином, хоча GRU часто використовується як легша альтернатива, її застосування може бути обмеженим у складніших задачах У деяких випадках GRU може бути менш універсальною в порівнянні з LSTM чи іншими типами RNN мереж. На теперішній час не існує універсальної нейронної мережі, яка б забезпечувала максимальну продуктивність у всіх можливих випадках. Кожна архітектура має свої переваги та недоліки, і оптимальний вибір завжди залежить від контексту завдання [6,8]. Тому важливим є аналіз інструментів за допомогою яких можна удосконалювати існуючі нейронні мережі з метою їх адаптації до контексту конкретної задачі. Таким інструментами є механізми уваги [49].

– ***Attention Mechanisms (Механізми Уваги)***. Механізми уваги використовуються для розрахунку коефіцієнтів уваги (ваг) для прихованих станів або вхідних даних нейронної мережі (наприклад, GRU) з метою формування вихідних даних в поточному завданні мережі. Значення коефіцієнтів уваги $\alpha_{t,i}$ розраховуються за формулами [8,49]:

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{k=1}^{T} exp(e_{t,k})}; \tag{2.12}$$

$$\mathrm{e}_{t,i} = v^T tanh(W_h h_{t-1} + W_x x_i). \tag{2.13}$$

Тут: $\mathrm{e}_{t,i}$ – оцінка уваги для i-го кадру; x_i – i-та компонента вхідного вектора послідовності; W_h, W_x та v^T – параметри, які налаштовуються у процесі навчання моделі.

Вихід моделі визначається зваженою сумою контексту:

$$\mathrm{c}_t = \sum_{i=1}^{T} \alpha_{t,i} x_i \tag{2.14}$$

Завдяки механізмам уваги нейронні мережі GRU, LSTM, тощо RNN набувають здатності вибірково зосереджуватися на найбільш важливих частинах вхідних даних, що покращує її продуктивність та точність у виконанні поточних завдань. Вони можуть бути використані для удосконалення відомих та побудови нових моделей пошуку ключових кадрів у відео, забезпечуючи точніше

визначення кадрів, які містять найважливішу інформацію або найбільш значущі зміни в контексті всього відео.

2.1.2. Графові нейронні мережі

У сучасних системах пошуку відео за його довільним фрагментом все частіше застосовуються графові нейронні мережі (Graph Neural Networks, GNNs). Вони є надзвичайно ефективними для вирішення задач моделювання взаємодії об'єктів відео, особливо у випадках, коли відео можна представити у вигляді графа, вершинами якого є кадри, а ребрами – зв’язки або просторово-часові відношення між ними.

Алгоритми GNNs оновлюють стан кожного вузла графа відео на основі інформації від сусідніх вузлів і тим самим дають змогу моделювати складні залежності та взаємодії в них об’єктів відео. Основним алгоритмом в цій нейронній мережі є графові згортки. Вони розширюють поняття згорток (convolutions) з традиційних CNN, які призначені в основному для обробки даних з регулярними структурами, на нерегулярні графові структури [50]. Для кожного вузла v його новий стан h'_v визначається функцією станів його сусідів $N(v)$:

$$h'_v = \sigma\left(\sum_{uN(v)} \frac{1}{c_{vu}} W h_u\right), \tag{2.15}$$

де σ – нелінійна функція активації (наприклад, ReLU); W – матриця ваг; c_{vu} – нормувальний коефіцієнт (наприклад, ступінь вузла v).

Окрім графових згорток, не менш важливим компонентом GNNs є графові агрегатори (Graph aggregators). Їх основним призначенням є об’єднання інформації сусідів вузла. Різні стратегії агрегації включають середні (mean), максимум (max) або суму (sum):

$$h'_v = AGG(\{h_u | u \in N(v)\}), \tag{2.16}$$

де AGG – функція агрегації.

Однією з основних проблем GNNs є проблема масштабованості, сутність якої полягає в істотному зменшенні ефективності роботи GNNs у випадку обробки великих графів. Вона зумовлена тим, що обробка графів з великою кількістю вузлів та ребер за допомогою алгоритмів згортки GNNs

потребує значних обчислювальних ресурсів, особливо у випадку роботи з відеодатасетами. Для наочного обґрунтування цього твердження покажемо, що в процесі обробки відеодатасетів GNN виконується значна кількість обчислювальних операцій. Адже чим більше операцій потрібно для виконання завдання, тим більше обчислювальних ресурсів (часу, пам'яті, енергії) необхідно для їх реалізації [51]. Для цього покажемо, що для опрацювання відеодатасету, де кадри відео виступають у ролі вузлів графа, а ребра утворюються між сусідніми кадрами. У такому графі з 10^3 відео по 10^4 кадрів маємо $\approx 10^7$ вузлів і $\approx 10^7$ ребер. Оскільки GNN обчислює нові представлення для кожного вузла шляхом агрегації інформації з його безпосередніх сусідів, складність одного шару становить $O(|E|)$, а для глибокої мережі з L шарами – $O(L \cdot |E|)$. Тобто загальна кількість операцій дійсно може сягати сотень мільйонів для великих датасетів. Це робить актуальними методи оптимізації, зокрема графового укрупнення (graph coarsening), що дозволяють зменшити розмір графа та знизити обчислювальне навантаження [50]. Вище наведений приклад також наочно демонструє, що масштабованість стає критичною проблемою під час обробки великих графів ще і тому, що обчислення, необхідні для визначення релевантних зв'язків між кадрами або об'єктами, можуть перевищити доступні ресурси. Ці зв'язки суттєво впливають на ефективність навчання та використання GNN. Їх врахування є можливим за умови складного попереднього оброблення даних та проведення додаткових обчислень. Тому для зменшення обчислювальних затрат та підвищення точності GNN моделей автори робіт [50,51] пропонують алгоритми, які у процесі самопідсилювального навчання та попередньої обробки даних автоматично навчаються структурам графів.

Графові нейронні мережі є перспективним напрямком для обробки відеоданих та пошуку відео за його довільним фрагментом. Їх використання дає змогу ефективно моделювати складні взаємодії між об'єктами. Проте, проблема масштабованості, а також їх чутливість до шуму та складність побудови графів можуть ускладнити застосування GNN у реальних умовах. Для вирішення цих викликів необхідні додаткові дослідження та розробка методів оптимізації, що

забезпечать більшу стійкість до шуму, спрощення побудови графів та підвищення ефективності роботи з великими наборами даних.

2.1.3. Глибокі конволюційні нейронні мережі

Одним із ефективних засобів для аналізу відео є глибокі конволюційні нейронні мережі (DCNNs). Вони ідеально підходять для вирішення завдань автоматичного пошуку відео за довільним фрагментом [44]. Завдяки ієрархічному навчанню, вони є універсальними інструментами обробки різноманітних типів даних, зокрема зображеннь, відео та аудіо. Окрім цього, DCNNs також легко масштабується, що має надзвичайно важливе значення для синтезу (побудови) глибоких моделей для виявлення складних закономірностей. Тому з огляду на тему та завдання даної дисертаційної роботи розглянемо більш детально архітектуру DCNN та алгоритми, які покладені в її основі.

Архітектура нейронної мережі DCNN складається з кількох типових шарів, кожен з яких виконує певну функцію. Основними компонентами архітектури DCNN є: конволюційні шари, шари підвибірки (pooling), повнозв'язні шари (fully connected) та рекурентні шари.

Конволюційні шари. Конволюційні шари призначені для автоматичного аналізу зображень (відеокадрів) та виділення з них найбільш важливих просторових ознак (країв, текстур, форм та інших візуальних характеристик) для подальшого формування карти ознак. Формально, вихід $Y(i,j)$ конволюційного шару (карта ознак) для вхідного зображення X з фільтром K розраховується за формулою згортки [44]:

$$Y(i,j) = \sum_m \sum_n X(i+m, j+n) \cdot K(m,n), \qquad (2.17)$$

де (i,j) – координати вхідного елемента, (m,n) – координати фільтра.

Pooling шари. Pooling шари зменшують розмірність вхідних даних (висоти та ширини карти ознак), зберігаючи при цьому найбільш важливі ознаки зображення або відеокадру. Це досягається шляхом обчислення максимуму або середнього значення у визначеному вікні. Наприклад для max-pooling:

$$Y(i,j) = max\{X(m,n) | (m,n) \in window(i,j)\}. \qquad (2.18)$$

Fully connected шари. Fully connected шари використовуються для остаточної класифікації кадрів відео або визначення регресійних залежностей для прогнозування певних безперервних числових значень на основі ознак, виділених з відео шляхом об’єднання виділених попередніми шарами ознак у вектор виходу. У випадку класифікації, цей вектор може містити ймовірності належності кадру до певного класу, наприклад типу об'єкта чи сцени. У випадку вирішення регресійних задач, fully connected шари використовуються для розрахунку числового значення, яке може відображати, наприклад, оцінку руху або прогноз певної величини на основі відео. Fully connected шари з’єднують всі нейрони попереднього шару з кожним нейроном поточного шару:

$$y = \sigma(Wx + b), \tag{2.19}$$

де W – матриця ваг; x – вхідний вектор; b – зміщення.

Рекурентні шари. Рекурентні шари, такі як LSTM або GRU, часто використовуються для обробки часових залежностей у відео. Їх використання дає змогу враховувати інформацію з попередніх кадрів під час обробки поточного кадру.У випадку LSTM, вони визначаються функціями (2.2) – (2.7).

Завдяки наявності шарів підвибірки та конволюційних, повнозв'язних і рекурентних шарів, нейронні мережі DCNN сьогодні успішно використовуються для автоматичної обробки та аналізу не лише просторових, але і часових ознак відео. Завдяки їм DCNN моделі дають змогу з високою точністю виявити складні взаємозв'язки між об'єктами та їх рухом у кадрах. Однак, попри ці переваги вони не є ідеальними: їх практична реалізація в багатьох випадках потребує значних обчислювальних ресурсів та великих наборів даних для досягнення високої точності. Окрім цього вони є складними в налаштуванні та навчанні [58]. Але, зазначимо, що деякі недоліки можна зменшити шляхом адаптації архітектури DCNNs до завдань та вимог поставленої у дисертаційній роботі задачі. Зокрема, обчислювальні витрати DCNNs без втрати точності можна істотно зменшити за допомогою легковагових фільтрів та структурних інновацій архітектур MobileNet та EfficientNet [44].

Структурною інновацією в архітектурі MobileNet є глибокі роздільні згортки, які реалізують обчислювальний процес у два етапи: depthwise convolution (глибинна згортка) і pointwise convolution (точкова згортка). Глибинна згортка призначена для обробки кожного каналу вхідного зображення окремо, а точкова згортка – для комбінування результатів з різних каналів зображення. Глибинна згортка виконує згортання лише по просторових вимірах для кожного каналу вхідного зображення, тоді як точкова згортка виконує згортання з використанням фільтрів 1x1 для об'єднання інформації з усіх каналів. Саме завдяки використанню глибинної згортки, точніше завдяки розділенню процесу обробки зображень на два етапи зменшуються обчислювальні витрати та зберігається точність DCNNs моделі [1,47].

Доцільність використання архітектури EfficientNet для адаптації існуючих DCNNs до задачі пошуку відео за довільним фрагментом обґрунтовується наявністю методу збалансованого масштабування ширини, глибини та роздільної здатності параметрів моделі EfficientNet. Завдяки цьому методу EfficientNet є ефективним засобом підвищення точності обробки зображень та кадрів відео за допомогою глибоких згорткових нейронних мереж (DCNNs).

Для удосконалення існуючих глибоких згорткових нейронних мереж DCNNs і їх подальшого застосування сьогодні часто використовують процедуру квантування моделей та апаратні прискорювачі, зокрема TPU та FPGA. Їх використання дає змогу складні моделі нейронних мереж адаптувати до умов з обмеженими обчислювальними ресурсами і забезпечити їх ефективну роботу на пристроях з низькою потужністю. Адже, квантування зменшує розрядність ваг та активацій, що в кінцевому підсумку призводить до зниження обсягу оперативної пам'яті (RAM) та пам'яті зберігання (наприклад, флеш-пам'яті), необхідної для зберігання моделі і її проміжних даних під час обчислень.

Сьогодні для модернізації існуючих DCNNs відповідно до умов задачі пошуку відео за довільним фрагментом, окрім процедури квантування, доцільним є використання також попередньо навчених моделей (pre-trained models) та методів трансферного навчання. Їх застосування дає змогу значно

зменшити час тренування моделі, обсяг необхідних даних для навчання, а також підвищити точність і ефективність моделі за рахунок використання вже наявних знань, отриманих на великих датасетах, зокрема на ImageNet [15].

Загалом DCNNs є оптимальним вибором для вирішення задачі пошуку відео за його довільним фрагментом. Він обгрунтовується:

1. Здатністю DCNNs виявляти важливі візуальні патерни та структури відео шляхом реалізації процедури витягування складних просторових ознак за допомогою конволюційних шарів. У поєднанні з рекурентними компонентами, зокрема з LSTM або GRU, DCNNs є відмінним вибором для моделювання часових залежностей між кадрами.

2. Обчислювальною ефективністю та масштабованістю DCNNs. Адже, сучасні архітектури MobileNet та EfficientNet глибоких згорткових нейронних мереж є оптимізованими для роботи в умовах обмежених ресурсів. Завдяки цим оптимізаціям істотно знижується обсяг обчислювальних витрат на виконання необхідних операцій та підвищується масштабованість мереж DCNNs.

3. Стійкістю DCNNs до шуму у вхідних даних та інтерпретованістю результатів їх роботи. Глибокі нейронні мережі (DCNNs) характеризуються ефективним витягуванням корисних ознак з даних, навіть коли ці дані містять шум або нерелевантну інформацію. Їх стійкість до різного роду спотворень у вхідних даних значною мірою визначаються операціями фільтрації та агрегації інформації, що здійснюються за допомогою конволюційних та pooling шарів. Ці шари допомагають виділяти суттєві патерни, ігноруючи або пригнічуючи шум, і тому DCNNs мережі є надійними для обробки складних і "шумних" даних. Інтерпретованість же результатів роботи DCNNs залишається складним завданням, вирішення якого потребує додаткових методів аналізу. Однак сьогодні існують архітектурні компоненти та механізми, які допомагають суттєво покращити здатність мережі розпізнавати важливі ознаки та підвищити інтерпретованість результатів її роботи. Зокрема, механізми уваги (attention mechanisms), шляхом інтеграції яких в нейронну мережу DCNNs можна істотно підвищити точність результатів обробки відео та їх інтерпретованість [3].

4. Підтримкою та інфраструктурою. DCNNs користуються широкою підтримкою в науковому співтоваристві та мають багатий набір програмних інструментів для розробки та оптимізації моделей. До цього набору належать фреймворки TensorFlow та PyTorch, які надають розробникам, дослідникам та інженерам доступ до великих попередньо навчених моделей, зокрема до моделей навчених на ImageNet. Ці та інші засоби та можливості дають змогу дослідникам та розробникам швидко адаптувати DCNNs до умов вирішення задачі пошуку відео за фрагментами.

Отже, глибокі конволюційні нейронні мережі (DCNNs) є потужним засобом обробки візуальних даних. DCNNs, разом з рекурентними нейронними мережами (RNNs), довготривалою короткочасною пам'яттю (LSTM) та графовими нейронними мережами (GNNs), можуть слугувати основою для синтезу моделі, оптимально адаптованої до умов задачі пошуку відео за його довільним фрагментом. DCNNs у поєднанні з RNNs, LSTM та GNNs дають змогу враховувати як просторові, так і часові характеристики відео, а також взаємозв'язки між різними елементами контенту [44].

2.2. Розроблення обчислювальної моделі для детекції сцен у відеопотоках

Одним із основних завдань задачі пошуку відео за його фрагментами є побудова згорткової нейронної моделі (DCNN) індексації контенту для швидкого пошуку потрібних фрагментів у відео на основі запитів або певних характеристик, зокрема сцен, об’єктів, дій тощо. Для його успішного вирішення у даній дисертаційній роботі розроблено архітектуру шару детекції сцен, оптимально адаптовану до поставленого завдання для її подальшої інтеграції у побудовану DCNN мережу. Необхідність та доцільність цієї розробки обґрунтовується детальним аналізом існуючих підходів до детекції сцен у відео, а також потребою в підвищенні точності та ефективності процесу індексації відеоконтенту. У процесі обґрунтування було враховано сучасні математичні методи, що забезпечують надійність і продуктивність розробленої моделі. Аналіз

цих методів та особливості їх практичного застосування викладено у першому розділі даної роботи.

В основі розробленої архітектури шару детекції сцен у відео DCNN мережі покладено метод виявлення меж кадрів (Shot Boundary Detection - SBD), який реалізується через аналіз гістограм кольорів у просторі HSV. Вибір простору HSV для аналізу обумовлений його здатністю окремо обробляти інформацію про кольори та яскравість, що є одним із основних чинників зменшення впливу шумів (випадкових змін в колірних компонентах пікселів або в умовах освітлення) на точність виявлення меж між сценами [7,8].

Для побудови шару детекції сцен у відео DCNN мережі використано такі математичні методи аналізу кадрів, а саме:

1. Метод різниці гістограм. Цей метод обраний через його здатність аналізувати зміни у розподілі кольорів між суміжними кадрами. Його практична реалізація полягає в обчисленні гістограм кадрів у HSV-просторі та подальшому розрахунку різниць між гістограмами сусідніх кадрів. Для кількісної оцінки відмінностей між кадрами використовуються Хі-квадрат метрики (χ^2) [16].
2. Метод абсолютних різниць кадрів. Вибір цього методу обґрунтовується його особливостями щодо ефективного виявлення різких змін у відео, які досить часто співпадають зі змінами сцени. Його практична реалізація полягає у розрахунку абсолютної різниці інтенсивностей пікселів між двома послідовними кадрами та подальшому порівняння значення цієї величини з певним порогом детекції: чим більше значення цієї різниці перевищує певний поріг детекції, тим більш різким є перехід між сценами [4].

Виявлення переходів між сценами у відео реалізується алгоритмами шару детекції сцен поетапно. На початковому етапі здійснюється попередня обробка кадрів шляхом їхнього перетворення в колірний простір HSV, результати якої слугують вхідними даними для алгоритмів і процедур наступних етапів вирішення поставленої задачі, зокрема для розрахунку гістограми кольорів та інтенсивності для кожного кадру [16].

Процес перетворення даних з колірного простору RGB (Red, Green, Blue) у дані колірного простору HSV (Hue, Saturation, Value) відбувається за такою схемою: спочатку значення компонентів червоного (R), зеленого (G) та синього (B) нормалізуються до діапазону від 0 до 1. Потім обчислюються максимальне та мінімальне значення серед цих трьох компонентів, і на основі отриманих результатів визначається різниця між найяскравішим та найтемнішим кольором у зображенні. Далі, на основі цієї різниці, розраховуються значення тону, насиченості та яскравості. Тон (Hue) визначається як кут у колірному колі, що відповідає тій компоненті (червоній, зеленій або синій), яка має найбільше значення в нормалізованому векторі RGB для конкретного пікселя. Таким чином, тон відображає відтінок, який найбільше впливає на колір цього пікселя. Значення насиченості зображення визначається відношенням різниці між максимальним і мінімальним значеннями компонентів до максимального значення, а значення яскравості дорівнює максимальному значенню серед компонентів RGB. Завдяки цьому перетворенню, аналіз кольорових характеристик зображення стає більш інтуїтивним та стійким щодо змін освітлення [31].

На наступному етапі обробки в шарі DCNN для кожної пари послідовних кадрів обчислюються метрики, а саме: різниця гістограм за метрикою χ^2 та сума абсолютних різниць пікселів. Пари кадрів формуються шляхом послідовного вибору кожного кадру та наступного за ним у часовій послідовності відео. Таким чином, для кожного кадру відео, окрім останнього, знаходиться пара з наступним кадром, що забезпечує покриття всього відео для аналізу. Ці метрики слугують критеріями для виявлення змін між кадрами, що дозволяє визначати переходи між сценами [44,52].

У запропонованій архітектурі шару детекції сцен переходи між сценами у відео ідентифікуються засобами аналізу збільшення значень метрик, що вказує на зміну сцени: чим більшою є різниця абсолютних значень метрик між наступним і попереднім кадрами відео, тим більш ймовірною є зміна сцени. Порогові значення для цих метрик встановлюються на основі емпіричних даних,

отриманих у процесі тренування моделі детекції сцен у відео, або шляхом адаптації цієї моделі до конкретних умов і характеристик відеоконтенту (частоти кадрів та динаміки сцен), з якими вона працює. Для виявлення переходів між сценами у відео модель аналізує декілька послідовних сцен і за результатами аналізу оцінює динаміку зміни метрик, а саме динаміку зміни різниць гістограм або тенденцію зміни абсолютних різниць пікселів. Залежно від характеру зміни цих метрик перехід від сцени до сцени відео може виявитися або поступовим, або різким. В обох випадках індикатором зміни сцен відео є адаптивний поріг метрики, перевищення якого вказує на наявність переходу між сценами.

Для наочного відображення результатів тестування запропонованої моделі детекції сцен було проведено експерименти на наборі відеоданих UCF-101 Action Recognition, де штучно створювалися різкі й плавні переходи між послідовними кадрами. На рис. 2.1 графік відображає покадрову різницю гістограм (синя лінія) у просторі HSV та адаптивний поріг (червона пунктирна). Вертикальними лініями позначено п’ять основних переходів: фіолетові відповідають різким (Hard) змінам, коли різниця суттєво перевищує поріг, а зелені - помірним (Soft) змінам. Це дає змогу моделі виявляти як різкі, так і поступові переходи між кадрами.

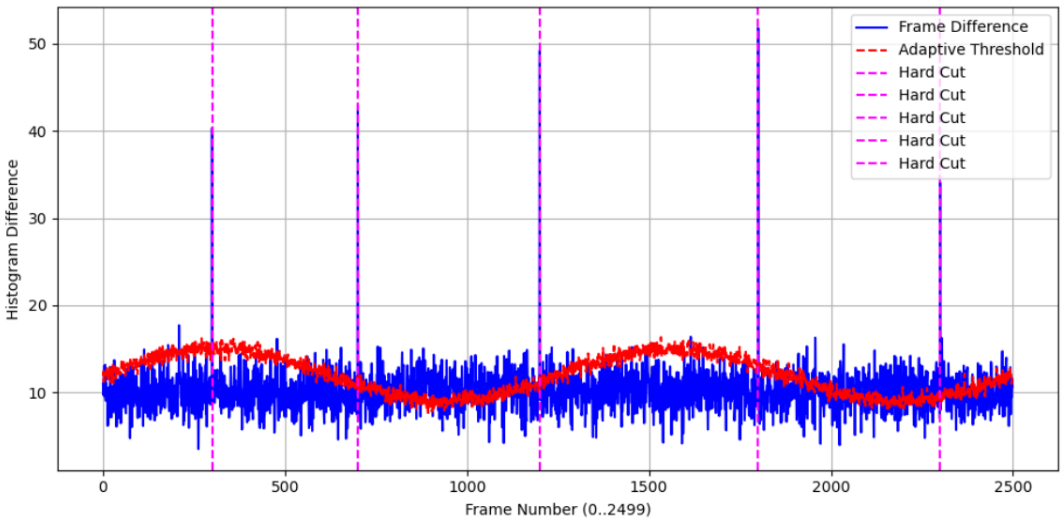

Рисунок 2.1. Залежність різниці гістограм двох сусідніх кадрів відеоданих UCF-101 Action Recognition у колірному просторі HSV

Аналіз цього цього рисунку дає змогу виявити та визначити поступові та різкі переходи між сценами, що свідчить про високу чутливість розробленої

моделі до різких змін у відеопотоці. Наочним підтвердженням цього висновку є відображені на рис.2.1 різні сцени, кожна з яких ілюструє конкретний тип переходу.

На рис.2.2 наведено результати аналізу для реального відеофайлу тривалістю 90 секунд, без використання механізму уваги. На відміну від штучно зібраного прикладу, тут спостерігається більше коливань, зумовлених природними змінами освітлення й руху об’єктів у кадрі. При цьому модель усе ще розрізняє різкі переходи (червоні лінії) та плавні (зелені лінії), але частка «помилкових» Soft Cuts може зростати, оскільки дрібні локальні зміни не фільтруються засобами уваги. Таким чином, інтеграція механізму уваги у подальших експериментах покращує стійкість до шуму і небажаних коливань, підвищуючи точність детекції.

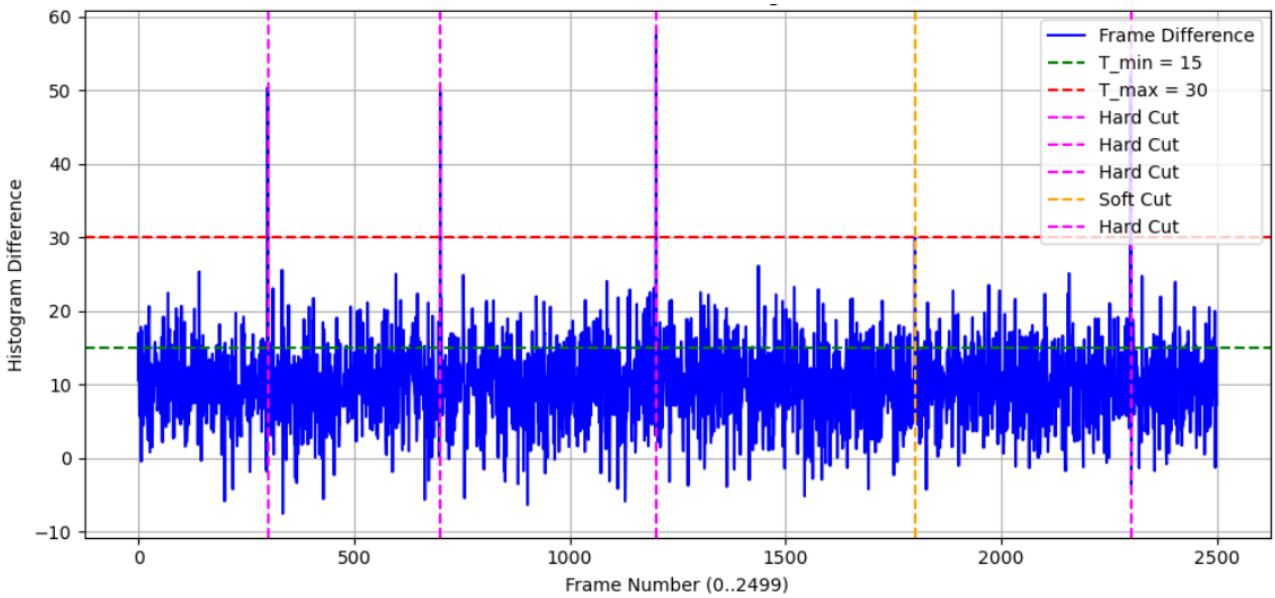

Рисунок 2.2. Динаміка різниці гістограм між сусідніми кадрами (реальний відеофайл тривалістю 90 секунд у просторі HSV). Показано два фіксовані пороги (T_min і T_max) та моменти, коли різниця перевищує кожен із них

У результаті впровадження механізму уваги зміниться не лише архітектура шару детекції сцен, але й спосіб обробки та аналізу даних. Особливості обробки кадрів відео у такому шарі визначаються особливостями взаємодії механізму уваги з методами різниці гістограм та абсолютних різниць пікселів. Зокрема, у процесі взаємодії з методом різниці гістограм механізм уваги виділяє найбільш важливі кадри або їх частини для подальшого виявлення поступових переходів

(Soft Cuts) між сценами. Завдяки цій взаємодії створена модель зможе зосереджуватися на найбільш значущих кадрах відео, навіть коли зміни у кольоровій палітрі є менш вираженими і розтягнутими в часі. У випадку взаємодії з методом абсолютних різниць пікселів механізм уваги спрямовує модель детекції сцен на області кадрів, де відбуваються найсильніші зміни в інтенсивності пікселів. Ця взаємодія використовується у випадках коли необхідно з високою точністю виявити різкі переходи (Hard Cuts) у зображеннях з різкими та значними змінами у кольоровій палітрі. На практичному рівні інтегрований в побудований DCNN-шар механізм уваги виконує функцію коригування важливості кадрів та їх частин за даними розрахунку значень метрик, визначених попередніми шарами. Завдяки цьому запропонована модель адаптує свої рішення до локальних особливостей відеопотоку, що покращує загальну ефективність системи [22,32].

На виході нашої системи шар детекції сцен генерує структуру даних, що містить детальну інформацію про кожний кадр відео та метадані про переходи між сценами. Ця структура описується масивом об'єктів, кожний з яких визначається індексом кадру (FrameIndex), часовою міткою (Timestamp), типом переходу (TransitionType), різницею пікселів (PixelDifference), різницею гістограм (HistogramDifference) та індикатором перевищення порогу (ThresholdExceeded). Такий формат є оптимальним щодо ідентифікації кадрів, класифікації переходів між сценами та адаптації моделі до контексту відео [53].

Таким чином, описана модель DCNN, важливим компонентом якої є розроблений у даній дисертаційній роботі шар детекції сцен, є інноваційним рішенням для автоматизованої обробки відеопотоку. Її практичне застосування забезпечує точну та ефективну індексацію відеоконтенту для проведення подальшого аналізу переходів між сценами.

На архітектурному рівні процес обробки в розробленій глибокій згортковій нейронній мережі (DCNN) для детекції сцен у відео є багатоступеневим (рис.2.3). Для його реалізації використано різні методи аналізу кадрів. Розмір вхідних зображень 224x224 пікселів був обраний не випадково [20]. Цей розмір

забезпечує достатню кількість деталей і текстур для ефективного розпізнавання та класифікації об'єктів, а також хороший баланс між деталізацією зображення і обчислювальними витратами. Без такого балансу навчання та прогнозування моделі в рамках багатьох задач глибокого навчання можуть бути неефективними. Окрім цього, зображення такого розміру не є надто великими, тому їх обробка є більш ефективною при збереженні навантаження на обчислювальні ресурси на оптимальному рівні порівняно з великими зображеннями.

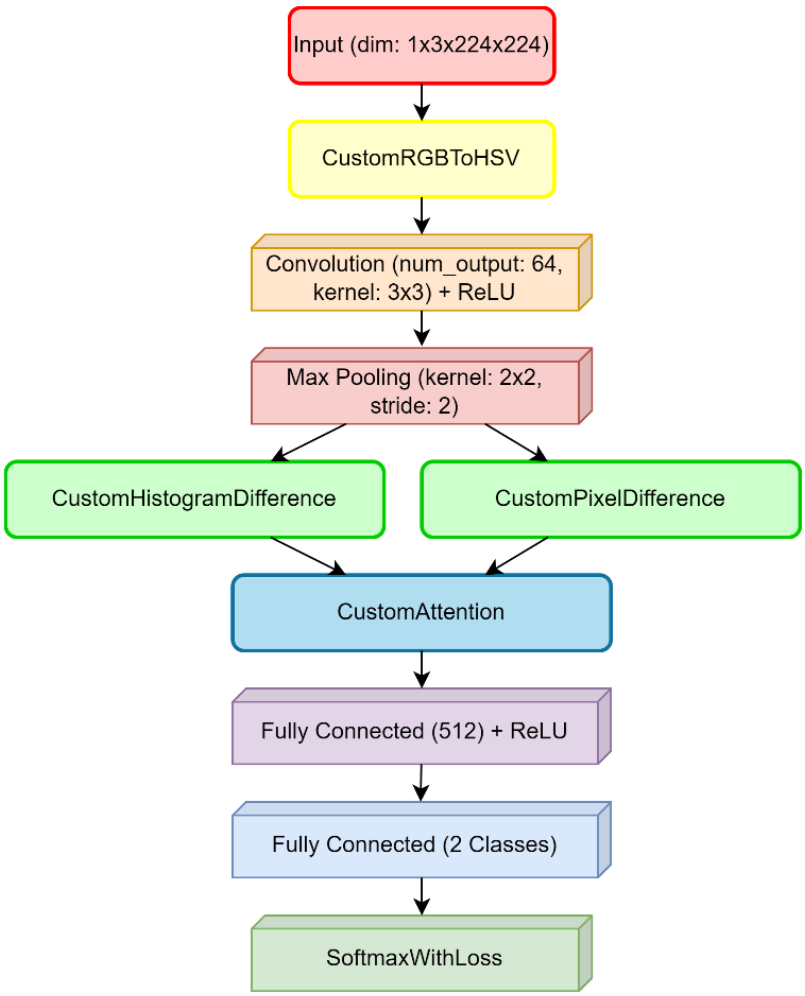

Рисунок 2.3. Структурно-функціональна схема шару детекції сцен у відео

Процес обробки починається з вхідного шару. Відеодані у форматі RGB надходять у мережу у вигляді кадрів розміром 224x224 пікселів. Після цього вони проходять через шар CustomRGBToHSV, який перетворює зображення з формату RGB в формат HSV. Доцільність (потреба) цього перетворення обґрунтовується тим, що виділення і відокремлення колірних компонентів, а

також детекція переходів між сценами є більш ефективними, коли відеодані представлені у форматі HSV [23,52]. Колірна інформація у цьому форматі є більш чіткою і інтуїтивно зрозумілою для обробки відеоданих, порівняно з форматом RGB.

На наступному етапі кадри відео, конвертовані у формат HSV, обробляються за допомогою кількох конволюційних шарів. У кожному шарі з використанням фільтрів розміром 3x3 і нелінійної активаційної функції ReLU виконується операція згортки. В результаті цієї операції отримується інформація про краї, текстури та інші важливі патерни кадрів, які можуть служити індикаторами переходів між сценами. Далі ця інформація обробляється за допомогою процедури Max Pooling, основним призначенням якої є зменшення розмірності вхідних даних та зниження чутливості до дрібних змін при збереженні найбільш значущих особливостей зображення [54].

Спеціалізовані шари CustomHistogramDifference та CustomPixelDifference шару детекції сцен у відео, функціонально-структурна схема якого наведена на рис.2.3, призначені для розрахунку значень різниць між гістограмами компонентів кольорових кадрів та різниць між інтенсивностями пікселів у цих кадрах відповідно [8, 32]. Перший шар призначений для виявлення зміни у розподілі кольорів між послідовними кадрами, а другий – для виявлення зміни в яскравості між цими кадрами. Завдяки цим шарам шар детекції сцен у відео може виявляти не лише поступові, але і різкі переходи між сценами. Зазначимо також, що окрім шарів CustomHistogramDifference та CustomPixelDifference в архітектуру запропонованої у даній роботі моделі індексації відео інтегровано механізм уваги (CustomAttention). Він впроваджений у модель з метою забезпечення додаткової можливості для її фокусування на найбільш важливих частинах кадру відео та підвищення точності і ефективності аналізу перехідних сцен у ньому [11].

Побудована у даній дисертаційній роботі модель індексації відео є результатом своєрідного поєднання конволюційних шарів, обчислювальних метрик (гістограмних та піксельних відмінностей) та механізму уваги. Вона є

ефективним інструментом для ідентифікації різних типів переходів між сценами відео, незалежно від того, чи це різкі зміни (Hard Cuts), чи поступові переходи (Soft Cuts). Конволюційні шари виділяють ключові особливості кадрів, обчислювальні метрики аналізують різниці між кадрами, а механізм уваги фокусується на важливих частинах кадру для підвищення точності обробки. Фінальні повнозв'язні шари з активацією Softmax дозволяють об'єднати всі ці особливості та класифікувати кадри для точного визначення меж сцен і покращення точності обробки відеопотоку [1,47].

2.3. Розроблення обчислювальної моделі для екстракції характеристик сцен у відео

Загальновідомо, що результати моделі детекції сцен є лише початковим етапом обробки даних для індексації відео. Після детекції сцен часто виникають ситуації, коли одна сцена розбивається на кілька дрібних сегментів через незначні або некоректні зміни в кадрах, а виявлені межі між сценами не завжди відповідають фактичними переходам між сценами. Окрім цього спроектована у даній роботі модель детекції сцен не виділяє основних характеристик сцени її текстури, об'єктів, рухів тощо. Це означає, що детекція сцен є необхідною, але не достатньою умовою успішного вирішення задачі індексації відео. Одним із шляхів її розв’язання є додаткова обробка результатів детекції сцен, а саме: виділення та аналіз основних характеристик кожної окремої сцени; відфільтровування помилкових детекцій і збереження меж між сценами, які дійсно відповідають фактичним переходам між сценами; та об’єднання дрібних сегментів кадру відео в одну цілісну сцену [6].

Визначальними для ідентифікації та поглибленого розуміння структури сцени і взаємозв’язків між подіями у ній, а також для формування моделі екстракції характеристик сцен у відео є процедури виділення та детального аналізу тимчасових характеристик кожної окремої сцени. Тому, зважаючи на важливість аналізу тимчасових характеристик сцен у відео у даній дисертаційній роботі розроблено декілька спеціалізованих шарів обробки результатів детекції

сцен, функціональним призначенням яких є виконання певних функцій, які у сукупності забезпечують гарантовану ефективність процедури виділення та аналізу ключових характеристик сцени. Одним із таких шарів є шар, основним призначенням якого є розрахунок векторів тривалостей сцен. Вхідними даними для цього шару є частота кадрів у відео та результати обчислень кількості кадрів у кожній сцені, які є вихідними даними практичної реалізації запропонованої у даній роботі моделі детекції сцен. Цей шар є фундаментальним, оскільки його вихідні дані – інформація про тривалість кожної сцени – є вхідними даними для інших шарів подальшої обробки результатів детекції [12]. Він є фундаментальним оскільки тривалість сцени є одним з основних показників оцінки відносної значимості сцени у структурі відео: короткі сцени можуть свідчити про швидкий темп подій, а довші – про важливість або інтенсивність розглянутих подій.

Окрім тривалості сцен важливою характеристикою для індексації відео є послідовність подій всередині сцени. Ця часова характеристка визначає, в якому порядку і як розгортаються дії та взаємодії між об'єктами чи персонажами в межах сцени. Вона допомагає зрозуміти зміст і логіку розвитку подій, що впливає на загальне сприйняття сцени, її наратив і контекст. Тому визначення послідовності подій всередині сцени є основним завданням систем індексації та пошуку відео (CBVIR). Для його вирішення у даній дисертаційній роботі синтезовано обчислювальну модель екстракції характеристик сцен у відео. В основі її побудови покладено багаторівневу архітектуру нейронної мережі SlowFast (рис.2.4). Впровадження цієї архітектури у даному випадку обгрунтовується тим, що у SlowFast обробка результатів детекції сцен у відео здійснюється не одним, а двома паралельними потоками: повільним та швидкісним [13]. Повільний потік обробляє довгострокові, повільно змінювані ознаки, а швидкий – фокусується на швидких (Fast pathway) та короткотривалих подіях (Slow pathway). Завдяки цим потокам SlowFast на відміну від інших нейронних мереж значно краще захоплює як просторові, так і часові залежності, що є одним із найбільш важливих чинників підвищення точності аналізу

складних послідовностей даних. Багаторівнева архітектура SlowFast є досить гнучкою, вона легко адаптується до різних типів відео шляхом налаштування окремих потоків для оптимальної роботи [55].

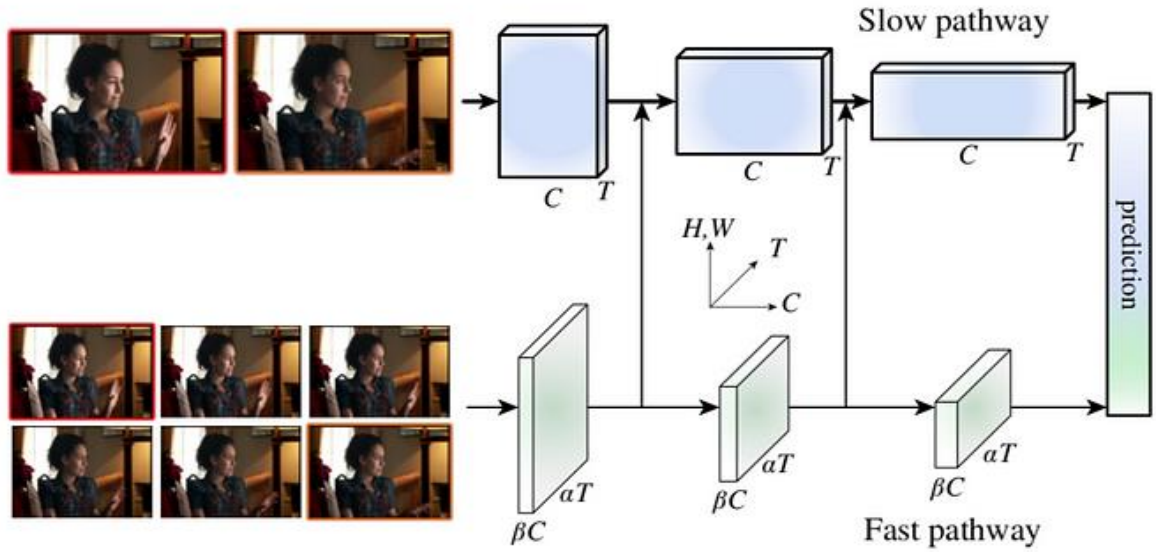

Рисунок 2.4. Архітектура нейронної SlowFast

Із зазначеного вище, випливає, що нейронна мережа SlowFast призначена в основному для обробки часово-просторових залежностей у відео [48], зокрема для аналізу повільних і швидких змін. Це означає, що її використання для вирішення інших специфічних даних відео не завжди може виявитися оптимальним, особливо у випадках обробки часових маркерів, послідовності подій та характеристик переходів між сценами. Тому для побудови повноцінної моделі екстракції характеристик сцен у відео окрім мережі SlowFast у цій дисертаційній роботі розроблено ще спеціальний додатковий шар [55], основними функціями якого є обробка та з'єднання різних типів інформації про часові маркери, послідовність подій та характеристики переходів між сценами. Цей шар працює поверх результатів SlowFast. Він призначений для створення комплексного представлення кожної сцени у вигляді її основних характеристик, а саме: її тривалості, деталей переходів, взаємодій між подіями тощо. Інтеграція цих даних дозволяє створити детальну карту відеоконтенту, яка може бути використана для точного індексаційного пошуку та подальшого аналізу. Такий підхід був обраний з огляду на необхідність комплексного розуміння сцени, що враховує всі можливі аспекти – від тривалості до деталей переходів.

Додатковим важливим аспектом аналізу сцени є аналіз характеристик динаміки руху об’єктів всередині сцен, а саме: швидкості руху, напрямку, зміни траєкторій, взаємодії між рухомими об'єктами та їхніми позиціями в просторі. Ці характеристики є визначальними щодо вирішення завдань, пов'язаних з оцінкою інтенсивності подій у сцені та виявленням важливих моментів взаємодії об'єктів. Їх визначення та аналіз є необхідними для удосконалення відомих та створення нових більш точних моделей індексації та пошуку відео за змістом, особливо у випадках коли потрібно ідентифікувати найбільш вагомі моменти сцени для їх подальшого використання в системах автоматичного аналізу відео. Тому наступним важливим завданням синтезу моделі екстракції характеристик сцен у відео є впровадження у неї додаткового шару для обробки характеристик динаміки руху об’єктів всередині сцени та їх подальшого аналізу. Для його вирішення у даній дослідницькій роботі розроблено шар оптичного потоку. В основі цієї розробки покладено метод Лукаса-Канаде [19], поскільки, згідно першого розділу цієї роботи, він є потужним засобом для визначення напрямку і значення швидкості руху та ефективного відстежування переміщень об’єктів сцени в її межах. Результати обробки динамічних характеристик сцени відео розробленим шаром оптичного потоку наведені на рис.2.5.

За даними досліджень [19,24,56] для побудови ефективної індексаційної структури та аналізу відеоконтенту необхідно сформувати зведену інформацію про кожну його сцену. У зв’язку з цим на завершальному етапі синтезу моделі екстракції у даній дисертаційній роботі розроблено шар інтеграції просторових та тимчасових характеристик сцен відео. Основними призначенням цього шару є об’єднання ознак, а саме тривалості сцени, переходів між сценами, динаміки руху об’єктів та послідовності подій у сцені в єдину систему знань про кожну сцену відео (рис.2.6). Доцільність його побудови обгрунтовується результатами досліджень [56], автори яких показали, що інтегровані знання про просторові та тимчасові ознаки сцени дають змогу оцінити як зовнішній вигляд об’єктів сцени так і їх поведінку в часі. Це дає можливість точніше класифікувати сцени, розпізнавати дії та події, що відбуваються у відео. Відсутність однієї з цих ознак

призводить до втрати важливої інформації, що негативно впливає на точність обробки даних.

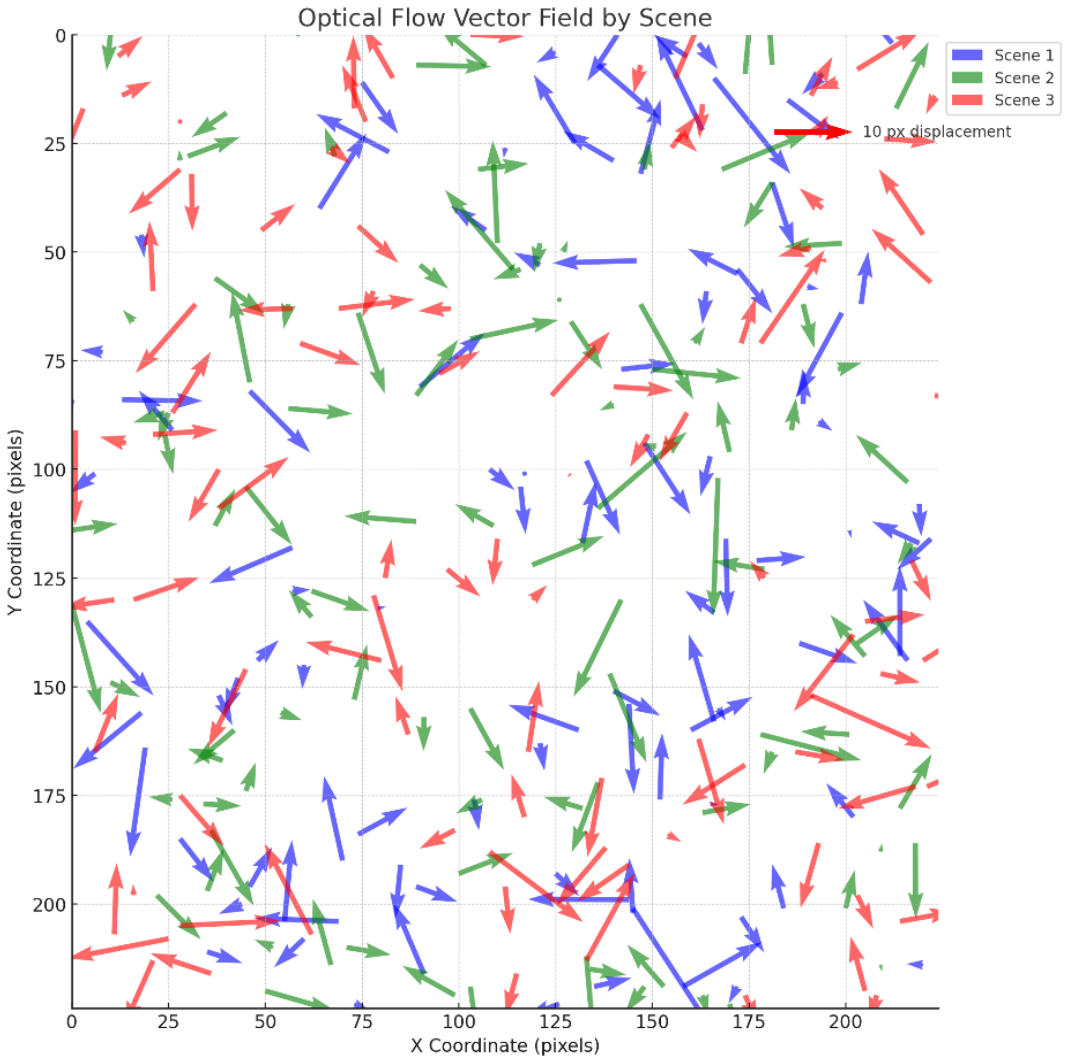

Рисунок 2.5. Поле векторів оптичного потоку за сценою

Інтеграція просторових і тимчасових характеристик дає змогу отримати повний спектр інформації про відео: просторові ознаки допомагають ідентифікувати об'єкти та їхні стани в окремих кадрах, а тимчасові – відслідковувати їхні зміни та взаємодії між кадрами. Наприклад, у відео про спортивні події просторові характеристики допоможуть розпізнати гравців, м'яч та поле, а тимчасові ознаки – відслідкувати переміщення м'яча, його взаємодію з гравцями та визначити ключові моменти (голи, паси). Інтеграція просторових і тимчасових характеристик створює повне багатовимірне представлення відео, що суттєво підвищує точність обробки складних відеоданих. Цей підхід

забезпечує кращу продуктивність у завданнях аналізу відео, таких як класифікація, детекція подій і розпізнавання дій.

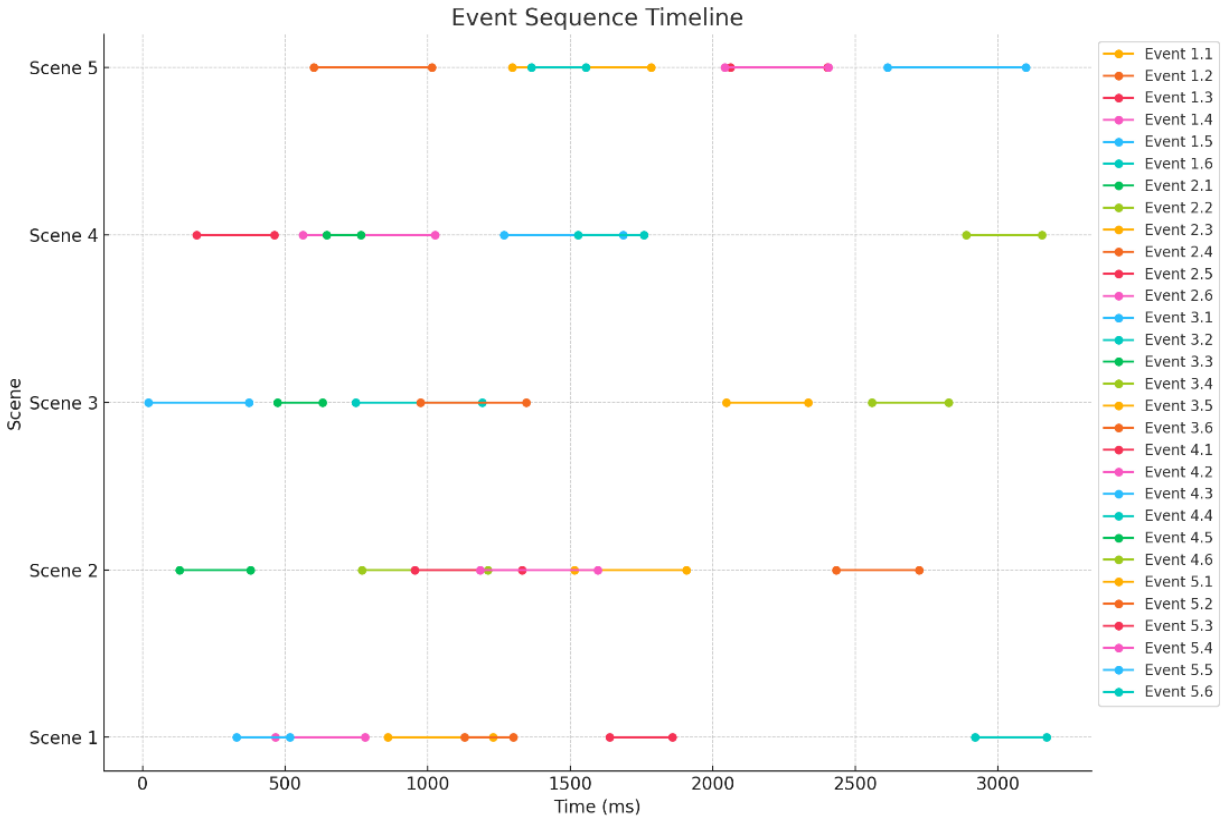

Рисунок 2.6. Результати аналізу характеристик динаміки змін у сценах

Використання нейронної мережі LSTM для обробки послідовностей у поєднанні з розробленими шарами інтеграції просторових і тимчасових ознак, аналізу тимчасових характеристик і оптичного потоку дозволяє створити комплексне представлення про відео. Такий підхід забезпечує високу точність і ефективність в індексації та пошуку відео, особливо при роботі з великими обсягами даних. Інтеграція результатів з кожного з цих шарів дозволяє отримати цілісне уявлення про сцену, що є важливою основою для подальшого аналізу та обробки відеоконтенту [47,51].

2.4. Розроблення модуля системи взаємодії обчислювальних моделей для індексації відео

Для індексації відео у дисертаційній роботі запропоновано модуль взаємодії розроблених моделей детекції та екстракції сцен. Складовими цього модуля є:

- ***SlowFast Networks:*** Використовуються для глибокого аналізу часових масштабів у відео. Ця архітектура нейронної мережі використовує два паралельні потоки: повільний потік для обробки довгострокових, повільно змінюваних ознак і швидкий потік для короткострокових, швидко змінюваних подій. Вона здатна забезпечити всебічний аналіз повільних та швидких подій у сценах відео, що є наочним обгрунтування доцільності її включення у структуру модуля системи взаємодії обчислювальних моделей для індексації відео [55]. У даному випадку цей вибір є оптимальним.
- ***Шар часових маркерів:*** У структурі розробленого модуля шар часових маркерів реалізує функцію інтеграції даних про тривалість сцени, послідовність подій та характеристики переходів між сценами, що є основою створення більш детальної та точної карти відеоконтенту. Доцільність його застосування підтверджується аналізом результатів тестування систем [57], складовими яких є моделі обробки просторових та часових характеристик відео. Зокрема, тестування цих систем на різних наборах даних показало, що врахування послідовності подій та часових маркерів підвищує точність класифікації сцен і покращує ефективність пошуку важливих моментів у відео.
- ***Шар аналізу оптичного потоку:*** Розроблений у даній дисертаційній роботі шар аналізу оптичного потоку є ефективним інструментом точного відстеження динаміки руху об’єктів у сцені. Завдяки покладеному в основу його побудови методу Лукаса-Канаде він “витягає” точні дані про переміщення об’єктів з послідовності кадрів навіть при незначних змінах між кадрами [19]. Окрім цього він забезпечує високий рівень деталізації при аналізі руху у відео. Він легко інтегрується у модуль системи взаємодії моделей для індексації відео, і тому він є важливим компонентом у даній структурі.
- ***Інтеграційний шар:*** Завершує обробку даних, об'єднуючи просторові та тимчасові ознаки для створення повного представлення сцени. Результати

цієї інтеграції використовуються для побудови індексаційної структури [57].

Для забезпечення комплексного аналізу кожної сцени у відео в архітектуру побудованого модуля впроваджено паралельний потік для обробки ключових кадрів (Keyframes Extraction) та ключових об'єктів (Key-objects Extraction) у сцені. Паралельний потік дозволяє системі більш ефективно використовувати ресурси та швидше обробляти відеодані. Це досягається завдяки одночасному виконанню декількох завдань, таких як обробка просторових і часових характеристик сцени та екстракція ключових кадрів і об'єктів. Паралельна обробка зменшує час аналізу та оптимізує використання апаратних ресурсів, що підвищує загальну продуктивність системи. Крім того, впровадження паралельного потоку покращує роботу системи з різними типами відеоконтенту. Гнучкість паралельної обробки дозволяє системі ефективно адаптуватися до відео з різними характеристиками, такими як різна роздільна здатність, частота кадрів або складність сцен. Це підвищує універсальність системи та її здатність обробляти широкий спектр відеоданих.

2.4.1. Методи виявлення та виділення ключових кадрів у відеопотоках

Метод екстракції ключових кадрів ґрунтується на аналізі візуальних змін у відео, які є найбільш значущими для опису подій у сцені. У процесі аналізу змін в кольорах, текстурах, інтенсивності та інших візуальних характеристиках між послідовними кадрами, за його допомогою можна виявити моменти, що містять важливу інформацію про розвиток подій. Однак, інколи цей метод може ідентифікувати незначущі зміни або пропускати поступові важливі моменти. Для покращення точності обробки ключових кадрів модулем взаємодії моделей детекції та екстракції сцен у даній дисертаційній роботі рекомендується використати метод перевикористання шарів шляхом його впровадження у зазначений модуль. Такий підхід забезпечує узгодженість аналізу і дає змогу уникнути дублювання обчислювальних ресурсів, що є важливим для збереження ефективності системи.

Перевикористання шарів для визначення ключових кадрів. Шари, які застосовуються для визначення сцен – особливо ті, що аналізують зміни візуальних характеристик кадрів – виявилися ефективними також і для ідентифікації ключових кадрів. Основною метою цих шарів є аналіз змін у відеопотоці, зокрема у розподілі кольорів, текстур та інтенсивності між послідовними кадрами. Ці шари вже включають потужні інструменти для виявлення змін, які можуть бути індикаторами переходів між сценами або важливих подій у відео.

Процес визначення ключових кадрів. Перевикористання шарів для визначення сцен полягає в аналізі результатів роботи цих шарів на більш детальному рівні. Замість того, щоб лише визначати моменти переходу між сценами, ці шари тепер також ідентифікують моменти значущих змін всередині сцени. Це досягається шляхом аналізу різниць у гістограмах кольорів, обчислених у просторі HSV, з використанням метрики Хі-квадрат для кількісної оцінки різниць між послідовними кадрами. Якщо різниця між кадрами перевищує певний поріг, цей кадр позначається як ключовий [8].

Переваги перевикористання шарів. Перевикористання, розроблених у дисертаційній роботі шарів для визначення сцен та екстракції ключових кадрів має такі переваги:

1. Ефективність обчислень: Перевикористання зазначених вище шарів є ефективним засобом уникнення повторного виконання однакових або схожих обчислювальних операцій для різних завдань визначення сцен та екстракції ключових кадрів. Повторне використання цих шарів призводить до значного зниження загального навантаження на систему та істотного підвищення ефективності обробки великих масивів відеоданих.
2. Консистентність аналізу: Повторне використання шарів визначення сцен та екстракції ключових кадрів покращує узгодженість результатів розпізнавання та інтерпретації однакових або схожих візуальних ознак відео. Завдяки його застосуванню зміни, які визначають початок або кінець сцени, можуть бути розпізнані як ключові моменти. Це сприяє більш

точному та надійному аналізу, оскільки спільні шари розпізнають загальні візуальні ознаки, що застосовуються до обох завдань. Таким чином, результати визначення сцен та ключових кадрів взаємно доповнюють одна одну, покращуючи ефективність та точність системи аналізу відео [22].

3. Покращена точність: Шари, що використовуються для аналізу сцен, є оптимізованими для розпізнавання змін у відео, зокрема таких як перехід між різними сценами, зміни освітлення, рух об'єктів тощо. Вони характеризуються високою точністю у виявленні цих змін. Це означає, що вони здатні ефективно розпізнавати значущі візуальні зміни та витягувати релевантні візуальні ознаки, які є ключовими для розуміння структури та змісту відео. Тому їх перевикористання для екстракції ключових кадрів дозволяє зберегти цю точність і застосувати її до ідентифікації кадрів, які мають найбільше значення для розуміння змісту сцени.

Для вирішення задачі екстракції ключових об'єктів у відео вирішено застосувати підхід, який базується на використанні згорткових нейронних мереж (DCNN), спеціально адаптованих для ефективного аналізу просторових і часових ознак. Зокрема, для вирішення завдань детекції об'єктів обрано легкі архітектури, а саме MobileNet та EfficientNet, які забезпечують високу точність при мінімальних обчислювальних витратах [58].

2.4.2. Обгрунтування вибору архітектури і методології MobileNet та EfficientNet.

На основі аналізу обчислювальних моделей Faster R-CNN та YOLO встановлено, що вони є надто ресурсомісткими для інтеграції в DCNN нейронну мережу, яка вже використовується для детекції сцен. За даними літературних джерел [15,59,60], ці моделі забезпечують високу точність розпізнавання об'єктів. Однак їх практичне використання в більшості випадків призводить до суттєвого збільшення часу обробки відеоданих, що негативно впливає на здатність системи відеоаналізу виконувати свою функцію ефективно та своєчасно, що є неприпустимим у багатьох реальних застосуваннях. Натомість

архітектури MobileNet та EfficientNet забезпечують достатньо високу точність розпізнавання об’єктів сцен відео при одночасному мінімальному використанні обчислювальних ресурсів. Ці архітектури вибрані через їхню здатність до швидкої обробки великих обсягів відеоданих без значного зниження продуктивності. MobileNet, наприклад, використовує глибоку сепарабельну конволюцію, що значно зменшує кількість параметрів та обчислювальних операцій, зберігаючи при цьому високу точність розпізнавання об'єктів. EfficientNet, з іншого боку, оптимізований через систематичне масштабування глибини, ширини та розміру мережі, що дозволяє досягати кращих результатів з меншими ресурсами. Окрім цього, використання MobileNet та EfficientNet є ефективними засобами реалізації моделі на пристроях з обмеженими обчислювальними ресурсами, що забезпечує гнучкість та масштабованість системи відеоаналізу в різних середовищах та сценаріях [60].

Перевикористання шарів детекції сцен для вирішення задачі екстракції ключових об'єктів у відеосцені. Для оптимізації процесу детекції об'єктів у відеосценах вирішено перевикористовувати розроблені у даній роботі шари для детекції сцен. Зокрема, рекомендується використовувати інформацію з конволюційних шарів, яка вже містить детальну карту ознак, для попередньої детекції об'єктів. Після цього ця інформація передається до легких нейронних мереж MobileNet та EfficientNet, що дозволяє уникнути додаткових обчислювальних витрат на аналіз тих же даних з нуля [59].

Процес обробки. Процес екстракції ключових об'єктів можна розділити на такі етапи:

1. Детекція сцен: Цей процес реалізується засобами згорткової нейронної мережі DCNN. DCNN розбиває відео на окремі сцени та за допомогою методу аналізу гістограм у просторі HSV визначає моменти зміни контенту. Після визначення сцени шари DCNN виділяють основні просторові ознаки форми, текстури та кольорові патерни кадрів відео для їх подальшої обробки.

2. Екстракція тимчасових характеристик: Цей процес здійснюється за допомогою зовнішньої попередньо навченої LSTM-моделі, яка забезпечує розуміння та інтерпретацію змін у вмісті відео, враховуючи їх часовий контекст. Використання LSTM-моделі дозволяє ефективно захоплювати тимчасові залежності та закономірності, що виникають у відеопотоці, сприяючи більш точному та глибокому аналізу відео на наступних етапах обробки [59].
3. Паралельна екстракція об'єктів: На цьому етапі обробки процеси витягування тимчасових характеристик та використання просторових ознак об'єктів сцен відео реалізуються одночасно. Витягування тимчасових характеристик здійснюється за допомогою LSTM-моделі, а використання просторових ознак – за допомогою DCNN під час детекції сцен. Використання легких архітектур MobileNet або EfficientNet, дозволяє швидко визначати об'єкти, які є важливими для подальшого аналізу, без збільшення навантаження на систему [44].

Контекстуальний аналіз ролі об'єктів: Важливим аспектом є також контекстуальний аналіз ролі об'єктів у сцені. Цей аналіз враховує, наскільки об'єкт важливий для наративу сцени, тобто як він взаємодіє з іншими об'єктами та персонажами. Він дає змогу не лише виявляти об'єкти, але й зрозуміти їхню роль у сюжеті, що є основою для більш глибокого аналізу відео. Результати екстракції ключових кадрів та об'єктів інтегруються з даними, отриманими від потоків, які витягують тимчасові та просторові характеристики сцени. Така інтеграція дозволяє створити більш повну і точну картину подій у кожній сцені. Ключові кадри служать опорними точками для подальшого аналізу, дозволяючи скоротити кількість даних для обробки без втрати важливої інформації [60]. Це значно підвищує ефективність обробки та зберігає обчислювальні ресурси. Інформація про ключові об'єкти використовується для точнішого розуміння взаємодій у сцені та покращення точності класифікації.

Завершальний етап. На цьому етапі всі отримані дані – тимчасові і просторові характеристики та інформація про ключові кадри та ключові

об'єкти – об'єднуються для створення комплексного представлення сцени. Він реалізується з метою забезпечення глибокого розуміння контексту подій та взаємозв'язків у сцені, що значно покращує точність подальшого аналізу, індексації та пошуку відеоконтенту.

Висновки до 2 розділу

Проведено порівняльний аналіз сучасних нейронних мереж, на основі якого встановлено, що для обробки характеристик відеоконтенту, а отже, і для створення моделі пошуку відео за довільним фрагментом, найбільш оптимальними є мережі, здатні моделювати просторово-часові залежності, адаптивно обробляти динамічний контент і забезпечувати високу продуктивність навіть за обмежених обчислювальних ресурсів. Найбільш ефективними в цьому контексті виявилися DCNN і LSTM-мережі, а також архітектури типу SlowFast Networks, які реалізують паралельну обробку даних у повільному (Slow) та швидкому (Fast) потоках відеоконтенту.

На основі результатів тестування моделей DCNN, LSTM та SlowFast Networks на різноманітних наборах даних встановлено, що ці моделі не завжди демонструють достатню адаптивність до динамічного контенту, зокрема до варіацій у якості відео, змін в освітленні або швидкості руху об'єктів. Виявлено, що одним із шляхів розв’язання цієї проблеми є розширення функціональних можливостей зазначених моделей шляхом інтеграції додаткових механізмів. Тому для підвищення ефективності роботи цих моделей у даному розділі розроблено архітектури шарів детекції та екстракції сцен, а також запропоновано модуль їхньої взаємодії.

РОЗДІЛ 3. РОЗРОБЛЕННЯ МОДЕЛІ ДЛЯ КОНСТРУЮВАННЯ ВЕКТОРІВ ОЗНАК ТА ПОШУКУ ПОДІБНОСТЕЙ У ВІДЕО

Цей розділ роботи присвячений завданням конструювання векторів ознак відео та пошуку подібностей між відеопотоками. Для їх вирішення синтезовано обчислювальну модель для конструювання векторів ознак та побудовано систему пошуку подібностей. В основі синтезу зазначеної моделі покладено процедури перетворення результатів практичної реалізації запропонованих вище моделей детекції та екстракції сцен у числові вектори, а в основі побудови системи пошуку – методи аналізу та порівняння цих векторів.

Матеріали розділу опубліковані у роботах автора [44, 92].

3.1. Синтез обчислювальної моделі для конструювання векторів ознак відео та його обгрунтування

Загальновідомо, що вектори просторових та часових ознак відео – це різні типи даних, які відображають різні аспекти відеоконтенту. Вектори просторових ознак описують статичні характеристики окремих кадрів, а саме характеристики форми, текстури, кольорів та контурів об'єктів. За їх допомогою можна визначити вигляд об’єктів у кожному кадрі, але у жодному випадку – зміни, які відбуваються між кадрами. Вектори часових ознак, навпаки, адекватно відображають динамічні зміни між послідовними кадрами, зокрема зміну положення об'єктів та інші часові залежності. Ці ознаки описують динаміку і розвиток подій у відео. За їх допомогою можна виявляти дії та події, які неможливо зрозуміти, аналізуючи лише окремі кадри. Таким чином, вектори ознак, які є значущими для аналізу статичних характеристик (наприклад, текстури або кольору об'єктів), можуть не мати значення або бути нерелевантними у випадку аналізу часових змін і руху об’єктів у відео [20]. Тому для повного та всебічного розуміння візуального контенту важливими є як вектори просторових так і вектори часових ознак. Одним із способів вирішення цієї задачі є спосіб, сутність якого полягає в інтеграції (поєднанні) просторових та часових характеристик відео в одному числовому векторі для комплексного

опису відео. Такий підхід значно підвищує точність аналізу та розпізнавання дій, об'єктів і подій у відео, оскільки комбіноване представлення забезпечує більш повну і багатогранну інформацію про кожен фрагмент. Він є особливо ефективним у випадку обробки великих обсягів даних та швидкої ідентифікації схожих візуальних патернів. Тому, оскільки основними ознаками відео є часові зміни, рух об'єктів, просторові характеристики кадрів, контекст сцени та ознаки об'єктів, то для формування векторів ознак доцільно використовувати системи, які здатні інтегрувати всі ці характеристики в єдину форму представлення. До таких систем належать сучасні системи контентно-орієнтованого відеопошуку (CBVIR), адже ці системи створені саме для цієї мети: вони здатні об’єднати різні види ознак в одне векторне представлення [61].

У розробленій в цьому дослідженні системі, архітектура якої відображена на рис. 2.4, для отримання просторових і часових ознак відео пропонується використовувати обчислювальну модель SlowFast Networks. Ця модель є однією з найсучасніших архітектур для вирішення завдань комп'ютерного зору, зокрема завдань розпізнавання дій, подій та об'єктів у відео. У ній реалізована концепція паралельної обробки даних (рис.3.1): обробка відео здійснюється двома потоками – повільним (Slow) та швидким (Fast) потоками [13]. Завдяки такій обробці розроблена система здатна одночасно аналізувати статичні просторові характеристики та часові динамічні зміни відео. Використання двох потоків з різною частотою кадрів (повільного для просторових ознак і швидкого для часових ознак) значно зменшує обчислювальні витрати. Ця архітектура забезпечує баланс між ефективністю та точністю за умов збереження високої якості аналізу без значного збільшення обчислювальних ресурсів. Просторові ознаки в SlowFast Networks формуються Conv3D-шарами [23]. Ці шари витягують, обробляють та представляють дані про форму і розташування об'єктів у кадрі. Іншими словами, шари Conv3D аналізують пікселі зображення, щоб виявити та зберегти важливі просторові характеристики, такі як контури, текстури, геометричні форми та позиції об'єктів. Вихідні дані цих шарів для відео можуть бути представлені у вигляді тензора розмірності (T, H, W, C), де T –

кількість кадрів, H і W – висота та ширина зображення, а C – кількість каналів (зазвичай 3 для RGB зображень). Часові ознаки в SlowFast Networks видобуваються за допомогою спеціальних конфігурацій фільтрів, які дозволяють уловлювати рух та зміни об’єктів з часом. Формат даних після цього кроку може бути представлений у вигляді тензора розмірності (T, F), де T – кількість кадрів, а F – кількість ознак, виділених для кожного кадру [6,21].

Для екстракції об'єктних ознак у відео пропонується використовувати архітектури MobileNet та EfficientNet [15]. Доцільність їх практичого застосування обгрунтовується їхніми можливостями щодо забезпечення балансу між високою точністю розпізнавання об’єктів та мінімальними обчислювальними вимогами. Або іншими словами забезпечення балансу між точністю та ефективністю. Окрім цього ці архітектури не потребують значних обчислювальних ресурсів для обробки кадрів відео. Результатом обробки об’єктів кадрів відео моделями MobileNet та EfficientNet [61] є багатовимірний числовий вектор, який адекватно відображає такі характеристики цих об'єктів: клас об'єкта, його координати у кадрі та інші релевантні ознаки. Формат даних, отриманих після обробки, має вигляд (N, M), де: N – кількість об'єктів, виявлених у кадрі; M – кількість характеристик для кожного об'єкта, таких як координати рамки, клас об'єкта, ймовірність належності до класу тощо [62].

Після отримання об'єктних ознак їх необхідно інтегрувати з просторовими та часовими ознаками, отриманими від інших компонентів розробленої мною системи для конструювання векторів ознак відео. Оскільки кількість об'єктів у кадрах може варіюватися, то для агрегації цих ознак слід вибрати методи, які здатні максимально адаптуватися до цієї варіативності кількості об’єктів.

Сьогодні найбільш поширеними методами агрегації об'єктних ознак відео з просторовими та часовими ознаками є методи конкатенції ознак (Feature Concatenation), зваженого усереднення (Weighted Average), агрегація на основі уваги (Attention Mechanism), пулингові методи (Pooling Methods) тощо. Для синтезу моделі для конструювання векторів ознак відео у даній роботі вибрано та агреговано такі три методи: метод конкатенції ознак, метод побудови

середньозваженого вектора та метод агрегації з використанням attention-механізмів.

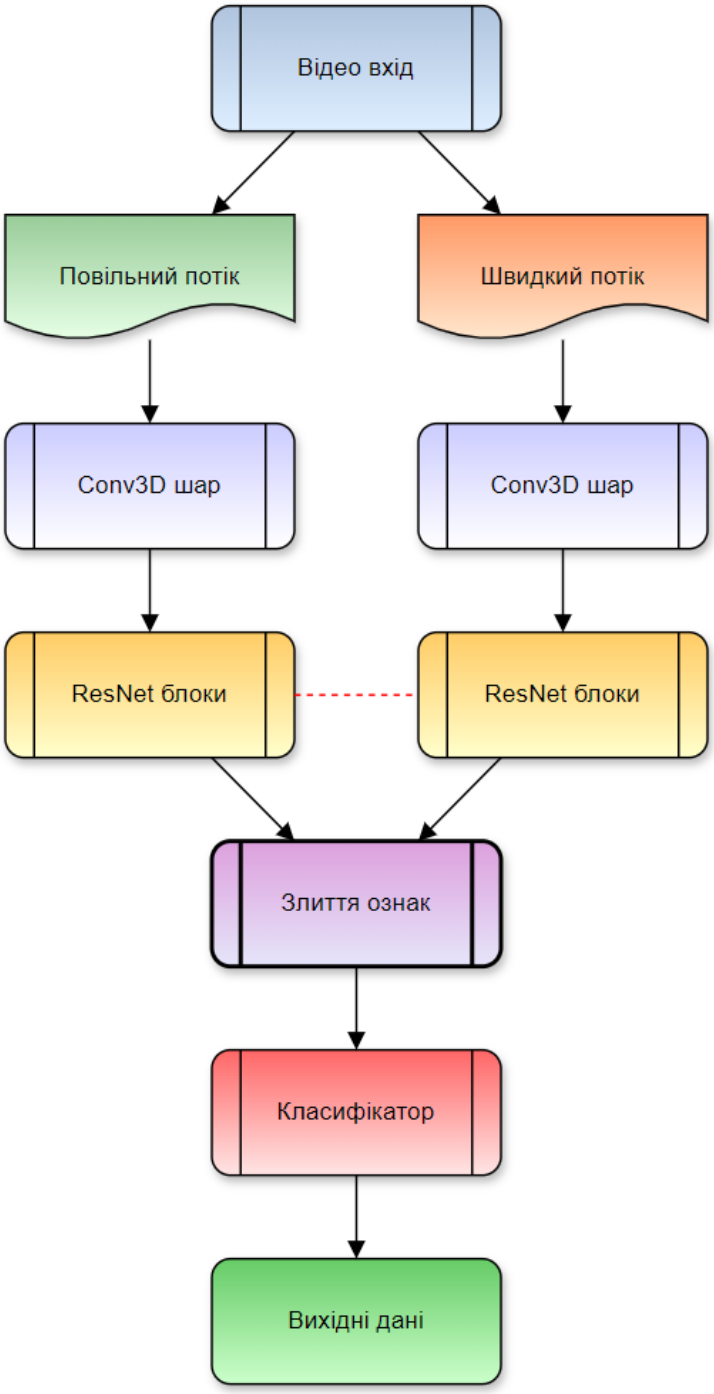

Рисунок 3.1. Блок-схема обробки відео обчислювальною моделлю SlowFast Networks

Метод конкатенації вибрано для об'єднання різних векторів ознак в один великий вектор, метод зваженого усереднення – для надання більшої ваги релевантним ознакам, а механізм уваги (Attention) – для фокусування уваги моделі конструювання векторів ознак на найбільш важливих частинах відео. Цей

підхід є оптимальним, оскільки єдиний великий інтеграційний вектор містить усю необхідну для визначення подібностей або відмінностей між відео фрагментами інформацію. Використання цього вектора істотно зменшує обсяг даних, необхідних для пошуку цих подібностей. У результаті зменшуються обчислювальні витрати, а також витрати на інфраструктуру та енергію системи пошуку. Окрім цього, такий підхід забезпечує узгодженість у представленні даних, які використовуються в системі. Усі ці дані є однаково структурованими та організованими. Це означає, що всі компоненти або модулі системи працюють з єдиним, стандартизованим форматом даних, що істотно спрощує їх обробку, інтеграцію та аналіз [63]. Це полегшує навчання моделей, оскільки система працює з єдиним представленням, яке включає всі необхідні аспекти відео.

Єдиний вектор забезпечує достатній рівень точності для більшості задач пошуку, хоча на перший погляд він може здаватися менш гнучким порівняно з окремими векторами ознак для різних характеристик відео. Точність з якою сформовано вектор є залежною від методів агрегування та нормалізації, які адаптують результати до вимог поставленої задачі. Наприклад, вектор ознак може бути сформований як конкатенація окремих векторів для кожного типу ознаки, де кожна компонента відповідає певному аспекту відео (колір, текстура, рух). Такий формат вектора дозволяє зберігати всі необхідні характеристики, покращуючи таким чином загальну продуктивність системи пошуку [64].

3.2. Розроблення системи пошуку подібностей та оптимізації: обґрунтування вибору математичних моделей та методів

Після розроблення інтеграційного вектора ознак у підрозділі 3.1, наступним важливим етапом є створення системи пошуку подібностей та оптимізації, яка ефективно використовуватиме цей вектор. Метою цього підрозділу є обґрунтування вибору математичних моделей та методів, що лежать в основі розробки цієї системи, зокрема модулів Temporal Similarity Matching (TSIM) та Object Similarity Matching (OSIM). Представлена система базується на

сучасних дослідженнях у галузі комп'ютерного зору та машинного навчання, що забезпечує її наукову новизну та ефективність [12].

Система пошуку подібностей грунтується на використанні інтеграційного вектора ознак V, який об'єднує просторові S, часові T та об'єктні O характеристики відео. Цей вектор є універсальним представленням відеоданих у багатовимірному просторі. Він використовується розробленою системою пошуку подібностей та оптимізації для порівняння відео на основі різноманітних ознак [20].

Формально, інтеграційний вектор визначається як:

$$V[\alpha \cdot S, \beta \cdot T, \gamma \cdot O],$$

де α, β, γ – вагові коефіцієнти, що визначають вплив ознак на загальну подібність, причому $\alpha + \beta + \gamma = 1$.

Для ефективного налаштування параметрів системи пошуку подібностей та досягнення оптимальної продуктивності її роботи використано метод автоматичного налаштування значень вагових коефіцієнтів α, β, γ і відповідні алгоритми оптимізації для їх визначення [12]. Ці алгоритми шляхом коригування значень коефіцієнтів α, β, γ і мінімізації функції втрат на валідаційному наборі даних, забезпечують збалансовану подібність між ознаками, що дозволяє системі коректно оцінювати схожість об'єктів. У результаті підвищується точність і надійність системи, забезпечується оптимальне співвідношення між чутливістю та специфічністю у процесі пошуку відеофрагментів. Окрім цього мінімізується кількість хибнопозитивних і хибнонегативних результатів, покращується загальна ефективність системи та підвищується її здатність точно ідентифікувати релевантні відеофрагменти серед великого обсягу даних [65].

В основі алгоритмів визначення коефіцієнтів α, β, γ покладено методи стохастичного градієнтного спуску для мінімізації функції втрат:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2, \quad (3.1)$$

де y_i – еталонні значення подібності, $\hat{y}_i$ – передбачені значення, N – кількість зразків, θ – набір вагових коефіцієнтів.

На кожному кроці ітерації ваги оновлюються згідно з формулою:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t), \quad (3.2)$$

де η – швидкість навчання. а $\nabla_\theta L(\theta)$ – градієнт функції втрат $L(\theta)$ щодо параметрів θ на ітерації t.

Після кожної ітерації ваги нормалізуються для збереження умови α + β + + γ = 1. Нормалізація здійснюється за формулами:

$$\alpha = \frac{\alpha}{\alpha+\beta+\gamma}, \beta = \frac{\beta}{\alpha+\beta+\gamma}, \gamma = \frac{\gamma}{\alpha+\beta+\gamma} \quad (3.3)$$

Оцінка продуктивності системи пошуку подібностей відеофрагментів здійснюється на валідаційних наборах даних після кожної ітерації оновлення ваг. Звичайно, даний метод має свої обмеження, зокрема такі, як високе споживання обчислювальних ресурсів та потенційний ризик перенавчання, особливо у випадках надмірного пристосування до особливостей валідаційного набору. У таких випадках система втрачає здатність до узагальнення на нових, тестових даних, що призводить до зниження її продуктивності на невідомих даних і суперечить основній меті – узагальнювати закономірності для нових ситуацій. Проте, незважаючи на ці недоліки, регулярна оцінка та коригування вагових коефіцієнтів залишаються одним із найважливіших способів підвищення продуктивності системи у виявленні подібностей між відеофрагментами. Загалом, автоматичне навчання ваг значно покращує об'єктивне та адаптивне налаштування системи, у результаті чого підвищується точність результатів обробки відеофрагментів.

Ключовою властивістю будь-якої системи пошуку подібностей у відео є її чутливість до вибору вагових коефіцієнтів ознак подібності фрагментів, оскільки саме ці коефіцієнти визначають ступінь впливу кожної ознаки (такої як колір, текстура, форма або рух) на кінцевий результат обробки відеоданих. Це означає, що чим більшим є вплив навіть незначних змін вагових коефіцієнтів на вихідні показники системи (такі як точність, повнота чи інші ключові метрики), тим більш чутливішою вона є. Системи з правильно налаштованими вагами характеризуються високими показниками точності та ідентифікації фрагментів відео. Тому для забезпечення ефективності, адаптивності оптимізації параметрів та оцінки стабільності розробленої системи пошуку подібностей та оптимізації

фрагментів у відео проведено аналіз чутливості цієї системи щодо вибору вагових коефіцієнтів ознак подібності. Зокрема, для кожної ознаки подібності (наприклад, текстурних, колірних чи просторових характеристик) визначено діапазони вагових коефіцієнтів, у межах яких оцінювалась стабільність та продуктивність системи. Методологія параметричного варіювання дозволила дослідити вплив різних комбінацій ваг на точність, повноту та F-міру системи, а також виявити критичні коефіцієнти, що найбільше впливають на якість пошуку. Наприклад, при варіюванні вагового коефіцієнта β у діапазоні від 0.1 до 0.5 може спостерігатися збільшення точності з 80% до 90%, після чого точність стабілізується. Це свідчить про важливість часових ознак, оскільки завжди можна знайти таке значення β, при якому їхній вплив на точність і ефективність системи пошуку відео буде максимальним. Візуалізація результатів налаштування вагових коефіцієнтів допомагає ідентифікувати ті коефіцієнти, які найбільше впливають на продуктивність системи пошуку відео, і встановити оптимальні діапазони їх значень [65]. Отримані результати дозволили визначити оптимальні значення вагових коефіцієнтів, які забезпечують збалансоване поєднання різних ознак для точнішого та надійнішого пошуку подібностей між відеофрагментами. Це підвищило адаптивність системи, зробило її стійкою до змін у характеристиках вхідних даних та забезпечило ефективну обробку запитів у різних умовах, що є важливим для подальшого масштабування та розширення системи на інші відеодані.

Розроблена у даному дослідженні система пошуку подібностей фрагментів відео складається з двох основних модулів: модуля порівняння часових ознак (TSIM) та модуля порівняння об'єктних ознак (OSIM). Обидва модулі використовують інтеграційний вектор V, але зосереджуються на різних його компонентах, що дозволяє здійснювати більш детальний та точний аналіз подібності між відео.

Модуль TSIM відповідає за аналіз часових характеристик відео, який здійснюється на основі часових ознак T з інтеграційного вектора. Часові ознаки містять інформацію про послідовність подій, тривалість сцен, зміни руху тощо.

Для порівняння часових послідовностей використовується метод динамічного часової вирівнювання (Dynamic Time Warping, DTW).

Метод DTW призначений для розрахунку мінімальної вартості вирівнювання між двома часовими послідовностями, навіть у випадках коли події в одній відбуваються повільніше порівняно з іншою. Формально, DTW-відстань між двома послідовностями T_1 та T_2, значення якої визначається за формулою [66]:

$$DTW(T_1, T_2) = \sum_{(i,j)\in\pi} d\left(T_{1,i}, T_{2,j}\right), \quad (3.4)$$

де П – множина всіх шляхів вирівнювання, а $d\left(T_{1,i}, T_{2,j}\right)$ – відстань між елементами послідовностей, яка зазвичай вимірюється за допомогою евклідової метрики.

Вибір DTW обґрунтовується його функціональними можливостями. Це ефективний інструмент для адаптивного вирівнювання часових послідовностей різної довжини та з різною швидкістю подій. Він містить всі необхідні функції для точного порівняння часових патернів, зокрема мінімізацію відмінностей між схожими подіями в послідовностях та пошук оптимального шляху вирівнювання. Окрім цього, DTW дає змогу розпізнавати подібності навіть за умов нерівномірного розгортання подій у патернах, тому його використання для вирішення поставлених у даному дослідженні завдань є очевидним [1].

Модуль OSIM зосереджується на аналізі об'єктних ознак O з інтеграційного вектора. Об'єктні ознаки включають інформацію про виявлені об'єкти в кадрах, їх класи, положення, розміри та інші релевантні характеристики. Для порівняння об'єктних ознак використно косинусну міру подібності:

$$cos(\theta) = \frac{O_1 \cdot O_2}{\|O_1\|\|O_2\|}, \quad (3.5)$$

де O_1, O_2 – об’єктні ознаки двох відео.

Використання методу косинусної міри у розробленій у даному дослідженні системі пошуку подібностей об’єктів у відео обумовлено його здатністю ефективно порівнювати векторні ознаки, що відображають характеристики

фрагментів відео, незалежно від їхнього масштабу [45]. Цей метод ґрунтується на порівнянні відносних пропорцій ознак об'єктів, а не їхніх абсолютних розмірів, що є важливим для вирішення задач відеоаналізу, особливо у випадках, коли об'єкти можуть мати різні розміри або з'являтися в різних кількостях у кадрах. Крім цього, метод є обчислювально ефективним: його використання забезпечує швидку обробку великих обсягів даних, що є одним із основних чинників підвищення продуктивності систем пошуку подібностей у відео. Отже, завдяки впровадженню цього методу запропонована система виявилася придатною для аналізу великих відеоколекцій, що підтверджує доцільність його застосування у даній та подібних розробках.

Для усунення впливу різних одиниць вимірювання мір подібності часових та об’єктних характеристик відео на кінцевий результат визначення схожості між відеофрагментами необхідно, щоб ці характеристики мали спільний масштаб. Одним із способів вирішення цієї задачі є приведення результатів обробки часових T та об’єктних O ознак відео, виконаних модулями TSIM і OSIM, до нормалізованого вигляду. Обробка нормалізованих ознак знижує ризик спотворення результатів, спричинених різними діапазонами їх значень, оскільки характеристики цих ознак представлені у стандартизованому форматі. У результаті значно спрощується процес інтеграції даних та підвищується стабільність роботи всієї системи під час аналізу та пошуку подібностей [45].

Для нормалізації векторів ознак використовуються методи мін-макс нормалізації та стандартизації (z-score нормалізація):

В основі методу мін-макс нормалізації покладено формулу [ ]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{3.6}$$

яка масштабує значення ознак до діапазону $[0,1]$.

Тут x – поточне значення ознаки, x_{min} та x_{max} – мінімальне та максимальне значення ознаки у наборі даних. У випадках, коли ознаки мають сильну асиметрію розподілу або містять викиди, перед застосуванням мін-макс нормалізації виконуються попередні перетворення, наприклад логарифмування,

що дозволяє зменшити вплив надвеликих значень та зберегти інформативність інших спостережень.

Метод стандартизації (z-score нормалізація) використовується у випадках коли дані потрібно відцентрувати навколо середнього значення ознаки та звести їх до єдиного масштабу на основі нульового середнього та одиничного стандартного відхилення:

$$x_{norm} = \frac{x-\mu}{\sigma}, \tag{3.7}$$

де μ - середнє значення ознаки, σ - стандартне відхилення.

Z-score нормалізація є ефективною у випадках, коли є потреба у порівнянні числових характеристик ознак незалежно від початкового діапазону їх значень та забезпеченні однакового впливу кожної ознаки на результати аналізу [67].

Після отримання результатів від модулів TSIM та OSIM необхідно об'єднати їх для формування загальної міри подібності між відео. Це здійснюється шляхом зваженої комбінації результатів:

$$Similarity = w_{TSIM} \cdot Sim_{TSIM} + w_{OSIM} \cdot Sim_{OSIM}, \tag{3.8}$$

де w_{TSIM} та w_{OSIM} – вагові коефіцієнти, що визначають внесок кожного модуля у загальну подібність, а Sim_{TSIM} та Sim_{OSIM} – нормалізовані міри подібності, отримані від відповідних модулів [8].

3.3. Метод навчання та адаптації глибокої нейронної мережі для аналізу відео

Згідно, викладеного у підрозділі 3.2 матеріалу основними елементами розпізнавання та порівняння відеофрагментів у контентно-орієнтованих системах пошуку відео загалом, а також у запропонованій системі зокрема, є методи побудови спільного вектора ознак та механізми пошуку подібностей фрагментів. Викладений у другому розділі роботи детальний аналіз алгоритмів витягування характеристик відеофайлів показав, що швидкість та точність пошуку подібних відеофрагментів у великих масивах даних залежать від показників точності алгоритмів сегментації, класифікації та кореляції обробки відео. Від точності кожного з цих алгоритмів залежить загальна якість і

ефективність функціонування систем пошуку, оскільки саме на цих етапах обробки реалізуються методи коректного виділення, структурування та порівняння фрагментів відео. Тому, оскільки основним модулем обробки візуальних ознак у запропонованій системі пошуку є глибока згорткова нейронна мережа (DCNN), то очевидно, що одним із способів підвищення ефективності, точності та надійності цієї системи є оптимізація архітектури та параметрів DCNN [15]. Для вирішення цієї задачі пропонується використання додаткових методів оптимізації, зокрема регуляризації, квантизації та адаптивних алгоритмів навчання. Доцільність їх впровадження у DCNN є незаперечною, оскільки вони містять повний інструментарій для адаптивного підлаштовування ваг мережі DCNN на основі даних попередніх ітерацій навчання. Використання цього інструментарію є ефективним способом адаптації існуючих нейронних мереж не лише до вхідних даних із постійними характеристиками, але і до даних зі змінними характеристиками. Така оптимізація підвищує гнучкість і стійкість DCNN, особливо у випадках обробки тих об’єктів відео, розташування, форма, розмір та зовнішній вигляд яких змінюються у просторі та часі [59].

Інтегровані в основний модуль (DCNN) запропонованої в роботі системи пошуку подібностей об’єктів у відео додаткові методи оптимізації суттєво покращили його можливості щодо обробки складних зорових патернів і візуальної класифікації фрагментів відео. Водночас слід зазначити, що через високу обчислювальну складність модернізованої мережі DCNN продуктивність цього модуля може знижуватися, особливо під час обробки великих обсягів даних або за умов змінних характеристик зйомки (наприклад, варіацій освітлення, кута зйомки чи швидкості руху об’єктів). Тому для забезпечення стабільної продуктивності роботи основного модуля зазначеної системи пошуку виникає необхідність у впровадженні додаткових заходів з оптимізації як його роботи, так і системи в цілому [68].

3.3.1. Архітектура адаптивної нейромережі FNN для моніторингу

Одним із найбільш ефективних способів вирішення цієї задачі є застосування сучасних методів регулярного налаштування параметрів і оптимізації DCNN. При виборі цих методів враховувалася вимога забезпечення ефективної інтеграції повнозв'язної нейронної мережі (FNN) в архітектуру DCNN. Доцільність її використання обґрунтовується функціональними можливостями FNN – здатністю узагальнювати та інтерпретувати ознаки, витягнуті згортковими шарами DCNN, для виконання завдань класифікації, пошуку подібностей між відеофрагментами та аналізу витягнутих ознак для прийняття рішень [59]. Окрім цього, у результаті використання цього підходу істотно підвищуються показники ефективності роботи оптимізованої через інтеграцію FNN моделі DCNN та моделі пошуку подібностей відео, яка пропонується у цій роботі. Зокрема, покращується релевантність знайдених фрагментів та збільшується точність обробки ознак об'єктів відео. Рівень покращення залежить від характеру вхідних даних і параметрів навчання моделі, проте може сягати значних значень у певних експериментальних умова.

Архітектура FNN зазвичай складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Вхідний шар приймає витягнуті ознаки, отримані згортковими шарами DCNN, а вихідний шар використовується для прогнозування результатів, таких як класифікація або оцінка. Використання нелінійних функцій активації, таких як ReLU та Leaky ReLU, у прихованих шарах FNN сприяє швидкій та стабільній обробці даних, а також допомагає уникнути проблеми градієнтного затухання [47,68].

Розрахунок вихідних значень FNN базується на ітеративному процесі навчання з ваговим коригуванням за допомогою алгоритму зворотного поширення помилки. На кожному кроці помилка обчислюється як різниця між прогнозованими вихідними значеннями FNN та істинними мітками, що дозволяє оптимізувати параметри мережі.

Для розширення традиційних функціональних можливостей стандартної DCNN запропоновано інтегрувати до її архітектури рекурентні шари LSTM. У

результаті такого впровадження отримано комбіновану модель DCNN+LSTM, основними компонентами якої є: згорткові шари (convolutional layers), шари підвибірки (pooling layers) для виділення просторових характеристик, а також рекурентні шари LSTM для врахування часових залежностей у даних. Запропонована модель, на відміну від стандартної DCNN, здатна опрацьовувати не лише просторові, але й часові характеристики послідовностей відео, що зумовлено наявністю спеціальних механізмів збереження станів у рекурентних LSTM-шарах. Однак, попри зазначені переваги, модель DCNN+LSTM має певні обмеження. Обчислювальні витрати цієї моделі зазвичай перевищують витрати стандартної DCNN через додаткові ресурси, необхідні для обробки часових залежностей у шарі LSTM.

Для оптимізації обчислювальних ресурсів моделі DCNN+LSTM проведено аналіз сучасних методів та підходів вирішення цього завдання [1,46]. У результаті встановлено, що з відомих на сьогодні методів зниження обчислювальних витрат стандартних нейронних мереж найбільш перспективними є архітектури MobileNet та EfficientNet. Доцільність їх застосування обґрунтовується їхньою здатністю забезпечувати високу точність обробки великих обсягів відеоданих при значно знижених обчислювальних витратах завдяки використанню механізмів збалансованого масштабування ширини, глибини та роздільної здатності нейронних шарів. Оскільки DCNN є частиною комбінованої моделі DCNN+LSTM, то використання MobileNet та EfficientNet як базової архітектури DCNN у цьому випадку є виправданим рішенням для оптимізації DCNN+LSTM на рівні згорткових шарів. На рівні рекурентних шарів (LSTM) їхнє використання є недоцільним, оскільки для оптимізації роботи DCNN+LSTM на цьому рівні потрібно враховувати специфічні витрати, пов’язані з обробкою часових залежностей. За даними літературних джерел [48], такими підходами є методи проріджування (pruning) та квантизації (quantization).

Отже, для оптимізації роботи комбінованої моделі DCNN+LSTM доцільним є використання комбінованого підходу вирішення задачі, ідея якого

полягає у поєднані MobileNet та EfficientNet для DCNN і методів проріджения та квантизації для LSTM [15]. Запропонований підхід дає змогу адаптувати DCNN+LSTM до різних середовищ застосування, включаючи мобільні пристрої, вбудовані системи та високопродуктивні сервери. Його запровадження сприятиме не лише зниженню обчислювальних витрат, але й забезпеченню стабільної продуктивності моделі DCNN+LSTM під час роботи з великими обсягами відеоданих, зберігаючи високу якість обробки як просторових, так і часових ознак.

3.3.2. Алгоритми прийняття рішень на основі моніторингу

Для повноти опису роботи оптимізованої комбінованої моделі проведемо більш детальний аналіз особливостей застосування та функціонування методів проріджения (pruning) та квантизації (quantization) в архітектурі запропонованої системи пошуку подібностей фрагментів у відео

Проріджения (Pruning) мережі. Метод проріджения є загальновідомим і потужним інструментом оптимізації моделей глибокого навчання. Оскільки архітектура DCNN+LSTM поєднує згорткові нейронні мережі (DCNN) для вилучення просторових ознак і рекурентні нейронні мережі (LSTM) для моделювання часових залежностей, кожна з яких належить до класу моделей глибокого навчання, то доцільність використання методу проріджения для оптимізації цієї комбінованої архітектури є природною та виправданою з огляду на результати оптимізації подібних моделей.

З відомих на сьогодні методів проріджения для оптимізації роботи комбінованої моделі найбільш ефективним є використання структурного проріджения для компонентів DCNN і неструктурного проріджения для компонентів LSTM.

Структурне проріджения орієнтоване на видалення тих фільтрів або каналів зі згорткових шарів DCNN, вплив яких на функцію втрат є незначним. Для практичної реалізації цієї процедури за критерій значущості компонентів DCNN вибрано умову:

$$F_i < \varepsilon \ . \tag{3.9}$$

Тут ε – порогове (допустиме) значення величини F_i, яка розраховується за формулою:

$$F_i = \frac{1}{n}\sum_{j=1}^{n}\left|w_{i,j}\right| \ , \tag{3.10}$$

де $w_{i,j}$ – значення ваги j-го з'єднання в i-му фільтрі, а n – кількість ваг у i-му фільтрі.

Поріг ε в умові (3.9) вибирається на основі аналізу значущості фільтрів (розподілу значень F_i), чутливості функції втрат до їх видалення, а також потреб задачі, які визначають баланс між точністю та оптимізацією. Додатково враховуються обмеження апаратної платформи та емпіричні результати, отримані під час тестування моделі DCNN зокрема та архітектури DCNN+LSTM загалом [59]. Наприклад, у задачах реального часу, таких як обробка відеоконтенту, поріг ε може бути обраний вищим, щоб значно зменшити кількість фільтрів і забезпечити швидку обробку кадрів навіть за умов незначного зниження точності. У таких випадках важливим є зменшення обчислювальної складності, що сприяє оптимізації використання ресурсів на пристроях із обмеженою обчислювальною потужністю, наприклад, мобільних телефонах чи embedded-системах. З іншого боку, для задач із критичною важливістю точності, таких як медична діагностика або фінансове прогнозування, поріг ε вибирається нижчим, щоб уникнути видалення навіть малозначущих фільтрів, які можуть мати опосередкований вплив на якість передбачення. У цьому випадку точність моделі має пріоритет над швидкодією, а баланс між продуктивністю та оптимізацією досягається шляхом ретельного тестування та аналізу чутливості.

Визначене за формулою (3.10) середнє абсолютне значення F_i ваг $w_{i,j}$ характеризує значимість i-го фільтра. Справді, якщо це значення виявиться меншим за поріг ε, то згідно нерівності (3.9) i-ий фільтр не є важливим і його можна видалити. У протилежному випадку цей фільтр є важливим, і його вилучення може негативно вплинути не лише на точність моделі DCNN, а й на загальну продуктивність комбінованої моделі DCNN+LSTM. Наприклад, якщо

фільтр із низькою значущістю F_i відповідає за вилучення малопомітних, але важливих просторових ознак, його видалення може призвести до втрати ключової інформації для подальшого аналізу часових залежностей у компоненті LSTM. Це особливо критично в задачах відеоаналізу, де DCNN витягує ознаки, які потім використовуються LSTM для моделювання послідовностей кадрів [48]. У такому випадку видалення важливого фільтра в DCNN може знизити точність ідентифікації динамічних змін між кадрами, що впливає на здатність комбінованої моделі DCNN+LSTM точно розпізнавати події або визначати кореляції у відеопослідовностях. Тому обережний вибір порогу є є вирішальним для забезпечення оптимального балансу між продуктивністю та ефективністю моделі.

Таким чином, впровадження процедури (3.9) – (3.10) суттєво зменшує кількість параметрів у конволюційних шарах, що є основним чинником зменшення кількість обчислень для кожного відеокадру та підвищення швидкості обробки відеоконтенту.

Оптимізація роботи комбінованої моделі DCNN+LSTM методом структурного проріджения є неповною, оскільки після його застосування в моделі залишаються параметри у високоточному форматі, зберігання яких вимагає великого обсягу пам’яті, а їх обробка потребує значних обчислювальних ресурсів. Метод структурного проріджения не вирішує проблем, пов’язаних із високоточним форматом параметрів. Сучасні підходи до вирішення цих проблем, зокрема у випадку моделі DCNN, базуються на застосуванні методів зниження точності представлення параметрів, які передбачають переведення високоточних форматів (наприклад, FP32) у низькоточні (наприклад, INT8). До таких методів належать методи квантизації ваг і активацій [55]. Вони є особливо ефективними у випадках обробки великих обсягів даних на пристроях із обмеженими обчислювальними ресурсами. Тому для досягнення повної оптимізації комбінованої моделі DCNN+LSTM вирішено доповнити методи проріджения методами *квантизації ваг і активацій*. Їх практична сутність полягає у заокругленні ваги *w* до найближчого дискретного рівня [48]:

$$w\tilde{} = Quantize(w, b) = round\left(\frac{w}{\Delta}\right) \cdot \Delta\,, \tag{3.11}$$

де $w\tilde{}$ є найближчим дискретним рівнем, до якого округлюється початкове значення ваги w у процесі квантизації.

Дискретні рівні $w\tilde{}$ визначаються кроком Δ:

$$\Delta = \frac{2^{b-1}-1}{max(|W|)}\,, \tag{3.12}$$

де W – це множина всіх ваг w у певній частині нейронної мережі (наприклад, у шарі або в усій моделі), а max(|W|) – максимальне за абсолютною величиною значення ваги w у множині W.

Крок Δ є залежним від кількості бітів b, які використовуються для подання ваг. Чим меншим є значення Δ, тим вищою буде точність відображення початкових значень ваг у квантизованому форматі, однак це потребуватиме більше обчислень та пам’яті для підтримки точності. Наприклад, якщо значення ваги w = 0,756 квантизувати з кроком Δ=0.1 (при b=8) дискретні рівні будуть $\{\ldots;\ 0.6;\ 0.7;\ 0.8; 0.9\ \ldots\}$, початкове значення 0,756 ваги w округлиться до найближчого рівня $w\tilde{} = 0{,}8$, а помилка становитиме $A = \lfloor 0{,}8 - 0{,}756 \rfloor = 0{,}044$. Якщо ж використати менший крок Δ=0.01 (при більшому b, наприклад, b=16), дискретні рівні будуть $\{\ldots;\ 0.75;\ 0.76;\ 0.77;\ \ldots\}$, вага w заокруглиться до рівня $w\tilde{} = 0{,}76$, а помилка зменшиться до $\lfloor 0{,}76 - 0{,}756 \rfloor = 0{,}004$. Таким чином, зменшення Δ призводить до того, що дискретні рівні, до яких округлюються значення ваг, розташовуються ближче один до одного [59,64].

Формули (3.11) та (3.12) наочно відображають сутність процедури, яка забезпечує необхідну точність збереження інформації про ваги w після квантизації, мінімізуючи ризик втрати якості моделі. Однак зі зменшенням Δ збільшується обчислювальна складність, необхідна для підтримки точності.

Застосування квантизації з правильно підібраним Δ істотно підвищує швидкість обробки ваг при роботі з відеокадрами та суттєво знижує обсяг пам’яті, необхідний для зберігання параметрів моделі DCNN+LSTM [64].

Таким чином, інтеграція методів проріджування та квантизації ваг в архітектуру моделі DCNN+LSTM значно оптимізує та істотно підвищує

ефективність роботи DCNN+LSTM. Проріджения видаляє малозначущі фільтри або канали у згорткових шарах, залишаються лише ті параметри, які найбільше впливають на функцію втрат. Зі зменшенням кількості параметрів, значно знижується обчислювальна складність моделі та суттєво пришвидшуються процеси навчання та інференсує. Тоді як квантизація оптимізує залишкові параметри, знижуючи вимоги до пам’яті та обчислювальних ресурсів, водночас зберігаючи високу точність моделі. Поєднання цих методів є особливо ефективним для вирішення задач реального часу, таких як детекція сцен у відеоконтенті, де ключовими факторами є швидкість та продуктивність.

Важливими завданнями задачі пошуку подібностей у відео є контроль основних показників роботи DCNN+LSTM-нейронної мережі, зокрема часу обробки кадрів, використання обчислювальних ресурсів, чутливості до змін контенту тощо. Ефективним засобом вирішення цих завдань є нейронна мережа прямого поширення (FNN), яка, згідно [70] є ефективним інструментом для аналізу та інтерпретації зазначених показників у реальному часі. Тому для моніторингу показників роботи моделі пошуку подібностей у відео було запропоновано інтегрувати FNN-мережу в архітектуру комбінованої DCNN+LSTM-мережі. Завдяки впровадженню цього компонента вдалося створити модифіковану комбіновану DCNN+LSTM-мережу, яка, на відміну від оригінальної, здатна адаптуватися до змінних умов роботи, зокрема до різних форматів відео, варіацій якості кадрів та роздільної здатності [70]. Структурно-функціональна схема модифікованої моделі наведена на рис. 3.2.

Основними компонентами FNN-мережі інтегрованої в DCNN+LSTM-мережу є:

1. ***Вхідний шар (Input Layer)***. Основною функцією цього шару є прийом даних від DCNN у вигляді показників продуктивності, зокрема таких як точність класифікації, швидкість обробки кадрів, обсяг використаної пам’яті та витрати обчислювальних ресурсів [44].
2. ***Приховані шари (Hidden Layers)*** призначені для обробки числових показників продуктивності роботи комбінованої DCNN+LSTM-моделі.

Вони складаються з нейронів, доповнених функціями активації ReLU (Rectified Linear Unit), завдяки яким модифікована комбінована модель здатна розпізнавати та навчатися складним залежностям у даних, які не можуть бути змодельовані лише лінійними співвідношеннями. Завдяки функціям активації приховані шари, здатні виявляти багатовимірні патерни та тренди, які лінійна модель не здатна відобразити. Наприклад, вони можуть аналізувати взаємозв'язок між зниженням швидкості обробки кадрів та зростанням витрат пам’яті [47].

3. ***Вихідний шар (Output Layer)*** виконує функцію адаптації комбінованої DCNN+LSTM-моделі до змін у вхідних даних, умов виконання завдань та робочих сценаріїв. Його впровадження в архітектуру мережі DCNN+LSTM реалізовано з метою забезпечення стабільної роботи моделі у реальному часі [60]. Алгоритм його роботи полягає в:

1) *Проведенні аналізу показників продуктивності комбінованої моделі,* складовими якого є отримання вихідним шаром числових даних про точність класифікації, швидкість обробки кадрів, витрати пам’яті та використання обчислювальних ресурсів, які були попередньо оброблені прихованими шарами
2) *Формуванні прогнозу ефективності моделі* на основі аналізу зібраних показників, який відображає поточний стан продуктивності та можливі зони покращення DCNN+LSTM;
3) *Генерації сигналів корекції,* які визначають необхідні зміни параметрів моделі для її адаптації до змінних умов, зокрема таких, як зниження частоти кадрів, що спостерігались під час апробації системи на відеоданих з різними джерелами. Зміни цих параметрів впливали на продуктивність DCNN+LSTM, і, відповідно, на основі аналізу таких показників генерувалися сигнали корекції для забезпечення стабільної роботи моделі;
4) *Передачі сигналів корекції до DCNN+LSTM*, де вони використовуються для налаштування параметрів ваги або гіперпараметрів, з метою підтримання стабільної роботи.

Оцінка продуктивності DCNN+LSTM-моделі, доповненої компонентами FNN-мережі базується на використані функції похибки (Error Function) та критеріях адаптації [58].

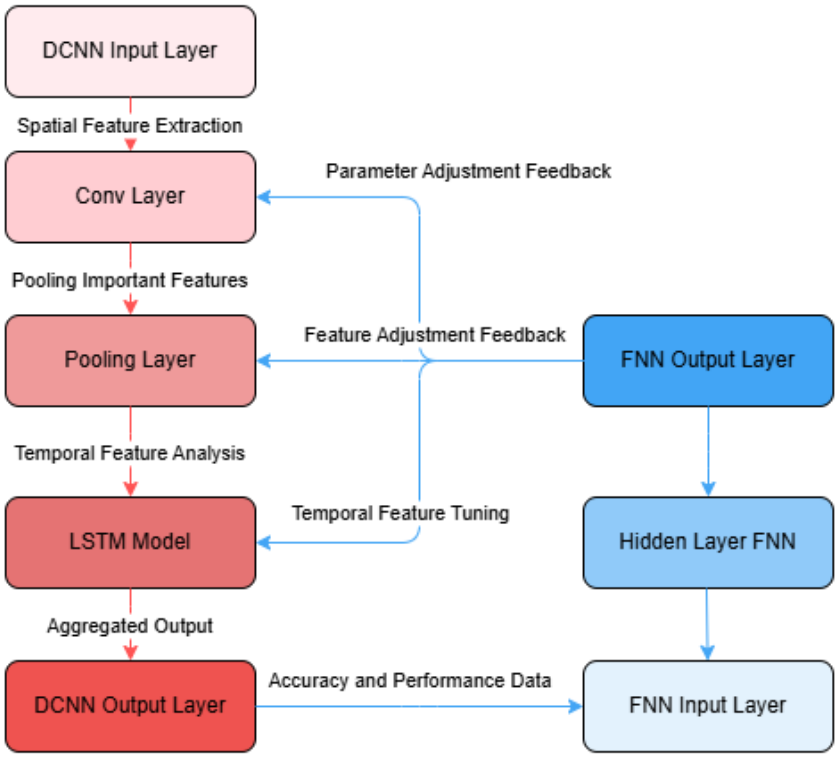

Рисунок 3.2. Архітектура розробленої моделі пошуку подібностей фрагментів у відео

Функція похибки (Error Function) відображає різницю між поточними показниками DCNN+LSTM та еталонними значеннями. Вона використовується для визначення похибки відносно заданих допустимих меж показників продуктивності DCNN+LSTM-моделі. Наприклад, у випадках, коли під час аналізу відеоконтенту DCNN+LSTM-моделлю спостерігається зниження точності або швидкості обробки даних.

Формально похибка для кожного показника E_i визначається за формулою [58]:

$$E_i = \left| P_i - P_{target} \right|, \tag{3.13}$$

де P_i – поточний показник ефективності (наприклад, точність), а P_{target} – еталонне значення.

Критерій адаптації використовується для оцінки поточного стану продуктивності DCNN+LSTM-мережі в умовах, коли модель налаштована на самонавчання та не має фіксованих еталонних значень. У таких випадках, замість статичних еталонів, застосовуються динамічні показники, які обчислюються FNN на основі даних попередніх результатів роботи мережі [44]. Це дозволяє розробленій моделі пошуку подібностей фрагментів у відео, архітектура якої наведена на рис. 3.2, ефективно вирішувати завдання:

1) Ідентифікації поступових відхилень показників продуктивності моделі і трендів, спричинених витратами памяті та іншими обчислювальними ресурсами;
2) Своєчасного виявлення спаду показників продуктивності моделі, особливо у випадках зниження точності класифікації або сповільнення швидкості обробки відеокадрів [58];
3) Формування сигналів корекції для налаштування ваг та гіперпараметрів моделі, що забезпечують стабільність та ефективність її роботи в умовах змін вхідного контенту.

Таким чином, функція похибки та критерій адаптації є важливими компонентами моделі пошуку подібностей фрагментів у відео, оскільки вони забезпечують ефективну оцінку поточного стану продуктивності моделі, своєчасне виявлення відхилень і падіння показників, а також адаптацію параметрів моделі для підтримання її стабільної роботи в умовах змінного контенту. Окрім цього їх інтеграція в архітектуру (рис.3.2) моделі пошуку подібностей фрагментів у відео значно розширює можливості її самонавчання в умовах динамічного контенту [71]. Адже, згідно рис. 3.2, у разі відсутності статичних еталонів самонавчання моделі здійснюється шляхом створення адаптивних еталонів FNN-мережею, які оновлюються динамічно з урахуванням середніх значень показників продуктивності DCNN+LSTM-мережі.

Самонавчання моделі пошуку подібностей здійснюється через навчання FNN, яке базується не лише на адаптивних еталонах, а й на постійному оцінюванні динамічних порогів показників продуктивності DCNN+LSTM. У

цьому процесі FNN використовує динамічні еталони як цільові значення, які оновлюються відповідно до стабільності показників продуктивності DCNN+LSTM. Динамічний поріг цих показників визначається для своєчасного виявлення відхилень їх значень від очікуваного рівня, що забезпечує ефективність і стабільність роботи моделі в умовах змінного контенту.

Кількісною характеристикою динамічного порогу продуктивності роботи DCNN+LSTM-мережі, а також будь-якої іншої мережі, у FNN є функція втрат. Формула для її визначення має вигляд [ ]:

$$L = \alpha|P_i - P^{-}{}_i| + \beta|\Delta P_i|, \quad (3.14)$$

де P_i – поточний показник продуктивності роботи нейронної мережі; $P^{-}{}_i$ – середнє значення показника P_i; ΔP_i – зміна показника продуктивності між ітераціями; α і β – коефіцієнти, які є основними параметрами контролю впливу похибки та тренду. Їх значення дають змогу уникнути надмірної чутливості DCNN+LSTM-моделі до незначних коливань показників її продуктивності [44, 58, 62].

Для навчання FNN-мережі, а отже, і моделі пошуку подібностей (рис. 3.2) загалом, застосовано метод стохастичного градієнтного спуску [44, 62]. Практична доцільність його використання обґрунтовується здатністю оперативно коригувати ваги FNN-мережі у відповідь на зміни показників продуктивності DCNN+LSTM. У FNN цей метод реалізовано за допомогою алгоритмів Adam або RMSprop, які є ефективними інструментами для автоматичного налаштування швидкості навчання та забезпечення швидкої конвергенції моделі пошуку подібностей.

Для оптимізації ваг DCNN і LSTM-мереж моделі пошуку подібностей застосовано підхід, в основі якого лежить використання міні-батчів. Поєднання цього підходу з методами регуляризації, такими як Dropout і L2-регуляризація, суттєво зменшує ризик перенавчання моделі. Практична реалізація цього підходу полягає в адаптивному налаштуванні ваг і їх подальшому оновленні на основі обчислень, отриманих із градієнтів функції втрат [1, 64].

На етапі адаптивного налаштування ваг FNN аналізує продуктивність DCNN+LSTM-мережі та ідентифікує ключові області, які потребують корекції для підвищення точності або швидкості її роботи. Це коригування виконується через тонке налаштування параметрів згорткових шарів та модифікацію параметрів, відповідальних за обробку просторового та часових контекстів у LSTM-шарах. Завдяки динамічним еталонам система автоматично адаптує порогові значення показників продуктивності, що забезпечує ефективну реакцію на зміни у вхідних даних [33].

Процедура оновлення ваг реалізується поетапно. На початковому етапі модель пошуку подібностей формує вхідні дані про поточну продуктивність DCNN+LSTM-мережі для передачі їх до FNN. Далі, за допомогою динамічних еталонів, FNN оцінює, наскільки поточна продуктивність відхиляється від оптимальної. Порівнюючи отримані показники продуктивності з їх еталонними значеннями, FNN розраховує градієнти функції втрат для визначення необхідних коригувальних змін у вагових параметрах DCNN+LSTM-мережі. І, нарешті, на завершальному етапі FNN впроваджує розраховані коригувальні зміни у вагові параметри DCNN+LSTM-мережі. Ці зміни спрямовані на мінімізацію похибки, підвищення точності класифікації та стабілізацію роботи моделі пошуку подібностей.

Висновки до 3 розділу

Встановлено, що для ефективної ідентифікації просторово-часових ознак у відео цінними є моделі, які здатні одночасно аналізувати структуру об'єктів і їхні зміни у часі, виконувати паралельну обробку даних, забезпечувати високу якість аналізу відеоданих за мінімального використання обчислювальних ресурсів, а також адаптуватися до змінних умов. Виявлено, що однією із таких моделей є обчислювальна модель SlowFast Networks у якій реалізовано паралельну обробку відеоданих повільним (Slow) та швидким (Fast) потоками.

На основі нейронних мереж DCNN, LSTM, FNN та моделі SlowFast Networks розроблено архітектуру моделі пошуку подібностей фрагментів у

відео, визначальною особливістю якої є її здатність моделювати просторово-часові залежності та ефективно обробляти великі обсяги інформації в режимі реального часу. Окрім цього для оптимізації роботи моделі у DCNN+LSTM-мережу впроваджено методи проріджения, а саме методи квантизації та активації ваг. Показано, що в порівнянні з відомими архітектурами, орієнтованими на пошук подібностей фрагментів у відеоконтенті запропонована архітектура є більш адаптивною, точною та ефективною.

Визначено засоби розширення можливостей самонавчання моделі пошуку подібностей в умовах обробки динамічного контенту, навіть за відсутності статичних еталонів. Зокрема, запропоновано інтегрувати у FNN-мережу моделі процедури створення адаптивних еталонів та їх динамічного оновлення з урахуванням середніх значень показників продуктивності DCNN+LSTM-мережі. Викладено алгоритм оновлення ваг DCNN+LSTM на основі рекомендацій FNN, а також визначено роль регуляризації для запобігання перенавчанню. Виявлено, що регуляризація, зокрема методи L2-регуляризації та dropout, сприяє збереженню узагальнюючої здатності системи, що забезпечує стабільність і точність її роботи за умов нових та непередбачуваних вхідних даних.

РОЗДІЛ 4 РОЗРОБКА АРХІТЕКТУРИ ТА АПРОБАЦІЯ РЕЗУЛЬТАТІВ

У цьому розділі дисертаційної роботи викладено результати розробки архітектури інформаційної системи для аналізу відеопотоків. Детально описано основні етапи проектування, зокрема вибір технологій для обробки відеоданих, побудову схеми зберігання інформації та структуру функціональних модулів системи. Визначено та представлено діаграми зв’язків між компонентами системи, послідовностей взаємодії модулів та варіантів використання інформаційної системи для аналізу відеопотоків, які ілюструють її роботу в цілому. Окрім цього для підтвердження практичної придатності і ефективності запропонованої системи проведено апробацію результатів її роботи. Зокрема, виконано порівняльний аналіз швидкодії системи, точності алгоритмів обробки та класифікації об'єктів, а також стабільності виявлення динамічних змін у відеопотоках.

Результати розробки реалізовано у вигляді прототипу системи, що демонструє її функціональність. Основні положення та напрацювання, викладені в цьому розділі, опубліковано в наукових працях автора [41,44,52].

4.1. Побудова архітектури інформаційної системи

Основним призначенням запропонованої в даному дисертаційному дослідженні інформаційної системи аналізу відеопотоків є ідентифікація відео за довільним фрагментом. У її розробці використано сучасні підходи до обробки відеоданих, зокрема алгоритми глибокого навчання, методи екстракції ознак ключових кадрів та аналізу змін у відеопотоках.

Основними складниками архітектури цієї системи є сервісна та клієнтська частини. У сервісній частині реалізовано зберігання відеофайлів у файловому сховищі, що забезпечує ефективну роботу з великими обсягами відеоданих. Ознаки, отримані з відео за допомогою алгоритмів глибокого навчання, зберігаються у вигляді векторів у системі індексації FAISS, що забезпечує високошвидкісний пошук схожих фрагментів за метриками подібності. Метадані, результати аналізу сцен, а також додаткові контекстні характеристики

(наприклад, часові мітки, ідентифікатори джерел, дані про ключові кадри) зберігаються в нереляційній базі даних MongoDB, що забезпечує гнучкість структури та масштабованість. У сервісній частині також інтегровано модулі об’єктного та темпорального порівняння, а також механізми оптимізації результатів пошуку: виявлення дублікатів, фільтрація шумових збігів та сортування релевантних результатів для подальшої обробки або відображення на клієнтській частині.

У клієнтській частині виконується обробка запитів користувачів, реалізуються методи сегментації та виділення ключових кадрів, а також механізми аналізу просторово-часових характеристик, що застосовуються для визначення змін у відеопотоках. На основі отриманих даних формуються вектори ознак, які використовуються для ідентифікації як окремих відеофрагментів, так і відео в цілому. Крім того, у клієнтській частині реалізовано інтуїтивний інтерфейс, що надає користувачам можливість здійснювати пошук відео за фрагментом, переглядати результати аналізу та налаштовувати параметри роботи системи.

Обмін даними між сервісною та клієнтською частинами здійснюється через API-сервіс, основним завданням якого є передача запитів від клієнтської частини до сервісної та повернення оброблених даних для відображення користувачеві.

Запропонована інформаційна система для аналізу відеопотоків має модульну архітектуру. Основними компонентами її архітектури є модулі індексації та ідентифікації відео, структурно-функціональні схеми яких наведені на рис. 4.1 та рис. 4.2 відповідно. Вона вирізняється високою надійністю, ефективністю та масштабованістю. Завдяки модульній архітектурі вона легко адаптується до змінних вимог, а висока масштабованість забезпечує можливість її ефективного використання як окремими користувачами, так і великими організаціями. Крім того, система є гнучкою у процесі удосконалення та модернізації, оскільки інтеграція нових компонентів не потребує суттєвих змін у її архітектурі.

Запропонована інформаційна система для аналізу відеопотоків починає свою роботу з отримання відеофайлів або посилань на них із зовнішніх джерел, зокрема від відеовендорів (рис. 4.1). Кожне отримане посилання перевіряється на дублювання шляхом аналізу унікальних ідентифікаторів або векторних ознак відео, з метою уникнення зайвого зберігання однакових даних. У разі виявлення дублювання система автоматично переходить до обробки наступного URL. Якщо відео підтверджується як унікальне, файл завантажується на сервер для подальшої обробки.

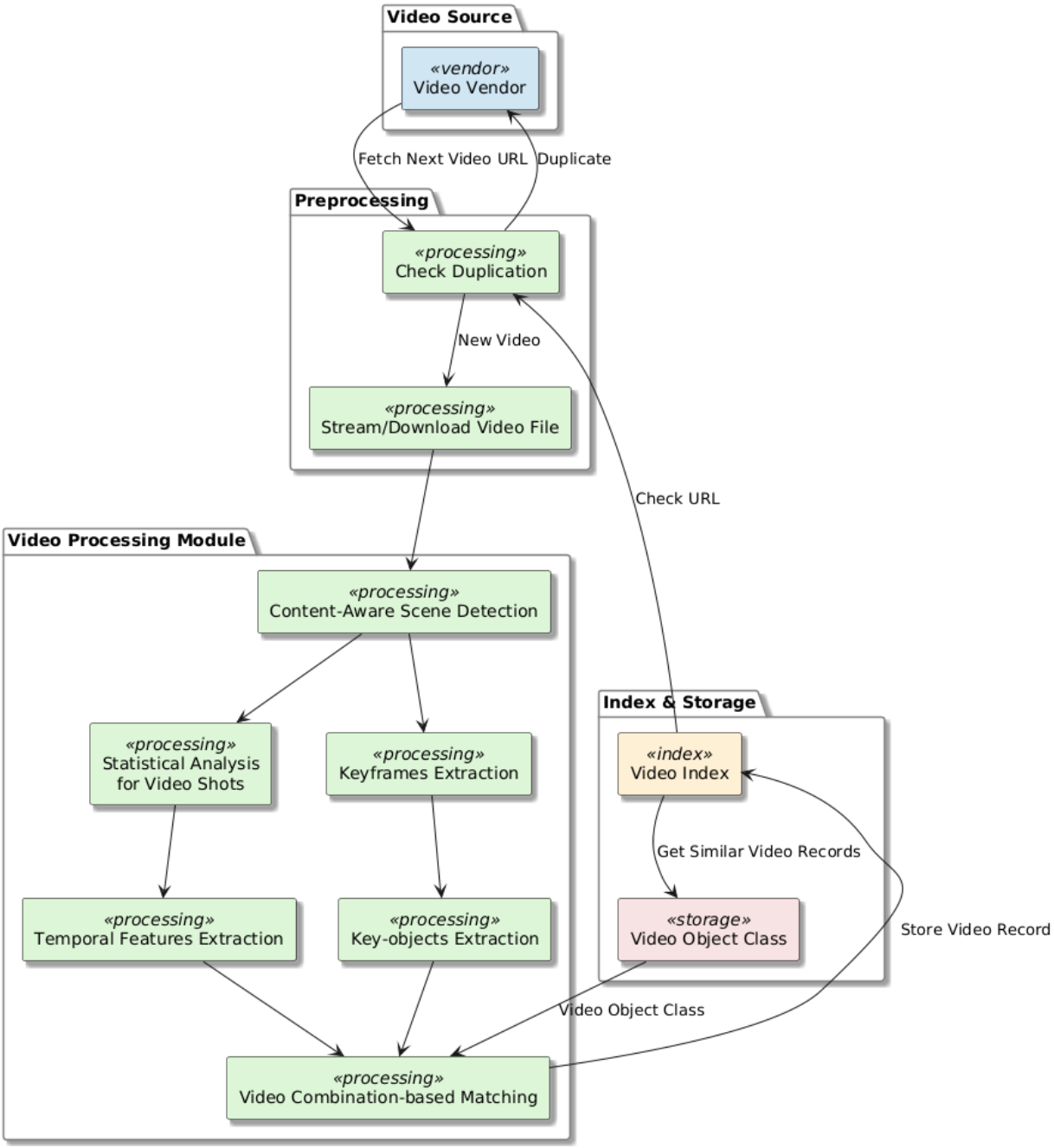

Рисунок 4.1. Архітектура модуля індексації відео

На наступному етапі роботи системи виконується процедура детекції сцен, яка здійснює сегментацію відео на логічні частини з урахуванням змін у його змісті, зокрема змін кольорової композиції, текстур, руху об'єктів або різких переходів між кадрами. Далі, після сегментації для кожної сцени виділяються кадри, які найбільш точно відображають її вміст. Для цього використовуються методи комп'ютерного зору та попередньо навчені моделі нейронних мереж, які з високою точністю визначають ключові візуальні елементи відео. Виділення ключових кадрів суттєво зменшує обсяг даних, необхідних для подальшого аналізу, що підвищує ефективність та продуктивність роботи системи. Паралельно з цим проводиться статистичний аналіз відеофрагментів, під час якого визначаються основні характеристики відео, такі як тривалість сцен, частота кадрів, кольорова гамма та динаміка змін. На основі результатів цього аналізу формуються унікальні ознаки відео.

На наступному етапі здійснюється виділення просторово-часових характеристик (темпоральних ознак) та ключових об'єктів, які є важливими для розпізнавання та ідентифікації відео. Цей етап також реалізується із застосуванням попередньо навчених моделей нейронних мереж, що забезпечують точний аналіз об'єктів і їхніх взаємодій у відео.

Характерною особливістю запропонованої системи є те, що на кожному етапі обробки відеофрагментів зібрані дані (метадані та ознаки) зберігаються в індексованій базі даних, яка забезпечує швидкий доступ до інформації та ефективний пошук подібних відеофрагментів. Завдяки структурованому зберіганню даних у базі оптимізується використання обчислювальних ресурсів під час виконання запитів. На основі цих даних у системі із застосуванням алгоритмів комп'ютерного зору та глибокого навчання реалізується порівняльний аналіз відеофрагментів для визначення їхньої схожості за контентом, сценами чи ключовими об'єктами.

Важливою складовою архітектури розробленої інформаційної системи аналізу відеопотоків є серверна частина, оскільки в ній здійснюється аналіз інформативних характеристик відеофрагментів, отриманих від користувача,

зокрема ознак сцен та їхніх відповідних векторів. Крім того, у ній реалізуються алгоритми ідентифікації відео, зокрема методи порівняння отриманих характеристик із векторами ознак, сформованими у клієнтській частині та збереженими в індексованій базі даних.

Основним модулем в архітектурі серверної частини є модуль ідентифікації відео, структурно-функціональна схема якого відображена на рис.4.2. Згідно з цією схемою, функціонування цього модуля починається з передачі векторів ознак до його основних компонентів: Temporal Similarity Matching (TSIM) та Object Similarity Matching (OSIM), які входять до складу модуля комбінаційного порівняння (Features Combination Module).

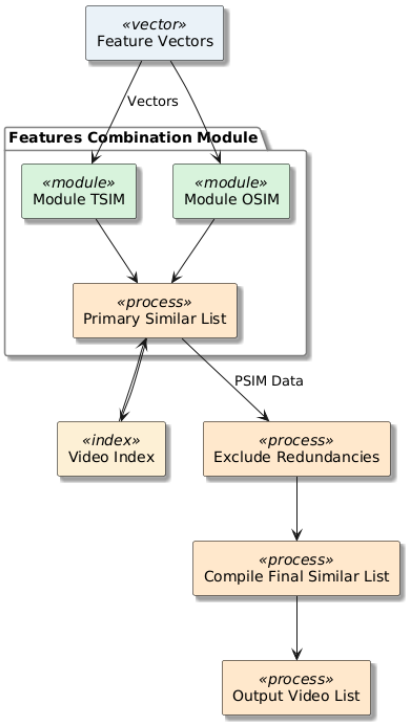

Рисунок 4.2. Архітектура модуля ідентифікації відео

У TSIM реалізується процедура аналізу часових характеристик відео. Визначається порядок подій, тривалість сцен та частота кадрів, а на основі отриманих результатів формується список відео, схожих за темпоральними параметрами.

У OSIM створюється список відео, схожих за візуальними ознаками, такими як колір, текстура та форма. Додатково здійснюється порівняння об'єктних характеристик відео, що дозволяє уточнити ступінь їхньої схожості.

На основі цього формується остаточний список відео, подібних за візуальними параметрами.

Обидва модулі, TSIM і OSIM, функціонують паралельно, що значно пришвидшує обробку даних. Результати їх роботи об'єднуються у первинний список схожих відео (Primary Similar List) за допомогою адаптивного алгоритму комбінації, який враховує вагові коефіцієнти для кожного типу ознак.

Такий підхід забезпечує гнучке налаштування системи відповідно до різноманітних запитів користувачів та підвищує точність формування проміжних результатів шляхом оптимального балансу між темпоральними та об'єктними характеристиками.

Дані з Primary Similar List передаються до модуля оптимізації, який поетапно виконує кілька важливих функцій:

- Очищає список від дублікатів, які могли з’явитися через збіг у результатах TSIM і OSIM.
- Формує фінальний список схожих відео (Final Similar Video List), у якому результати сортуються за релевантністю та точністю, щоб забезпечити максимальну відповідність пошуковим запитам.

Остаточні результати передаються до вихідного модуля (Output Video List), де вони форматуються у структурований список, зручний для перегляду користувачем. Користувач отримує відсортований список відео, що відповідають його запиту, з можливістю перегляду деталей та подальшої взаємодії із системою.

Завершальним етапом функціонування системи аналізу відеопотоків є взаємодія між серверною та клієнтською частинами в процесі пошуку подібних відеофрагментів. Механізм цієї взаємодії наочно представлено на рис. 4.3, де поетапно відображено повний ланцюг обробки — від моменту надсилання запиту до формування відповіді з релевантними результатами.

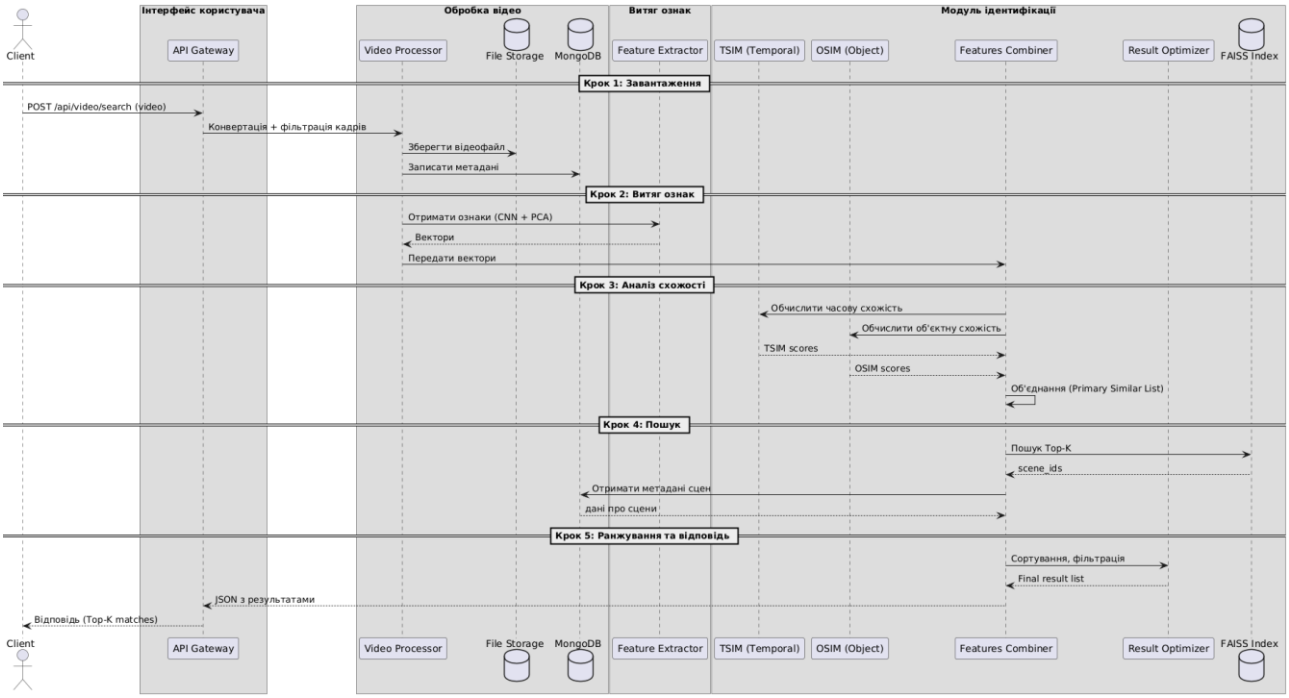

Рисунок 4.3. Послідовність взаємодії компонентів системи при пошуку відеофрагментів за подібністю з використанням ознак, отриманих у результаті попередньої обробки

Користувач через клієнтський інтерфейс надсилає відеофайл або його фрагмент до API-сервісу. В API виконується валідація запиту та ініціалізується процес попередньої обробки: відео конвертується у стандартизований формат (H.264, 30 FPS) з усуненням зайвих кадрів. Оброблений файл зберігається у локальній файловій системі, а технічні метадані записуються до бази даних.

Далі відео передається до модуля витягу ознак, де за допомогою глибинних моделей (EfficientNet та LSTM) формуються вектори ознак, які репрезентують просторово-часові характеристики фрагмента. Ці вектори передаються до модуля ідентифікації, що складається з двох паралельних підсистем: TSIM (Temporal Similarity Module) та OSIM (Object Similarity Module). Кожна з них аналізує фрагмент за своїм критерієм, після чого результати комбінуються у попередній список релевантних сцен.

На наступному етапі модуль ідентифікації виконує пошук у FAISS-індексі для визначення найбільш подібних векторів, а також звертається до MongoDB для отримання додаткових метаданих про знайдені сцени. Результати оптимізуються: видаляються дублікати, виконується сортування та ранжування.

Сформований фінальний список результатів передається до API-сервісу й повертається клієнтові у вигляді структурованої JSON-відповіді.

Таким чином, процес ідентифікації реалізовано як синхронну послідовність викликів, що забезпечує мінімальну затримку й негайне отримання результатів пошуку. Асинхронна взаємодія (через RabbitMQ) використовується лише у сценаріях повної індексації відеофайлів і не залучається під час інтерактивного пошуку.

4.2. Проектування структури сховища даних

З попередніх розділів цієї дисертаційної роботи випливає, що обсяг даних, які обробляються інформаційними системами аналізу відеопотоків, зокрема запропонованою системою, є значним та структурно неоднорідним – від метаданих відео до багатовимірних ознак, сформованих нейронними моделями. Така неоднорідність вимагає застосування багаторівневої системи зберігання, що поєднує різні типи сховищ: відеофайли фізично зберігаються у файловому сховищі, метаінформація – у документно-орієнтованій базі MongoDB, а векторні представлення – у спеціалізованому індексі FAISS.

Збереження відеофайлів поза СУБД дозволяє уникнути проблем із масштабуванням та перевантаженням бази при обробці великої кількості потоків. Для забезпечення високошвидкісного пошуку фрагментів за ознаками, отриманими в результаті глибинного аналізу (зокрема ознак CNN+LSTM), система використовує FAISS – векторну базу даних із підтримкою індексів типу IVF-PQ. Пошук у FAISS базується на скороченому векторному представленні (PCA-reduced vectors), а метадані до векторів зберігаються у відповідних колекціях MongoDB (video_scenes, features) для швидкого зіставлення результатів із відеофрагментами.

MongoDB у цій архітектурі використовується не як класична заміна реляційної СУБД, а як ефективне рішення для зберігання ієрархічно організованої та динамічно змінюваної метаінформації. Вона дозволяє створювати логічно пов’язані дані (videos, scenes, keyframes, features). Посилання

між документами реалізовано через унікальні ідентифікатори, що дозволяє виконувати, агрегаційні запити із збереженням продуктивності навіть за великих обсягів даних. Таким чином, у MongoDB структура даних адаптована під типові завдання пошуку за векторною схожістю: дані, що використовуються для пошуку, зберігаються в FAISS, а супровідна інформація – у MongoDB [72].

Порівняно з реляційними СУБД, які орієнтовані на нормалізовані таблиці та складну модель зав’язків, MongoDB не потребує жорстких схем і дозволяє адаптувати структуру збереження під конкретні вимоги глибинного відеоаналізу. Крім того, механізми горизонтального масштабування та підтримка репліка-сетів забезпечують стабільну роботу навіть за високих навантажень. Завдяки шардінгу за ключами система зберігає здатність до масштабування без деградації продуктивності

Окрім технологічних переваг, MongoDB має ліцензію SSPL, що робить її економічно доцільною для використання на початкових етапах розробки та апробації системи. Така модель зберігання, де компоненти FAISS і MongoDB логічно доповнюють одне одного, є ефективною відповіддю на виклики зберігання та обробки відеоданих у сучасних CBVIR-системах.

4.2.1. Переваги та обмеження використання спеціалізованих векторних баз даних у задачах аналізу відеопотоків

У межах архітектури розробленої системи аналізу відеопотоків, для забезпечення ефективного управління контекстними даними про відео та результатами аналітичної обробки, реалізовано спеціалізовану модель збереження метаінформації на базі документно-орієнтованої СУБД MongoDB [73]. Вона слугує координаційним шаром між вхідними відеоданими, індексом ознак у FAISS та логікою обробки на рівні прикладної частини системи.

MongoDB у цій структурі відповідає за агреговане представлення відеофрагментів, ключових кадрів та векторних ознак, що дозволяє забезпечити швидкий доступ до даних, незалежно від формату початкового відео чи типу обраної моделі ознак. Для цього реалізовано логічно цілісну модель з двома

основними колекціями – videos та features, що акумулюють інформацію про вміст візуального контенту та відповідні векторні описи.

Колекція *videos* містить метадані щодо відеофайлів, опис сцени, часові межі, характеристики переходів та список ключових кадрів із часовими позначками та шляхами до їх зображень. Структура документа включає:

- video_id: унікальний ідентифікатор відео;
- title: назва відеофайлу;
- path: шлях до збереженого файлу (локальний);
- upload_date: дата завантаження;
- status: етап обробки відео (UPLOADED, PROCESSING, DONE);
- technical: вкладений об’єкт із параметрами тривалості, формату, роздільної здатності та розміру файлу;
- scenes: масив сцен, де кожна містить:
    - scene_id: ідентифікатор сцени;
    - start_frame, end_frame, frame_rate, transition_type;
    - keyframes: масив ключових кадрів з полями keyframe_id, timestamp, thumbnail_path.

Така структура забезпечує логічну цілісність і спрощує виконання запитів до інформації про фрагменти.

Колекція *features* реалізує механізм зберігання результатів глибинного аналізу відеофрагментів у вигляді векторних представлень. Кожен запис пов’язаний з відеофрагментом через поля video_id, scene_id, keyframe_id і містить:

- feature_id: унікальний ідентифікатор ознаки;
- feature_type: тип ознаки (CNN_EMBEDDING, LSTM_TEMPORAL, OPTICAL_FLOW);
- vector_id: ідентифікатор відповідного вектора у FAISS-індексі;
- model_metadata: об’єкт із результатами моделі:
    - confidence, detected_objects, bounding_box,
    - color_histogram, optical_flow, timestamp.

Це дозволяє зберігати не лише вектори, а й усі параметри, що можуть бути використані при повторній інтерпретації або комбінованому аналізі результатів.

Для ефективної роботи у розподіленому середовищі в колекції features використовується комбінований shared key – video_id + feature_type. Така схема має кілька переваг:

- забезпечує локальність доступу до ознак одного відео;
- дозволяє логічно розділити навантаження між різними типами моделей ознак (просторових, часових, об'єктних);
- гарантує однорідний розподіл записів при масштабуванні системи.

Цей підхід також добре узгоджується з пошуковим шаром FAISS, який використовує vector_id як точку входу для векторного запиту, а MongoDB – як постпроцесингове джерело для розшифрування знайдених збігів. Структурно-функціональна схема цієї моделі наведена на рис.4.4.

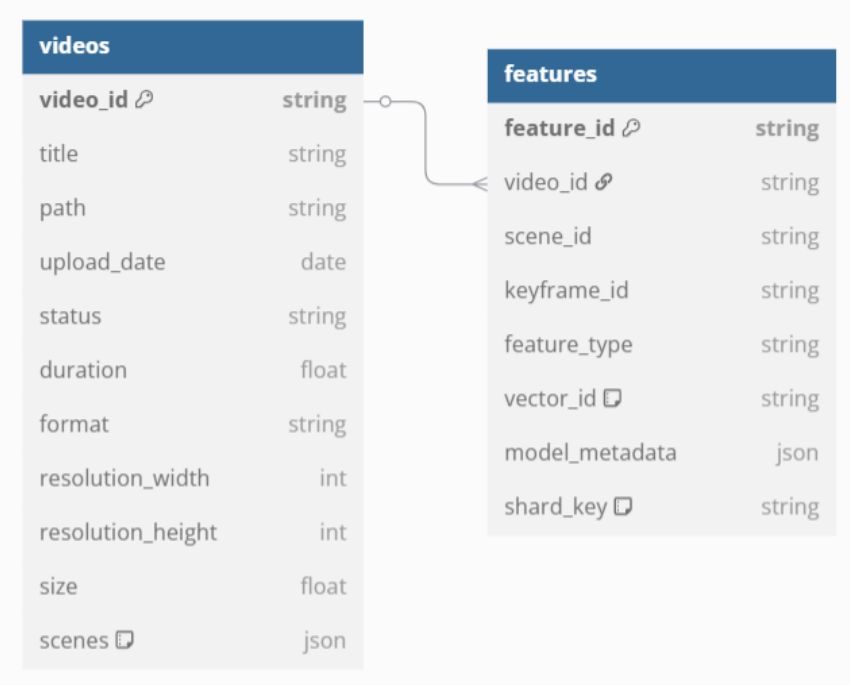

Рисунок 4.4. Схема бази даних

На завершення викладу матеріалу зазначимо, що запропонована структурно-функціональна схема збереження метаінформації (рис. 4.4) є масштабованою та оптимізованою для аналізу відеоданих. Завдяки підтримці механізму шардінгу в MongoDB кожна колекція може бути розподілена за окремими ключами, що забезпечує рівномірний розподіл навантаження між

вузлами бази даних та ефективну обробку великих обсягів даних у розподіленому середовищі. Ця схема є гнучкою, оскільки архітектура MongoDB містить усі необхідні механізми для динамічної зміни структурно-функціональної схеми збереження метаінформації та її адаптації до нових вимог без необхідності складних міграцій, характерних для реляційних баз даних. Завдяки цьому інформаційна система аналізу відеопотоків залишається гнучкою та придатною для поступового розширення функціоналу без суттєвих змін у вже існуючих структурах, що забезпечує її довгострокову ефективність та масштабованість.

Окрім цього, пошук векторної подібності в системі здійснюється за допомогою FAISS, який забезпечує ефективний пошук з використанням індексів IVF-PQ. FAISS обрано у зв’язку з:

- відкритим вихідним кодом і локальним контролем над даними;
- високою швидкодією для підготовлених векторних представлень (PCA-знижених);
- можливістю інтеграції з Python на рівні сервісу ознак.

Водночас архітектура системи не обмежується використанням FAISS. Вона підтримує можливість інтеграції з іншими векторними базами даних – такими як Milvus, Weaviate або Pinecone – шляхом абстрагування пошукового шару. Milvus, наприклад, надає механізми розподіленого зберігання з автоматичним балансуванням, Weaviate дозволяє поєднувати векторний пошук зі структурованою фільтрацією, а Pinecone – сервісну масштабованість у хмарному середовищі. Вибір FAISS як основного пошукового інструменту обґрунтовано вимогами до автономного локального розгортання, можливістю гнучкого налаштування та зменшенням залежності від зовнішніх сервісів.

Таким чином, модель даних, реалізована у MongoDB, у поєднанні з векторною базою FAISS, забезпечує ефективну організацію метаінформації, масштабованість, модульність та підтримку зовнішніх векторних індексів. Це дозволяє системі не лише зберігати структуровано великі обсяги відеоданих, але й здійснювати високоточний пошук за вмістом у реальному часі.

4.3. Розробка інформаційної системи

В основі розробки та функціонування запропонованої системи пошуку відео за фрагментом, архітектура та структура сховищ даних якої визначені й детально описані у підрозділах 4.1 та 4.2, лежать сучасні веб-технології:

- Next.js/React.js – для клієнтського інтерфейсу;
- .NET (C#) API – для обробки запитів і доступу до бази даних;
- Python (TensorFlow, PyTorch) – для реалізації алгоритмів аналізу відео.

У серверній частині системи впроваджено підхід CQS (Command–Query Segregation), який реалізовано в середовищі .NET 9 із використанням Minimal API [74] та бібліотеки Wolverine [75]. Використання CQS з Faiss як окремої моделі читання забезпечує часткову відповідність принципам CQRS, зокрема ефективний розподіл навантаження між операціями читання та запису. Це дозволяє зменшити конфлікти доступу до даних, підвищити продуктивність запитів і забезпечити масштабованість системи відповідно до потреб обробки відеопотоків.

Використання бібліотеки Wolverine спрощує управління запитами (Query) та командами (Command), забезпечує асинхронну обробку подій, що є критично важливим для індексації та глибинного аналізу відео. Minimal API значно спрощує архітектуру серверної частини, зменшує кількість коду, необхідного для обробки HTTP-запитів, що, у свою чергу, знижує накладні витрати та покращує читабельність коду. Завдяки інтеграції Wolverine і Minimal API розділення команд (Command) і запитів (Query) реалізується без надлишкових витрат (таб. 4.1.). Такий підхід дозволяє адаптувати компоненти системи до нових функціональних вимог без необхідності значних змін у кодовій базі.

Таблиця 4.1. Логічна схема потоків обробки (CQS-підхід)

Потік	Операції	Write-store
Command(конвертація, індексація)	INSERT / UPDATE	MongoDB (video_meta, scenes) + додавання вектора у Faiss
Query(пошук Top-K)	ANN-пошук → lookup метаданих	—

Архітектура серверної частини системи побудована за функціональним принципом. Її складовими компонентами є взаємопов’язані мікросервіси, кожен з яких виконує окрему задачу. Основними мікросервісами є сервіси обробки відеофайлів, індексації ознак відео, управління збереженням метаданих і відеофайлів, а також пошуку подібностей відеофрагментів.

Сервіс обробки відеофайлів виконує первинне зчитування, конвертацію та збереження відеопотоків. Його основне призначення – перетворення вхідного відеофайлу у стандартний формат, що використовується в системі, а також нормалізація параметрів, зокрема роздільної здатності, частоти кадрів тощо. Крім того, сервіс створює технічні метадані відео та здійснює його сегментацію на окремі сцени й кадри для подальшої обробки іншими сервісами.

Після завершення попередньої обробки відео передається до сервісу індексації ознак, який відповідає за аналіз вмісту відеофрагментів, екстракцію ознак та генерацію векторних представлень із використанням бібліотек TensorFlow та PyTorch. Він формує метаінформацію, яка використовується для швидкого пошуку та порівняння відеофрагментів.

Збереження відеофайлів та метаданих здійснюється за допомогою сервісу управління збереженням, який взаємодіє з базою даних і відеосховищем. Він відповідає за організацію ефективного доступу до відеофайлів, збереження структурованої метаінформації та оновлення даних у разі змін. Сервіс забезпечує запис і отримання метаданих відеофрагментів, сцен і ключових кадрів, а також підтримує індексацію для швидкого пошуку та отримання необхідної інформації. Крім того, він оптимізує розподіл даних між сховищами, що мінімізує затримки під час запитів і підвищує продуктивність обробки відеопотоків.

Для реалізації пошуку за подібністю використовується сервіс пошуку, який порівнює відеофрагменти на основі векторних подібностей. Він аналізує вхідні запити користувачів та, застосовуючи методи зіставлення ознак і векторних представлень відео, формує релевантні результати пошуку.

Синхронна взаємодія між мікросервісами здійснюється через API-шлюз, який об'єднує всі точки входу для клієнтських застосунків і зовнішніх сервісів.

Для асинхронного обміну повідомленнями між окремими сервісами в архітектуру серверної частини системи пошуку відео за фрагментом інтегровано брокер повідомлень RabbitMQ. Він забезпечує надійну передачу даних, рівномірний розподіл запитів і завдань між серверами, а також дає змогу зменшити навантаження на мікросервіси. Окрім цього, використання RabbitMQ допомагає уникнути блокувань під час виконання обчислювально інтенсивних завдань, таких як індексація відеофайлів або глибинний нейромережевий аналіз [76].

Взаємодія користувача з API здійснюється через набір ендпоінтів – точок доступу до вебсервісу серверної частини системи пошуку відео за фрагментом. Кожен ендпоінт визначається унікальною URL-адресою та відповідним HTTP-методом. URL-адреса вказує на розташування ресурсу в мережі Інтернет або локальній мережі та визначає спосіб доступу до нього, а HTTP-метод визначає тип операції, яка може бути виконана над цим ресурсом. Перелік основних кінцевих точок наведено в таблиці 4.2.

Таблиця 4.2.

Методи API для завантаження, індексації та пошуку відео та URL-шляхи

Методи	URL-шлях (endpoint)	Опис
POST	/api/video/upload	Завантаження відеофайлу
POST	/api/video/index	Запуск процесу індексації відео
GET	/api/video/{id}	Отримання інформації про відео
POST	/api/video/search	Пошук відео за подібністю

Таким чином, робота системи пошуку відео за фрагментом починається з моменту завантаження відеофайлу або його фрагмента. Після цього завантажене відео обробляється модулем обробки відеофайлів, у якому виконується низка операцій для підготовки контенту перед подальшою обробкою. На початковому етапі здійснюється первинне зчитування та перевірка файлу: аналізується завантажене відео, перевіряється його відповідність підтримуваним форматам (MP4, AVI, MKV) і здійснюється витяг основної метаінформації, зокрема роздільної здатності, частоти кадрів і типу кодека. Якщо файл не відповідає визначеним вимогам або має пошкоджену структуру, система повідомляє про помилку та припиняє його подальшу обробку. У разі успішної перевірки відео

конвертується до єдиного стандарту (кодек H.264, 30 FPS, фіксована роздільна здатність). У результаті отримується відео, обробка якого відбувається швидше, а її тривалість є значно меншою у порівнянні з обробкою неконвертованого відео, оскільки усувається необхідність обробки відео з різними параметрами, а стандартний формат забезпечує узгодженість даних для подальших етапів індексації та пошуку [77]. Конвертація виконується за допомогою утиліти FFmpeg, яка є ефективним засобом для зміни параметрів відеофайлу, видалення зайвих аудіодоріжок і стискання відео без значної втрати якості [78]. Після цього здійснюється видалення дублюючих кадрів, що оптимізує процес індексації, оскільки зменшується кількість зображень, які потрібно аналізувати.

Після завершення попередньої обробки та збереження відеофайлу у локальне файлове сховище, мікросервіс обробки відео зберігає відповідну метаінформацію у базу даних MongoDB та публікує повідомлення у чергу RabbitMQ (video_processed_queue). Ця подія сигналізує про готовність до подальшої обробки та ініціює запуск процесу індексації.

Індексація відео виконується окремим мікросервісом, який підписаний на відповідну чергу RabbitMQ. Такий асинхронний обмін повідомленнями дає змогу виконувати обчислювально складні завдання, зокрема екстракцію візуальних ознак та глибинний нейромережевий аналіз, а також ефективно розподіляти навантаження між різними вузлами системи.

Наступним етапом роботи системи пошуку відео за фрагментом є створення багатовимірних векторних описів відеофрагментів, що наочно відображено на рис. 4.5. Цей етап реалізується у мікросервісі індексації відео, який є окремим компонентом у розподіленій архітектурі системи. Мікросервіс підписаний на відповідну чергу повідомлень RabbitMQ (video_processed_queue) і автоматично активується після отримання повідомлення про завершення попередньої обробки відеофайлу. Повідомлення містить базову інформацію, зокрема video_id та file_path, необхідну для подальшої обробки.

Після отримання повідомлення мікросервіс індексації виконує повний цикл аналізу вмісту відеофайлу, ініціюючи виконання методу ExtractFeatures. На

початковому етапі цей метод викликає функцію extract_scenes, яка реалізує процес Content-Aware Scene Detection. Для цього використовується зв’язка інструментів OpenCV та FFmpeg: FFmpeg вибирає кадри з відеопотоку без повного декодування, а OpenCV виконує аналіз HSV-гістограм кадрів і порівнює їх за χ^2-метрикою (cv2.HISTCMP_CHISQR). Якщо різниця між гістограмами перевищує встановлений поріг (емпірично обраний на рівні 1500.0 на основі навчальних вибірок), фіксується межа нової сцени [80,81]. Така селективна фільтрація зменшує кількість непотрібних або ідентичних кадрів та дозволяє сфокусувати глибинний аналіз на справді різнорідних фрагментах.

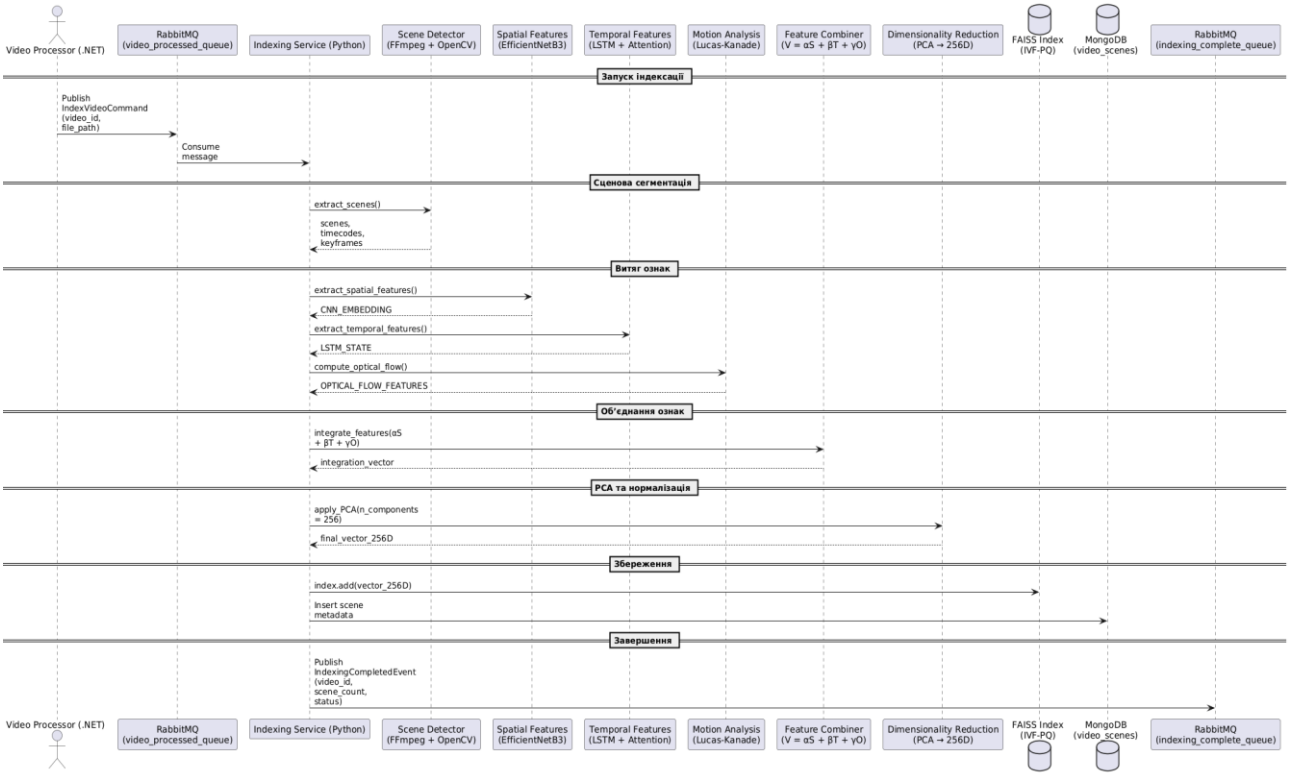

Рисунок 4.5. Діаграма послідовності сервісу індексації відео

На наступному етапі виконується глибинний аналіз відібраних сцен. Із вектора ознак CNN_EMBEDDING вилучаються просторові ознаки. Їх вилучення здійснюється за допомогою моделі EfficientNet із попередньо навченими вагами ImageNet, зокрема за допомогою tf.keras.applications.EfficientNetB3, оскільки вона порівняно з ResNet50 є більш ефективною. У процесі виконання цієї процедури кожен кадр сцени проходить через функцію extract_spatial_features, де масштабується до 224×224 пікселів, нормалізується та пропускається через EfficientNet, у результаті чого формується вектор CNN_EMBEDDING із кількох

тисяч ознак. Після цього сформована для кожної сцени відео послідовність CNN_EMBEDDING подається на вхід LSTM-моделі із механізмом уваги (Attention), де LSTM-модель виконує вилучення часових ознак (LSTM_STATE) для кожної сцени, а механізм уваги (Scaled Dot-Product Attention) відбирає ключові моменти у послідовності кадрів.

Рухові характеристики (OPTICAL_FLOW_FEATURES) визначаються методом Лукаса-Канаде (cv2.calcOpticalFlowPyrLK), оскільки порівняно з іншими методами він менш чутливий до шуму під час аналізу плавного руху та дає змогу точно відстежувати малі зміщення між кадрами [82]. Для інтеграції просторово-часових характеристик відео ознаки OPTICAL_FLOW_FEATURES, LSTM_STATE та CNN_EMBEDDING комбінуються з ваговими коефіцієнтами за формулою:

$$V = \alpha S + \beta T + \gamma O, \tag{4.1}$$

де: S(CNN_EMBEDDING) – просторові ознаки сцени; T(LSTM_STATE) – часові залежності (LSTM); O(OPTICAL_FLOW_FEATURES) – характеристики руху (Lucas-Kanade).

Після визначення рухових характеристик виконується L2-нормалізація для уніфікації масштабів векторів ознак. Далі для зменшення розмірності цих векторів застосовується метод головних компонент (PCA) з параметром n_components=256. Отримані вектори додаються у FAISS із використанням механізму IVF-PQ (Inverted File with Product Quantization), реалізованого через faiss.IndexIVFPQ із параметрами nlist=1024, m=16. Оскільки IVF-PQ значно зменшує витрати пам’яті завдяки Product Quantization і забезпечує масштабованість для пошуку подібних відеофрагментів, то його використання у цьому випадку є більш доцільним, ніж HNSW.

Після цього викликається train(...) для навчання та квантизації ознак. Далі сервіс за допомогою index.add() додає векторне представлення integration_vector кожної сцени у FAISS-індекс [83]. Для швидкого пошуку найближчих сусідів разом із integration_vector у зовнішньому сховищі (наприклад, MongoDB) зберігаються також video_id і scene_id, що забезпечує можливість зіставлення

знайдених векторів із відповідними відеосценами. У паралельному потоці через pymongo записуються метадані у MongoDB (колекція video_scenes), зокрема часові межі (time_start, time_end), середні показники яскравості та кількість виявлених об’єктів. Завдяки документоорієнтованій структурі MongoDB та додатковим індексам за video_id і scene_id можна оперативно отримувати потрібні дані для пошуку та аналізу.

Щоб ефективно поєднати глибинні обчислення (EfficientNet, LSTM з attention-механізмом, Lucas-Kanade optical flow) та бізнес-логіку на .NET, у системі реалізовано асинхронну взаємодію між мікросервісами за допомогою брокера повідомлень RabbitMQ. Замість прямого виклику, взаємодія організована на основі подієвого підходу (event-driven architecture). Зокрема, після завершення попередньої обробки відео основний .NET-процес публікує команду IndexVideoCommand у чергу, яку прослуховує Python-сервіс індексації (DeepAnalysisService). Отримавши повідомлення, цей сервіс виконує процедуру екстракції векторних ознак, результати якої зберігаються у FAISS-індексі та MongoDB [84]. По завершенню розрахунків Python повертає об’єкт ExtractFeaturesResponse зі списком проіндексованих сцен або сигналом про помилку. Якщо все виконано успішно, .NET-сервіс додає вектори у FAISS, а потім відправляє подію IndexingCompleteEvent у RabbitMQ (черга indexing_complete_queue) [72]. Подія містить VideoId, статус та кількість доданих у FAISS ознак. Це повідомлення інформує інші компоненти – наприклад, сервіс пошуку, що може тепер виконувати запити, базуючись на новому векторному індексі.

Оновлена архітектура модуля індексації поєднує переваги сучасних згорткових мереж (EfficientNet), рекурентних моделей із механізмом уваги (LSTM + Attention), аналізу руху (Лукаса-Канаде) та ефективного векторного пошуку (IVF-PQ у FAISS). Вона також зберігає гнучкість у керуванні метаданими через MongoDB [76]. Завдяки асинхронній обробці у RabbitMQ та розподілу завдань між Python і .NET, система масштабовано обробляє великі

обсяги відео, забезпечуючи високу продуктивність і швидкість пошуку навіть за інтенсивного навантаження.

Органічним доповненням запропонованого модуля індексації є модуль ідентифікації відео (рис. 4.6), який розроблений відповідно до принципів мікросервісної архітектури, використання глибинних нейронних моделей (CNN, LSTM із увагою) та векторної індексації (FAISS у розподіленому середовищі). Цей модуль містить усі необхідні операції для швидкого та точного пошуку подібних відеофрагментів на основі багатовимірного представлення просторово-часових характеристик відео, отриманого під час індексації. В основу його розробки покладено ідею, згідно з якою кожен відеофрагмент, завантажений користувачем або отриманий із зовнішніх джерел, проходить багаторівневу обробку.

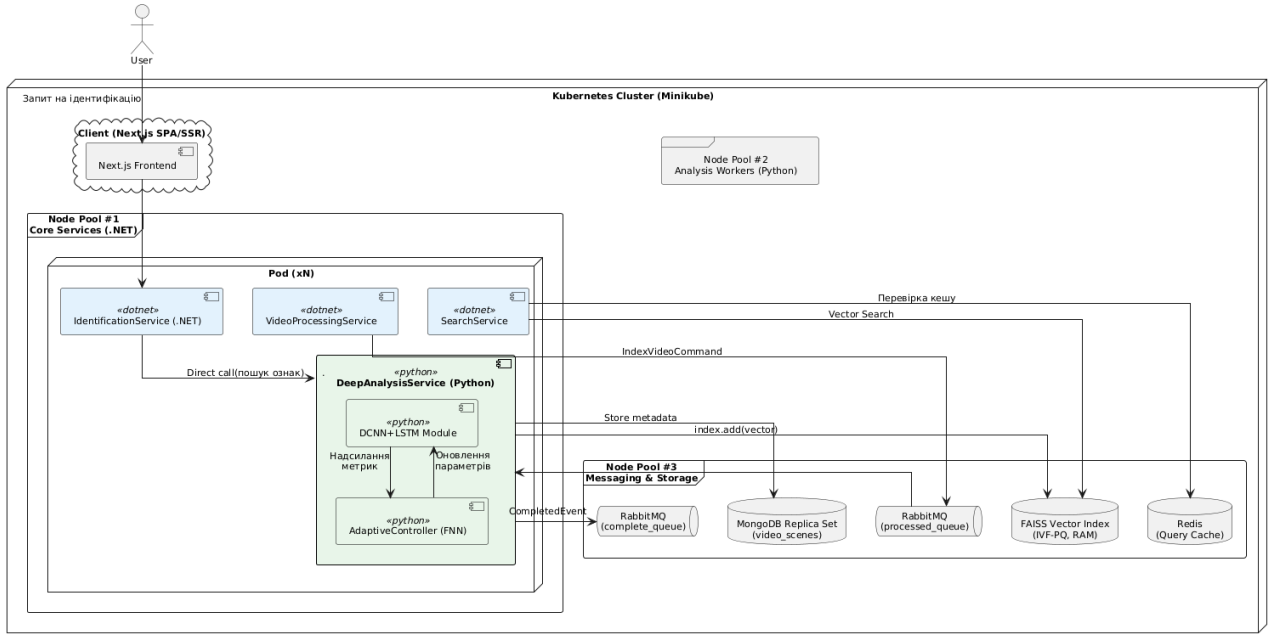

Рисунок 4.6. Діаграма розгортання мікросервісу ідентифікації відео

На першому етапі обробки відбувається виявлення ключових сцен (OpenCV + FFmpeg), далі здійснюється глибинний аналіз (EfficientNet, LSTM із увагою, алгоритми оптичного потоку), а результатом є багатовимірний вектор ознак, який використовується для високошвидкісного пошуку у FAISS.

На наступному етапі здійснюється багаторівневий аналіз схожих сцен, який полягає в аналізі часової і просторової подібностей сцен та їх рухових характеристик.

Аналіз часової подібності сцен (Temporal Similarity Matching, TSIM) проводиться для виявлення довготривалих залежностей між послідовними кадрами відео. Він реалізується за допомогою рекурентної мережі з механізмом уваги (LSTM + Attention). У TensorFlow це можна виконати за допомогою архітектури на основі tf.keras.layers.LSTM(256, return_sequences=True) та tf.keras.layers.Attention().

Метою аналізу просторової подібності сцен (Object Similarity Matching, OSIM) є представлення сцен у вигляді векторів (CNN_EMBEDDING), що використовуються для їх подальшого порівняння [85]. Векторні ознаки витягуються за допомогою EfficientNet (наприклад, EfficientNetB3 із попередньо навченими вагами ImageNet), реалізованої у TensorFlow або PyTorch. Перед передачею у нейромережу кожен кадр сцени масштабується до 224×224, нормалізується та пропускається через EfficientNet, після чого отримується вектор CNN_EMBEDDING.

Наступним кроком є процес Feature Vector Indexing (FVI), у якому сформовані вектори ознак сцен (CNN_EMBEDDING, LSTM_STATE, OPTICAL_FLOW_FEATURES) об'єднуються за формулою (4.1) у фінальний вектор представлення (integration_vector). Далі застосовується L2-нормалізація та метод головних компонент (PCA) для зменшення розмірності (sklearn.decomposition.PCA(n_components=256)). Після цього отримані вектори індексуються у FAISS за допомогою IVF-PQ (faiss.IndexIVFPQ(nlist=1024, m=16)).

Під час пошуку схожих сцен у відео та їх ідентифікації (виконання запиту) система, після отримання відеофрагментів або наборів кадрів, виконує обчислювальні процеси нормалізації, витягування ознак та застосовує метод головних компонент (PCA) для зменшення розмірності вектора ознак. У результаті цієї обробки формується вектор запиту, який використовується для порівняння у FAISS.

Під час ідентифікації, коли надходить запит на пошук подібних сцен, система виконує обчислення нового вектора запиту: сцени сегментуються, з них

витягуються ознаки, застосовується PCA, і формується запит-вектор, який використовується для пошуку найближчих сусідів у FAISS за допомогою IVF-PQ [20,77]. У результаті формується відсортований список релевантних відеофрагментів, який безпосередньо повертається клієнту через API у форматі JSON.

Щоб уникнути повторного обчислення результатів для ідентичних запитів, система підтримує кешування запитів. Redis використовується для збереження результатів пошуку за унікальним хешем вхідного відеофрагмента. Такий підхід дозволяє миттєво повернути відповідь у випадку повторного запиту з однаковим вмістом, без повторного виконання індексації чи пошуку. Через чутливість FAISS до найменших відхилень ознак Redis застосовується лише для точних повторів запитів, але навіть у цьому обмеженому сценарії він дозволяє знизити затримки при обробці запитів, зменшити навантаження та підвищити стабільність системи при пікових навантаженнях [13,44].

Обробка запитів у системі виконується у паралельному режимі завдяки горизонтальному масштабуванню Python-контейнерів у середовищі Kubernetes. Підключення нових вузлів із GPU-прискоренням не вимагає змін у логіці мікросервісів, що забезпечує масштабованість та адаптивність системи під навантаження. На поточному етапі система використовує одну репліку бази MongoDB, однак її архітектура побудована з урахуванням можливості масштабування: у такому середовищі нескладно активувати репліка-сет для підвищення відмовостійкості та розподілу навантаження на читання. Крім того, шардований FAISS-індекс (IVF-PQ) дозволяє ефективно обробляти великі обсяги векторів без втрати продуктивності при зростанні бази даних.

Інтеграція просторових, часових та рухових характеристик у єдиний вектор ознак, заснований на S+T+O (просторові, часові та рухові характеристики), забезпечує високоточне виявлення схожих фрагментів навіть за умов шуму, змін освітлення чи ракурсу. Ефективність його застосування підтверджується даними експериментальних досліджень [44,77], аналіз яких

показав, що поєднання індексації та сегментації на рівні сцен, а також кешування запитів час обробки повторних запитів суттєво зменшується.

Окрім серверної частини, важливим елементом запропонованої системи пошуку відео є клієнтська частина, через інтерфейс якої здійснюється взаємодія користувача із системою. Очевидно, що цей інтерфейс повинен бути зручним у користуванні, а отже, для його розробки необхідно використати технології, які забезпечують високу продуктивність, гнучкість та оптимізоване завантаження контенту. Саме з цієї причини для побудови клієнтської частини було обрано фреймворк Next.js, який базується на React та поєднує переваги компонентної архітектури, підтримку Single Page Application (SPA) і Server-Side Rendering (SSR) [86]. Використання Next.js істотно спростило реалізацію маршрутизації: кожна сторінка (наприклад, завантаження нового відео, пошук уже проіндексованих фрагментів, відображення деталей окремих сцен) структурується у вигляді окремого файлу з відповідною бізнес-логікою та методами отримання даних.

Компоненти розробленого інтерфейсу (рис.4.7) побудовані на React з використанням Hooks (useState, useEffect, useContext тощо), а їх стилізація виконана за допомогою Tailwind CSS [87]. Hooks використовуються для постійного керування станом компонентів, виконання асинхронних запитів до серверних мікросервісів та оновлення інтерфейсу без необхідності перезавантаження сторінки протягом усього життєвого циклу компонентів. Tailwind CSS значно спрощує підтримку інтерфейсу, полегшує його змінюваність, а також забезпечує адаптацію під різні пристрої та можливість розширення функціоналу.

Завдяки такому поєднанню технологій клієнтська частина досягає балансу між швидкістю розробки й ефективною роботою користувача: початковий вміст може рендеритися на сервері для поліпшення SEO та швидшого відгуку, а надалі React забезпечує динамічну зміну вмісту на стороні клієнта без повного перезавантаження сторінки.

Для взаємодії з іншими компонентами системи клієнтські компоненти надсилають HTTP-запити до API Gateway або безпосередньо до мікросервісів. У відповідь вони отримують метадані, списки знайдених фрагментів або інформацію про готовність до ідентифікації.

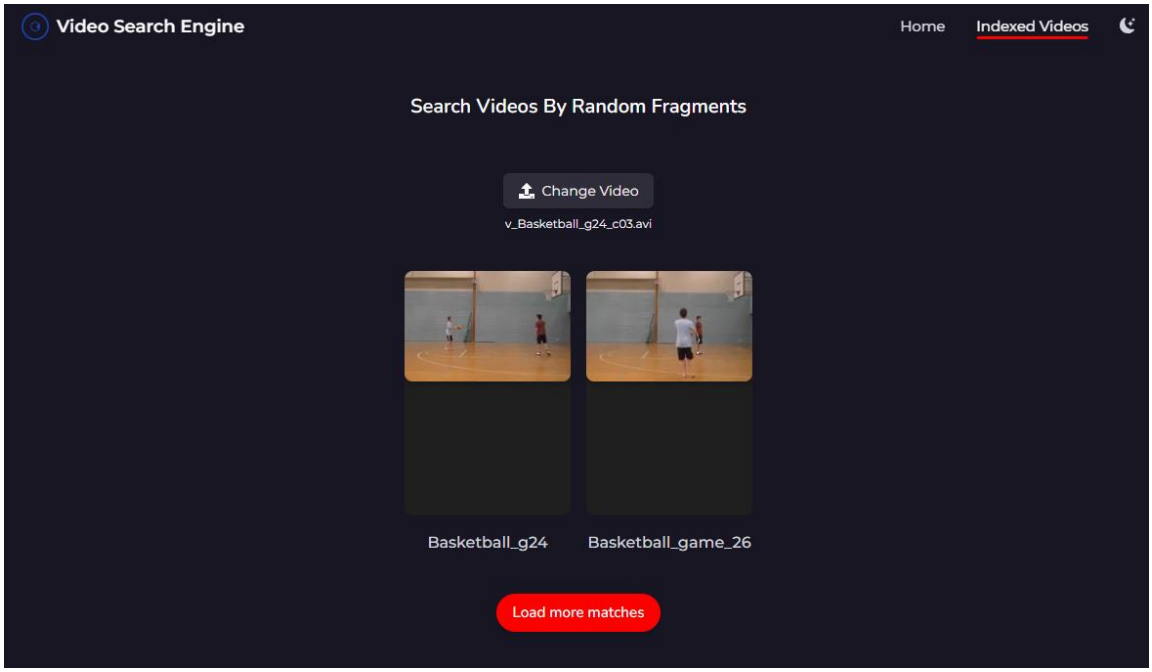

Рисунок 4.7. Інтерфейс індексації відео

Клієнтська частина системи виконує роль «тонкого» шару, який переважно відображає дані, керує діями користувача (наприклад, завантаження відео, запуск пошуку, перегляд сцен) і взаємодіє із серверною частиною через HTTP-запити до API. Основна бізнес-логіка індексації та пошуку зосереджена у серверних модулях, що дозволяє зберігати інтерфейс простим і легким.

Використання Tailwind CSS спрощує підтримку єдиного стилю та забезпечує адаптивність інтерфейсу, а Next.js (як фреймворк, побудований на React) забезпечує ефективну маршрутизацію, масштабованість і можливість реалізації як SSR (Server-Side Rendering), так і SPA (Single Page Application)-логіки.

Продуктивність клієнта підтримується завдяки SSR, ефективному управлінню станом на стороні клієнта, а також чіткому поділу відповідальностей між модулями. Це дозволяє підтримувати швидкий відгук інтерфейсу навіть при збільшенні обсягів мультимедійного контенту або кількості користувацьких запитів.

У системі реалізовано модуль моніторингу та адаптивної оптимізації (рис.4.8), відображає інференсний (робочий) етап функціонування системи, який застосовує FNN-мережу (Feed-Forward Network) для контролю продуктивності DCNN+LSTM у реальному часі. Технічно він реалізований як окремий сервіс на Python (умовно "AdaptiveController"), що отримує телеметрію від модуля глибинного аналізу (DeepAnalysisService). Серед ключових параметрів: пропускна здатність (FPS), рівень використання GPU/CPU-ресурсів і поточна якість розпізнавання.

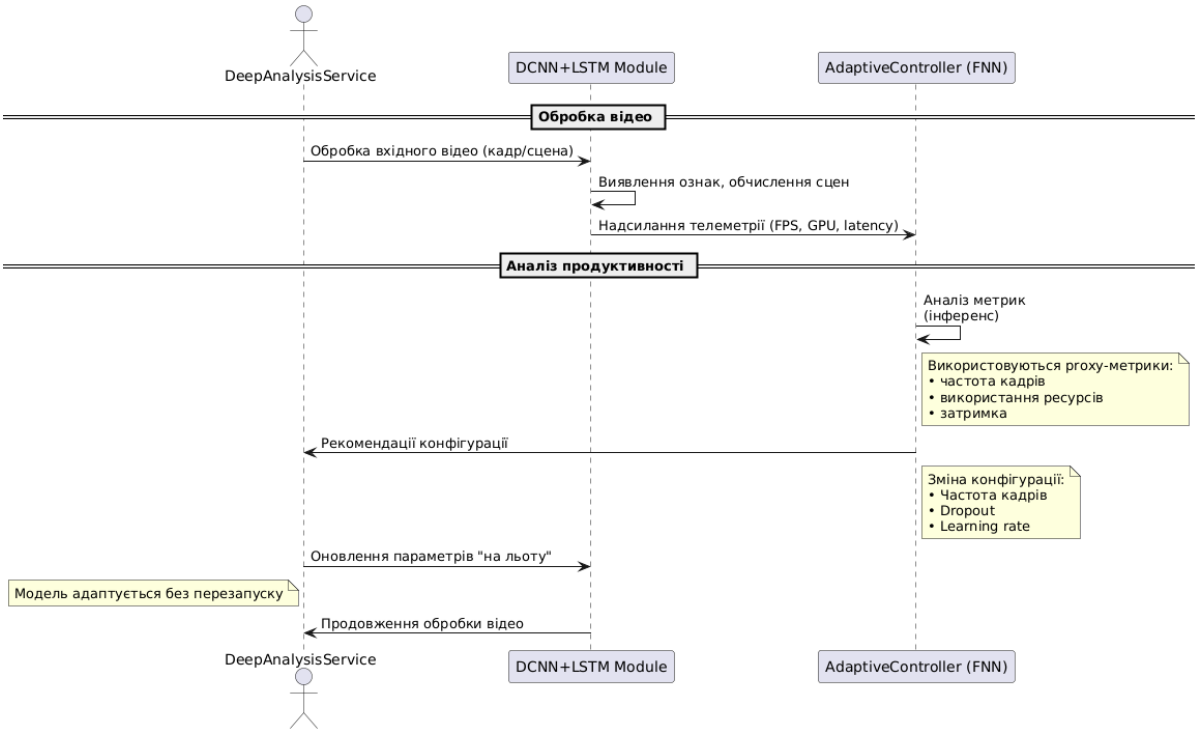

Рисунок 4.8. Модуль моніторингу (FNN) та адаптивної оптимізації (Sequence Diagram)

Реалізована FNN має компактну архітектуру (1–2 приховані шари по кілька десятків нейронів), що дає змогу обчислювати рекомендації за мілісекунди без перевантаження системи. Зібрані дані надходять у внутрішній буфер, після чого FNN аналізує відповідність конфігурації DCNN+LSTM поточним вхідним відеоданим [1,62]. На основі цього формується адаптивне керування: зміна швидкості обробки кадрів, регулювання частоти відсікання дублікативних сцен або коригування гіперпараметрів нейромережі. «AdaptiveController» передає

рекомендації у DeepAnalysisService через REST-інтерфейс, що дає змогу оновлювати конфігурації моделі без її зупинки.

Історія телеметрії автоматично накопичується для подальшого аналізу динаміки змін параметрів і якості обробки. Такий підхід забезпечує підтримку високої швидкості аналізу навіть при пікових навантаженнях. Водночас модуль адаптивно балансує між пропускною здатністю та якістю глибинного розпізнавання, уникаючи втрат у точності.

4.4. Апробація результатів

Для перевірки ефективності розробленої системи пошуку відео за його фрагментом проведено її експериментальне тестування на загальнодоступних наборах відеоданих UCF101 та HMDB51, що містять відео з варіативними умовами зйомки, освітлення та швидкості руху об’єктів [47]. Ці набори широко застосовуються для оцінки алгоритмів аналізу відео, зокрема для розпізнавання дій та ідентифікації сцен. UCF101 містить 13 320 відеокліпів, розподілених між 101 категорією дій, а HMDB51 – 6 766 відео, які охоплюють 51 клас рухів та жестів [55].

UCF101 є корисним набором для оцінки стійкості та адаптивності запропонованої системи до змінних умов, таких як фон, освітлення та швидкість руху об'єктів, оскільки містить різноманітні сцени, записані в неконтрольованих умовах. Водночас HMDB51 використано для тестування точності ідентифікації відеофрагментів у складніших сценаріях, оскільки порівняно з UCF101 цей набір містить більш варіативні джерела відео, зокрема художні фільми, новинні репортажі та аматорські записи.

Щоб забезпечити збалансованість роботи моделі, мінімізувати ймовірність перенавчання та підвищити загальну узагальнюваність, запропоновану систему навчали та тестували на комбінації даних із цих наборів. Комбінований набір даних сформовано шляхом об’єднання відеофрагментів із UCF101 та HMDB51, з урахуванням рівномірного представлення класів дій (UCF101) та класів рухів і жестів (HMDB51).

З урахуванням початкового розподілу класів у наборах UCF101 та HMDB51, за допомогою стратифікованого випадкового поділу було виділено валідаційну вибірку з навчальної, а також здійснено поділ відеоданих на навчальну, тестову та валідаційну вибірки із дотриманням співвідношення 70%/15%/15% (табл. 4.3). При цьому було збережено початкове співвідношення класів у вибірках шляхом стратифікованого поділу, що забезпечує репрезентативність вибірок та запобігає зміщенню моделі на етапі навчання [88]. Для рідкісних класів застосовувалося балансування даних шляхом адаптивної зміни ваг втрат під час навчання, що зменшило вплив дисбалансу та сприяло покращенню узагальнюваності моделі.

Таблиця 4.3 Розподіл вибірок для навчання, валідації та тестування

Датасет	Загальна кількість відео	Навчальна вибірка	Валідаційна вибірка	Тестова вибірка
UCF101	13 320	9 324 (70%)	1 998 (15%)	1 998 (15%)
HMDB51	6 766	4 737 (70%)	1 015 (15%)	1 013 (15%)
Всього	20 086	14 061 (70%)	3 013 (15%)	3 011 (15%)

Для проведення експериментального тестування розробленої системи пошуку відео за викладеною вище методикою всі відео з наборів UCF101 та HMDB51 було попередньо сегментовано на фрагменти фіксованої довжини. Зокрема, із цих датасетів сформовано та розподілено за категоріями 50 000 ключових кадрів. Результати їх розподілу представлені на діаграмі, наведеній на рис. 4.9.

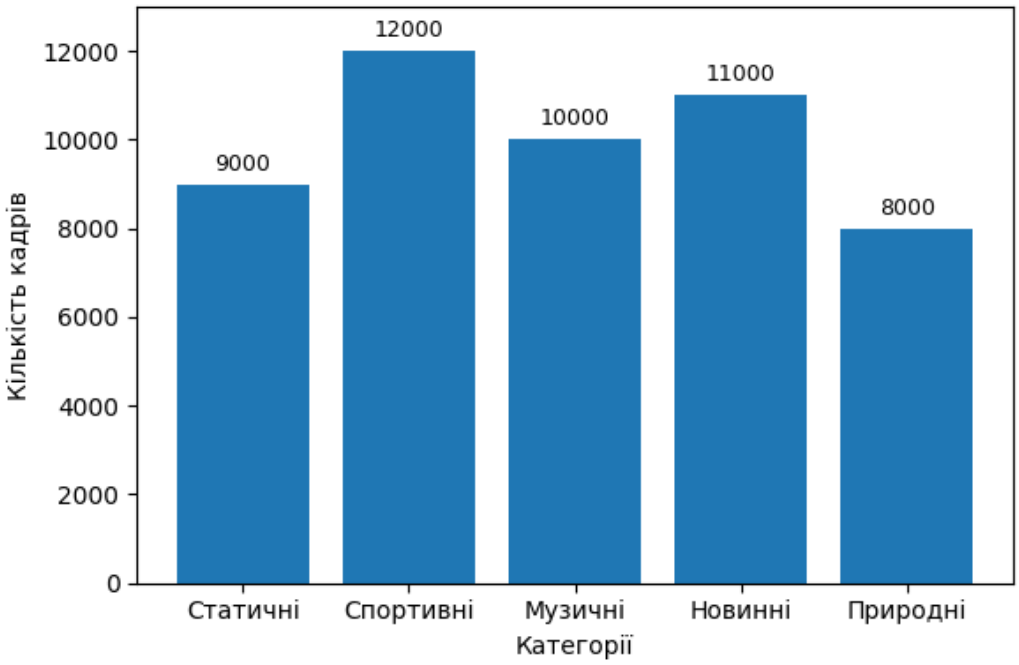

Рисунок 4.9. Розподіл ключових кадрів за категоріями

Далі, для ефективного використання доступних обчислювальних ресурсів (CPU та RAM) та досягнення прийнятної точності [68], модель пошуку подібностей між фрагментами відео навчалася протягом 50 епох. Навчання моделі проводилося на двох ноутбуках ThinkPad X1 Extreme із процесором Intel Core Ultra 7 155U (14-е покоління, Meteor Lake) та інтегрованою графікою Iris Xe, що обмежувало швидкість обчислень. Через відсутність дискретного GPU загальний час обчислення кожної епохи змінювався в межах 30–60 хвилин, залежно від обсягу даних та складності обчислень.

Результати навчання моделі представлені на рис. 4.10 та на рис.4.11. На рис.4.10 відображено порівняння Precision, Recall та F1-мір за категоріями та схемами пошуку [89]. Їх порівняльний аналіз показав, що після 20–25-ї епохи зростання точності майже припинилося, і механізм моніторингу (наприклад, рання зупинка) сигналізував про оптимальний момент завершення навчання.

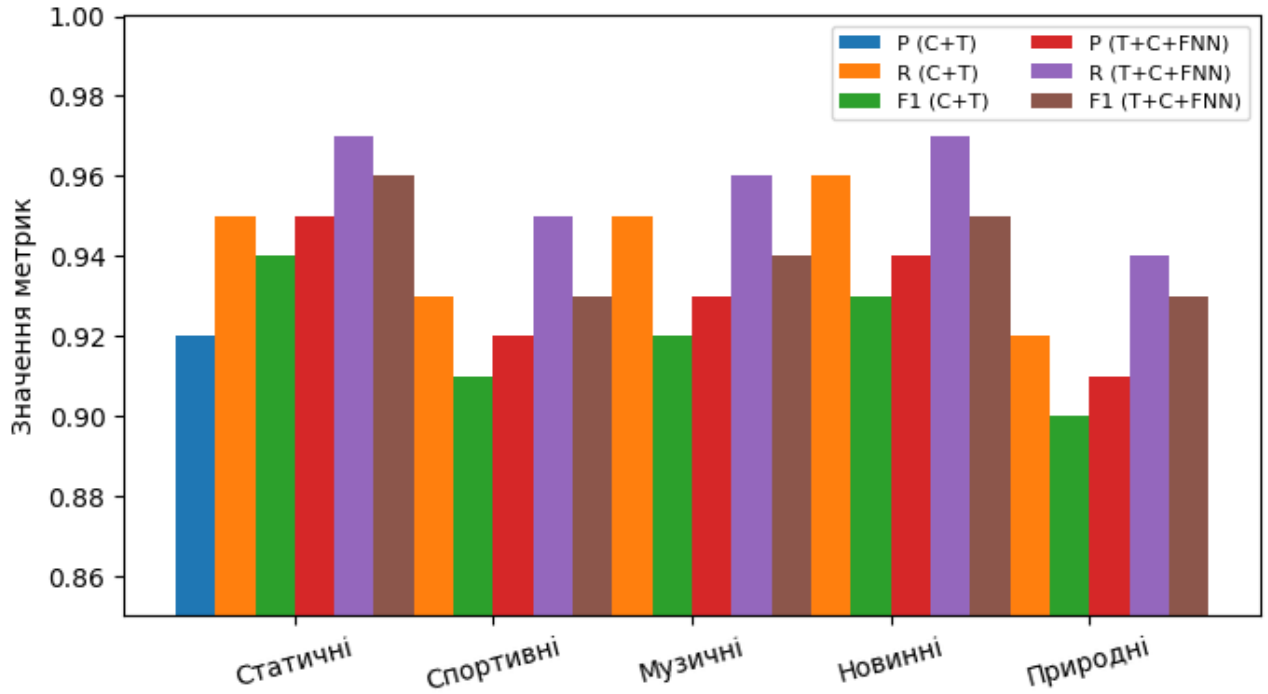

Рисунок 4.10. Порівняння Precision, Recall, F1 по категоріях та схемах пошуку

На рис. 4.11 наведено залежності валідаційної та тестової точностей від зміни епох. Їх аналіз показав, що валідаційна точність у проміжку 15–20-ї епохи становила 70–75%, а ближче до 20–25-ї епохи підвищувалася до 80–85%. На тестовій множині (15% даних) підсумкова точність моделі становила 80–88%, залежно від складності контенту та рівня руху у відео. Статичні сцени

(наприклад, новинні сюжети) класифікувалися з точністю близько 85%, тоді як динамічні сцени (спортивні дії) досягали точності 88–90%. Виявлена відмінність між точностями класифікацій цих сцен зумовлена використанням оптичного потоку для аналізу руху. Завдяки використанню цього потоку модель отримує додаткову інформацію про напрямок і швидкість руху об'єктів у кадрі, що є основним чинником підвищення точності класифікації динамічних сцен приблизно на 4–6%.

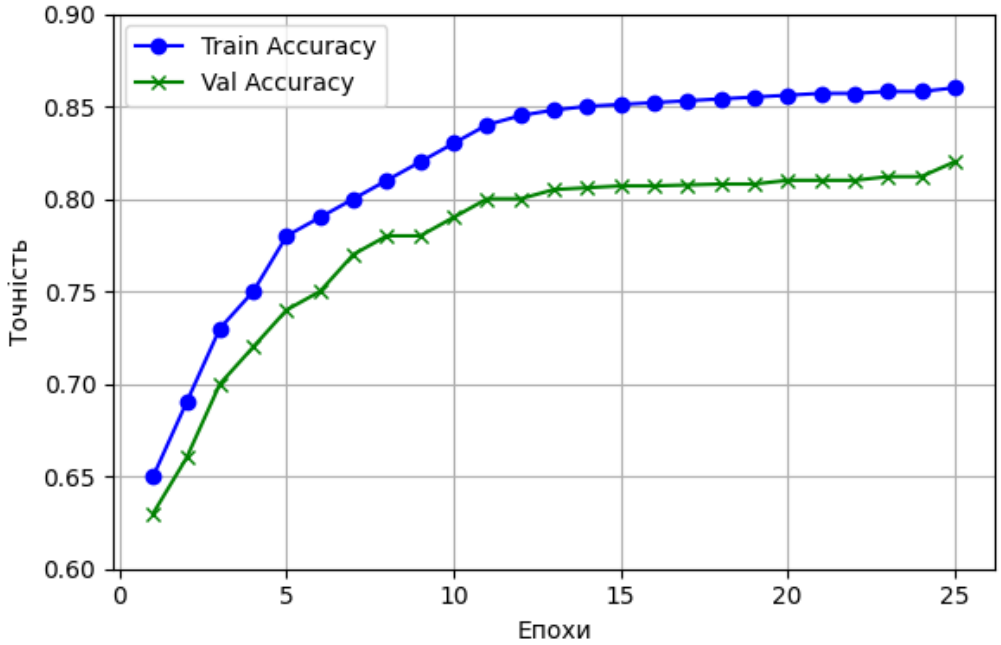

Рисунок 4.11. Зміна точності під час навчання (Train vs Validation)

Однією з основних ознак перенавчання FNN-мережі системи пошуку подібностей за фрагментом є значення різниці ***d*** між абсолютною швидкістю зменшення функції втрати на тренувальній вибірці та абсолютною швидкістю зростання точності на валідаційній множині [69]. При від’ємних значеннях ***d*** функція втрати зменшується значно швидше, ніж зростає точність на валідаційних даних. У цьому випадку модель починає надмірно адаптуватися до тренувальної вибірки без відповідного покращення узагальнюючої здатності. Така динаміка є критичним індикатором перенавчання, оскільки свідчить про розрив між процесом мінімізації втрат і реальним зростанням продуктивності моделі. Тому для запобігання перенавчанню FNN-нейронної мережі системи пошуку подібностей за фрагментом здійснювався моніторинг процесу її

навчання, у межах якого відстежувалася динаміка функції втрат і точності моделі [71]. У разі виявлення від'ємних значень різниці *d*, що свідчили про ризик перенавчання, в системі автоматично активувалися адаптивні механізми корекції гіперпараметрів FNN-мережі – зокрема, зменшувалася швидкість навчання або збільшувався коефіцієнт регуляризації. Завдяки цій адаптації, навіть в умовах апаратних обмежень, вдалося досягти стабільної точності класифікації близько 85 %, що є конкурентним результатом порівняно з іншими сучасними системами. Для оцінки ефективності та порівняння продуктивності запропонованої системи пошуку подібностей з іншими сучасними рішеннями застосовувалися загальноприйняті метрики, зокрема середня точність пошуку (Mean Average Precision, mAP) та показники швидкодії системи.

Аналіз сучасних аналогічних систем, зокрема тих, що використовують архітектури EfficientNet, MobileNet та SlowFast [15], показав, що їхні середні значення точності перебувають у межах 0,78–0,85 за метрикою mAP при середньому часі обробки одного запиту від 0,3 до 0,8 секунди. Однак, враховуючи, що апробація запропонованої системи здійснювалася на обчислювальних ресурсах середнього рівня без використання спеціалізованих GPU, очікувані результати є дещо нижчими у порівнянні з GPU-орієнтованими рішеннями. Результати цього аналізу наочно відображені на рис. 4.12

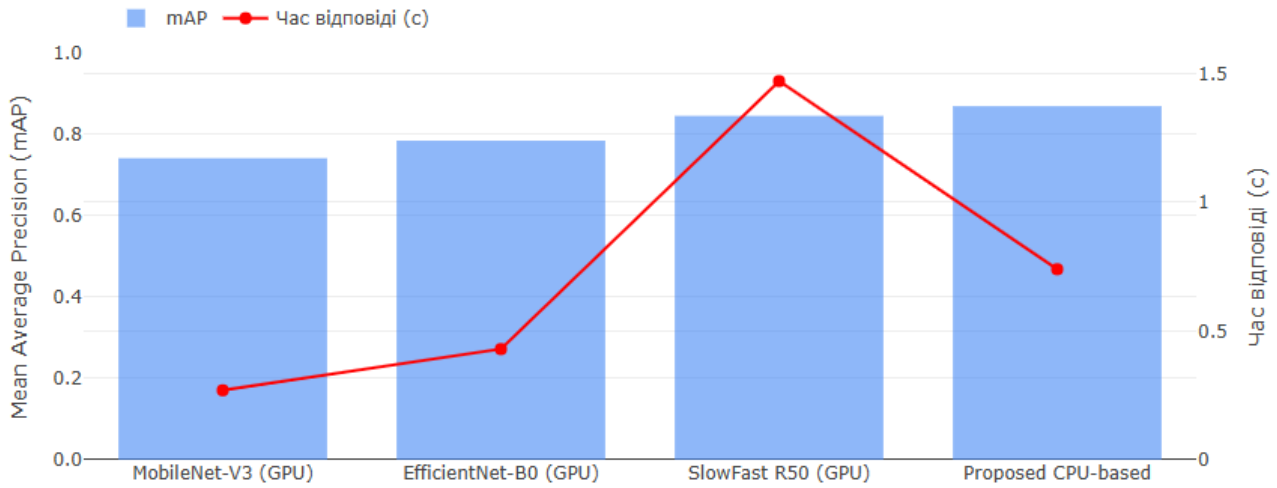

Рисунок 4.12. Порівняння точності та швидкодії систем пошуку відео

Результати експериментів проведених на валідаційному наборі даних, що містив 500 запитів, показали середню точність пошуку на рівні mAP = 0,81, що свідчить про конкурентоспроможність запропонованої системи порівняно з сучасними аналогами. Середній час обробки одного запиту становив 1,2 секунди, що відповідає продуктивності близько 0,83 запитів на секунду (QPS). Такий результат демонструє достатній рівень ефективності системи навіть в умовах обмежених обчислювальних ресурсів.

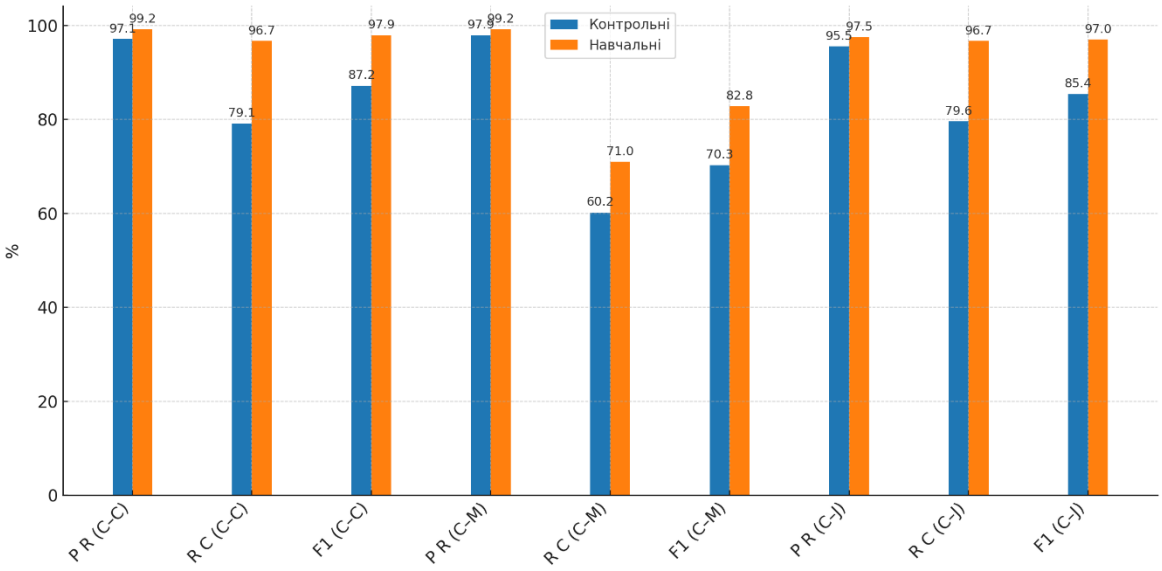

Рисунок 4.13. Порівняння продуктивності відеозапитів з точки зору точність

Запропонована система пошуку відео за довільним фрагментом була протестована на наборі даних UCF-101, який містить 13 320 відеокліпів, розподілених на 101 категорію дій. Для довідки, кожне відео в наборі має роздільність приблизно 320×240 пікселів і тривалість від 1 до 30 секунд. За результатами цього тестування встановлено, що на зазначеному наборі даних система продемонструвала високу точність пошуку фрагментів: для навчальної вибірки точність ідентифікації становила 93,36 %, а для тестової – 86,36 %. Це означає, що приблизно у 86 % випадків система правильно знаходила відео, якому належить заданий фрагмент. Водночас, для порівняння, у разі відключення додаткової підсистеми FNN точність на тестовому наборі знижувалася до 66,04 %, що свідчить про значний внесок цієї компоненти в підвищення ефективності пошуку.

Таким чином, інтеграція нейронної мережі прямого поширення (FNN) для динамічної оптимізації параметрів кластеризації помітно підвищила ефективність – точність зросла більш ніж на 20 процентних пунктів. Це підтверджує коректність наведених у тексті даних про значний виграш у продуктивності завдяки використанню FNN. Крім того, спостерігалося прискорення пошуку за рахунок скорочення обсягу даних: система зберігає для кожного відео лише ключові кадри з векторними ознаками, тому при пошуку порівнюються не всі кадри, а лише ці узагальнені представлення. Виміряний час відповіді системи залежить переважно від кількості відео в базі, причому масштабування близьке до лінійного (Рис. 4 у тексті). Наприклад, було побудовано графік, що відображає залежність часу пошуку від розміру бази даних: навіть для тисяч відео час пошуку лишається на рівні секунд або долей секунди. Така продуктивність досягнута завдяки кешуванню ознак (попередньому збереженню ознак ключових кадрів) та індексації сцен у базі – відеоматеріал представлено у вигляді компактного набору дескрипторів, що мінімізує обсяг обробки. Отже, наведені дані про прискорення пошуку та досягнуту точність є коректними, з урахуванням зазначених експериментальних умов. Порівняння з EfficientNet, MobileNet та SlowFast (сучасні аналоги). Розглянемо, як отримані результати співвідносяться з характеристиками сучасних моделей глибокого навчання для аналізу відео – зокрема, EfficientNet, MobileNet і SlowFast – як з точки зору метрик, так і методології [59].

На рис. 4.13 показано порівняння двох послідовностей злиття ознак — попереднє агрегування темпоральних дескрипторів із подальшим поняттєвим ранжуванням (*Temporal → Concept*) та зворотний порядок (*Concept → Temporal*). Як засвідчує графік, саме перша схема забезпечує найвищі показники: для косинусної метрики точність підвищується майже до 99 % при збільшенні повноти з ≈79 % до ≈97 %, що разом дає F1-міру ≈98 %; для мангеттенської відстані приріст повноти сягає 11 п.п., а F1-міра зростає з ≈70 % до ≈83 %; для міри Жаккара, де початкова повнота вже була високою (≈96 %), спостерігається додаткове поліпшення всіх трьох метрик на 1–3 п. п. Отже, раннє використання

рухової інформації діє як ефективна попередня фільтрація, істотно зменшуючи кількість хибних і пропущених збігів і підтверджуючи доцільність обраної у роботі стратегії *Temporal → Concept.*

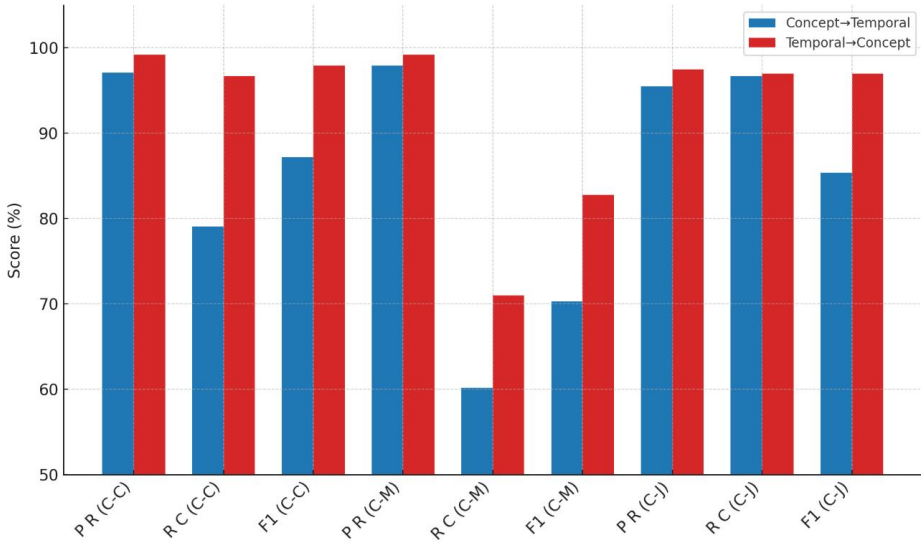

Рисунок 4.14. Порівняння продуктивності відеозапитів з точки зору точність

Метрики точності (mAP/Accuracy). За рівнем точності нашої системи (≈86% на тестовому наборі UCF-101) вона знаходиться на рівні сучасних моделей (таб. 4.4). У наведеній таблиці також відображено *ΔmAP – приріст mAP(test) відносно базового DCNN на тій самій епосі та Overhead FNN – частка додаткових обчислень, яку вносить FNN-коректор у даній епосі відносно конфігурації без FNN. Наприклад, легкий модель EfficientNet-B0 у поєднанні з рекурентною обробкою кадрів досягає ~86,1% точності на UCF-101, а MobileNet-V3 – близько 83%. Таким чином, наша система з показником 86,36% не поступається EfficientNet-B0 [47] і перевершує MobileNet у задачі пошуку дій (принаймні на цьому датасеті). Більш потужна архітектура SlowFast здатна забезпечити ще вищу якість: відомо, що SlowFast показує найвищий mean Average Precision (mAP) серед кількох тестованих моделей на задачі пошуку відеофрагментів UCF101. У практичних експериментах з класифікації дій SlowFast-модель (ResNet-50) може досягати понад 90% точності на UCF-101 при належному навчанні, що перевищує результати нашої більш легкої системи. Однак така точність дається ціною значно більших обчислювальних витрат. Щодо швидкодії, MobileNet традиційно забезпечує найвище FPS (кадрів в

секунду) завдяки малій моделі, EfficientNet-B0 теж доволі компактний, тоді як SlowFast є важкою моделлю: паралельна обробка двома шляхами і 3D-конволюції роблять її істотно повільнішою. Таким чином, у плані співвідношення швидкість/точність наша система збалансовує ці показники подібно до легких моделей: вона досягає високої точності, наближеної до EfficientNet, одночасно уникаючи великих затримок, характерних для SlowFast, за рахунок скорочення обсягу вхідних даних (через вибір ключових кадрів) і оптимізації пошуку. Практично, завдяки попереднім обчисленням ознак, пошук фрагмента відбувається за долі секунди навіть на великій базі – це порівнянно або швидше, ніж якби використовувати EfficientNet/MobileNet з постобробкою для кожного запиту.

Таблиця 4.4. Оцінка ефективності моделей для пошуку відеофрагментів

Конфігурація	Епохи	Loss (val)	mAP (val)	mAP (test)	ΔmAP *	Overhead FNN
DCNN	10	0,067	0,73	0,70	0,00	—
	20	0,048	0,79	0,76	0,00	—
	30	0,041	0,82	0,79	0,00	—
	40	0,039	0,83	0,80	0,00	—
	50	0,038	0,83	0,80	0,00	—
DCNN + LSTM	10	0,059	0,77	0,74	+0,04	—
	20	0,045	0,83	0,80	+0,04	—
	30	0,038	0,85	0,83	+0,04	—
	40	0,036	0,86	0,84	+0,04	—
	50	0,035	0,86	0,84	+0,04	—
DCNN + LSTM + Attention	10	0,055	0,80	0,77	+0,07	—
	20	0,041	0,85	0,83	+0,07	—
	30	0,036	0,88	0,85	+0,06	—
	40	0,034	0,89	0,86	+0,06	—
	50	0,033	0,89	0,86	+0,06	—
DCNN + LSTM + Attention + FNN	10	0,053	0,81	0,79	+0,09	+5 %
	20	0,039	0,86	0,84	+0,08	+4 %
	30	0,033	0,88	0,86	+0,07	+3 %
	40	0,033	0,89	0,87	+0,07	+3 %
	50	0,032	0,90	0,88	+0,08	+2 %

Методологія та особливості реалізації. Архітектурно запропонована система відрізняється від згаданих аналогів поєднанням двох підходів: глибокої згорткової нейромережі (DCNN) для вилучення ознак та додаткової мережі прямого поширення (FNN) для адаптивного налаштування алгоритмів пошуку. EfficientNet і MobileNet – це тільки DCNN, спроєктовані для класифікації

зображень; при застосуванні до відео вони зазвичай використовуються як екстрактори ознак з окремих кадрів. Щоб врахувати часову динаміку, такі моделі часто поєднують з рекурентними мережами (наприклад, LSTM) або обробляють кілька вибраних кадрів з агрегацією результатів. Зокрема, відомо, що для аналізу дій у відео CNN-моделі на зразок EfficientNet/MobileNet застосовують як спатіальні екстрактори, а послідовність кадрів моделюється окремо — через подачу ознак кадрів у рекурентну мережу або через усереднення прогнозів по набору кадрів [15]. Натомість, SlowFast реалізує інший підхід: це глибока 3D-CNN, яка одночасно оперує як просторовими, так і часовими ознаками без використання LSTM. Модель SlowFast складається з двох паралельних конволюційних потоків: «повільного» з низькою частотою кадрів для вичерпного аналізу просторових ознак, та «швидкого» з високою частотою кадрів для детального аналізу руху [59]. Така архітектура дозволяє безпосередньо вчитися на відеопослідовностях, але потребує більше ресурсів. На відміну від неї, наша система обробляє відео двома етапами: спочатку кластеризує кадри кожного відео (метод DBSCAN) для визначення ключових кадрів, а потім аналізує тільки ці обрані кадри DCNN-моделлю. Це своєрідна альтернатива явному моделюванню часових залежностей: ми зменшуємо довжину послідовності, фокусуючись на репрезентативних моментах (кадрах) відео. Такий підхід простіший та швидший, хоча теоретично може пропустити деяку інформацію між ключовими кадрами. Втім, результати показали, що якість пошуку майже не постраждала, зате продуктивність зросла. Кешування ознак і індексування відео в нашій системі – ще одна методична перевага: ознаки ключових кадрів зберігаються в базі та індексуються, тому пошук зводиться до порівняння векторів ознак (через обчислення косинусної відстані чи іншої міри схожості) замість прогону всієї нейромережі по всіх кадрах кожного разу [90]. Самі моделі EfficientNet, MobileNet, SlowFast не надають вбудованих механізмів кешування – це радше рівень системної реалізації. Проте на практиці будь-яка система контентного пошуку відео, що використовує ці моделі, також може виграти від попереднього обчислення та збереження ознак, як зроблено в нашій

реалізації. Час обробки. Завдяки згаданому скороченню даних наша система досягає кращої швидкодії на інференсі, ніж якщо б довелося аналізувати весь відеопотік нейромережею. Наприклад, SlowFast для досягнення своєї високої точності обробляє відео з високою частотою кадрів у двох потоках паралельно, що значно збільшує час обчислень (модель містить мільйони параметрів і виконує на порядки більше операцій)

Натомість, MobileNet чи EfficientNet обробляють по одному кадру за раз і є дуже швидкими на одиничному зображенні, проте без додаткових оптимізацій їм довелося б послідовно проганяти десятки кадрів для кожного відео-запиту, що теж збільшує загальний час. У нашій системі цей компроміс вирішено через попередній аналіз: DCNN опрацьовує відео офлайн, виділяючи ключові ознаки, а під час онлайн-пошуку виконується лише порівняння ознак та мінімальний додатковий аналіз [91]. Отже, методично наша система поєднує сильні сторони кількох підходів (DCNN для візуальних ознак, кластеризація для скорочення даних, FFNN для автонастройки, індексація для швидкого пошуку), що забезпечило їй конкурентні показники. Це узгоджується з сучасними тенденціями: комбінування кількох моделей і алгоритмів дозволяє досягти балансу між точністю та швидкістю, якого важко дістатись, використовуючи одну лише глибоку модель або наївний перебір даних. Загалом, порівняльний аналіз показує, що запропонована система не поступається сучасним аналогам: вона є більш адаптивною (динамічно настроює свої параметри), забезпечує високу точність пошуку на рівні найкращих легких моделей і наближається до ефективності важчих моделей, одночасно залишаючись ефективною за часом обробки та ресурсами. Це робить її перспективною для практичного використання у задачах швидкого пошуку відео за фрагментом у великих базах даних.

Висновки до 4 розділу

У розділі роботи було розроблено архітектуру прикладної інформаційної системи для пошуку відеоконтенту за довільним фрагментом, яка ґрунтується на попередньо створених нейронних обчислювальних моделях. Побудовано

комплексну ієрархічну структуру модулів, що узгоджено поєднує процедури детекції сцен, екстракції семантичних ознак, формування векторних представлень для порівняння, а також застосування механізмів зіставлення на основі Temporal Similarity Matching (TSM) та Object Similarity Matching (OSM). Це дозволило врахувати як часову динаміку відеоданих, так і їх візуальний контекст, значно покращивши точність пошуку.

Також було забезпечено ефективну інтеграцію розробленої системи з NoSQL-базою даних MongoDB, структура якої адаптована до зберігання та швидкого доступу до складної, ієрархічно організованої і динамічно змінюваної відеоінформації. В базі даних реалізовано зберігання метаінформації про сцени, ключові кадри, а також їхні ознаки у формі компактних векторних представлень, що значно прискорює процес індексації та вибірки інформації.

Результати апробації системи свідчать, що запропоноване поєднання глибоких згорткових нейронних мереж (DCNN) для візуальної обробки відеоданих, рекурентних нейромереж типу LSTM для моделювання часових залежностей та повнозв’язних нейронних мереж (FNN) для автоадаптації параметрів забезпечує високу точність та швидкодію роботи системи. При цьому система продемонструвала можливість масштабування та ефективної роботи в режимі реального часу, підтвердивши конкурентні переваги над існуючими аналогами за критеріями точності, адаптивності до нових типів контенту та ресурсоефективності.

Таким чином, запропонована інформаційна система підтвердила свою працездатність, практичну цінність та ефективність для розв'язання завдання пошуку відеоконтенту за довільними фрагментами, повною мірою відповідаючи сучасним вимогам до Content-Based Video Information Retrieval (CBVIR) систем. Це відкриває перспективи для подальшого вдосконалення й імплементації розроблених алгоритмічних підходів у прикладних рішеннях широкого спектра задач аналізу та пошуку відеоінформації.

ВИСНОВОК

У дисертаційній роботі розв’язано актуальне наукове завдання підвищення точності аналізу й пошуку подібних фрагментів у відеопотоках контентно-орієнтованих систем відеоінформації. Наведені нижче підсумки відображають основні результати:

1. Проведено аналіз існуючих методів сегментації та індексації відео. Виконано порівняння підходів до детектування меж сцен (SBD), виділення ключових кадрів та інтегрування нейронних мереж. Це дало змогу виявити обмеження традиційних рішень і сформулювати перелік невирішених проблем, пов’язаних із точністю та ефективністю пошуку відео за довільним фрагментом.
2. Удосконалено методи детекції сцен у відеопотоках, що поєднують аналіз гістограм у просторі HSV, абсолютні різниці пікселів та механізми уваги. Показано, що інтегрована модель зменшує вплив шуму і неточностей у межах сцен, зокрема за рахунок автоматичного налаштування порогів та врахування поступових переходів (Soft Cuts).
3. Розроблено обчислювальні моделі для екстракції характеристик сцен, зокрема просторових і часових ознак. Основна реалізація використовує ефективну комбінацію моделей EfficientNet (для просторових ознак) і LSTM із увагою (для часових залежностей), що дозволяє досягти балансу між точністю та обчислювальними витратами. Окремо досліджено архітектуру SlowFast, що об'єднує два паралельні потоки обробки відео: повільний (Slow) для аналізу статичних сцен і швидкий (Fast) для динамічних фрагментів. Розроблено експериментальну реалізацію цієї моделі у TensorFlow, яка показала високу точність при обробці складних відеоматеріалів з інтенсивним рухом.
4. Запропоновано ефективний підхід до побудови векторів ознак відео, що комбінують просторові характеристики об’єктів, часову інформацію (динаміку руху), а також геометричні та колірні атрибути. Доведено, що

формування єдиного інтегрованого вектора ознак спрощує процедури порівняння та пошуку подібностей у великих відеобазах.

5. Розроблено систему пошуку подібностей у відео на основі метрик евклідового простору, косинусної подібності та Жаккара. Показано, що застосування глибинних згорткових нейронних мереж (DCNN) у комбінації з рекурентними (LSTM) ефективніше виявляє складні закономірності й суттєво поліпшує точність пошуку за фрагментами.
6. Вдосконалено методи побудови модульної архітектури для індексації та аналізу відео, що охоплюють етапи детекції меж сцен, виділення ключових об’єктів (MobileNet, EfficientNet) та узгодження часових ознак (LSTM). Така архітектура масштабовано працює з великими відеобазами й забезпечує високу швидкодію при збереженні якості аналізу.
7. Розроблено проект інформаційної системи пошуку фрагментів відео, орієнтований на роботу з обмеженими обчислювальними ресурсами. Запропоновано використання хмарного підходу та розріджених представлень даних (Sparse Representation) для підтримання швидкого доступу та мінімізації навантаження.
8. Практична цінність одержаних результатів полягає в можливості впровадження створених моделей і методів у контентно-орієнтовані системи (CBVIR) для ефективного пошуку та класифікації відеоматеріалів у реальному часі. Запропоновані в роботі підходи забезпечують високу точність аналізу динамічних сцен, а також спрощують адаптацію системи під різні умови та формати відео.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Xia, K., Huang, J., & Wang, H. (2020). LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access*, *8*, 56855–56866. https://doi.org/10.1109/access.2020.2982225
2. Sunuwar, J., & Borah, S. (2024). A comparative analysis on major key-frame extraction techniques. Multimedia Tools and Applications, 83(30), 73865–73910. https://doi.org/10.1007/s11042-024-18380-z
3. Tiwari, V., & Bhatnagar, C. (2021). A survey of recent work on video summarization: approaches and techniques. Multimedia Tools and Applications, 80(18), 27187–27221. https://doi.org/10.1007/s11042-021-10977-y
4. Jain, R., Jain, P., Kumar, T., & Dhiman, G. (2021). Real time video summarizing using image semantic segmentation for CBVR. Journal of Real-Time Image Processing, 18(5), 1827–1836. https://doi.org/10.1007/s11554-021-01151-6
5. Ye, Y., Zhang, X., Lin, Y., & Wang, H. (2019). Facial expression recognition via region-based convolutional fusion network. Journal of Visual Communication and Image Representation, 62, 1–11. https://doi.org/10.1016/j.jvcir.2019.04.009
6. Sharma, V., Tapaswi, M., & Stiefelhagen, R. (2020). Deep multimodal feature encoding for video ordering. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2004.02205
7. Lungisani, B., Thuma, E., & Malema, G. (2017). A classification approach to video shot boundary detection. International Journal of Signal Processing Image Processing and Pattern Recognition, 10(12), 103–118. https://doi.org/10.14257/ijsip.2017.10.12.08
8. Yuan, Y., & Zhang, J. (2023). Shot boundary detection using color clustering and attention mechanism. ACM Transactions on Multimedia Computing Communications and Applications, 19(6), 1–23. https://doi.org/10.1145/3595923

9. Smith, M. A., & Chen, T. (2005). Image and video indexing and retrieval. In Elsevier eBooks (pp. 993–XXXI). https://doi.org/10.1016/b978-012119792-6/50121-2
10. Zhang, Q.L., Li, S.L., Duan, J.G. et al. Moving Object Detection Method Based on the Fusion of Online Moving Window Robust Principal Component Analysis and Frame Difference Method. Neural Process Lett 56, 55 (2024). https://doi.org/10.1007/s11063-024-11463-w
11. Cui, C., Liu, L., & Qiao, R. (2024). A cutting-edge video anomaly detection method using image quality assessment and attention mechanism-based deep learning. Alexandria Engineering Journal, 108, 476–485. https://doi.org/10.1016/j.aej.2024.07.103
12. Kar, T., Kanungo, P., Mohanty, S. N., Groppe, S., & Groppe, J. (2024). Video shot-boundary detection: issues, challenges and solutions. Artificial Intelligence Review, 57(4). https://doi.org/10.1007/s10462-024-10742-1
13. Xu, H., Ling, Z., Yuan, X., & Wang, Y. (2024). A video object detector with Spatio-Temporal Attention Module for micro UAV detection. Neurocomputing, 597, 127973. https://doi.org/10.1016/j.neucom.2024.127973
14. Saini, R., Kumar, P., Roy, P. P., & Pal, U. (2019). Modeling local and global behavior for trajectory classification using graph based algorithm. Pattern Recognition Letters, 150, 280–288. https://doi.org/10.1016/j.patrec.2019.05.014
15. Mouhcine, K., Zrira, N., Elafi, I., Benmiloud, I., & Khan, H. A. (2024). STEFF: Spatio-temporal EfficientNet for dynamic texture classification in outdoor scenes. Heliyon, 10(3), e25360. https://doi.org/10.1016/j.heliyon.2024.e25360
16. Sasithradevi, A., & Roomi, S. M. M. (2019). Video classification and retrieval through spatio-temporal Radon features. Pattern Recognition, 99, 107099. https://doi.org/10.1016/j.patcog.2019.107099
17. Yousefi, S.M.M., Mohseni, S.S., Dehbovid, H. et al. A Practical Approach to Tracking Estimation Using Object Trajectory Linearization. Int J Comput Intell Syst 17, 175 (2024). https://doi.org/10.1007/s44196-024-00579-5

18. Fu, L., Zhang, Q., & Tian, S. (2024). Real-time video surveillance on highways using combination of extended Kalman Filter and deep reinforcement learning. Heliyon, 10(5), e26467. https://doi.org/10.1016/j.heliyon.2024.e26467
19. Uusinoka, M., Haapala, J., & Polojärvi, A. (2025). Deep Learning-Based Optical Flow in Fine-Scale Deformation mapping of sea Ice dynamics. Geophysical Research Letters, 52(2). https://doi.org/10.1029/2024gl112000
20. Nallappan, M., & Velswamy, R. (2024). Exploring deep learning-based content-based video retrieval with Hierarchical Navigable Small World index and ResNet-50 features for anomaly detection. Expert Systems With Applications, 247, 123197. https://doi.org/10.1016/j.eswa.2024.123197
21. Spolaôr, N., Lee, H. D., Takaki, W. S. R., Ensina, L. A., Coy, C. S. R., & Wu, F. C. (2020). A systematic review on content-based video retrieval. Engineering Applications of Artificial Intelligence, 90, 103557. https://doi.org/10.1016/j.engappai.2020.103557
22. Su, P., & Wu, C. (2015). Efficient copy detection for compressed digital videos by spatial and temporal feature extraction. Multimedia Tools and Applications, 76(1), 1331–1353. https://doi.org/10.1007/s11042-015-3132-1
23. M. Asim, M. Zhu, and M. Y. Javed, “CNN based spatiotemporal feature extraction for face anti-spoofing,” 2017 2nd Int. Conf. Image, Vis. Comput. ICIVC 2017, pp. 234–238, Jul. 2017
24. H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 2013, pp. 3551-3558, doi: 10.1109/ICCV.2013.441.
25. Shi, K., Xiao, W., Wu, G., Xiao, Y., Lei, Y., Yu, J., & Gu, Y. (2021). Temporal-Spatial feature extraction of DSA video and its application in AVM diagnosis. Frontiers in Neurology, 12. https://doi.org/10.3389/fneur.2021.655523
26. Yildirim, Y., & Yazici, A. (2007). Ontology-Supported video modeling and retrieval. In Springer eBooks (pp. 28–41). https://doi.org/10.1007/978-3-540-71545-0_3

27. Mazaheri, A., Gong, B., & Shah, M. (2018). Learning a Multi-Concept Video Retrieval Model with Multiple Latent Variables. ACM Transactions on Multimedia Computing Communications and Applications, 14(2), 1–21. https://doi.org/10.1145/3176647

28. Mühling, M., Korfhage, N., Müller, E., Otto, C., Springstein, M., Langelage, T., Veith, U., Ewerth, R., & Freisleben, B. (2017). Deep learning for content-based video retrieval in film and television production. Multimedia Tools and Applications, 76(21), 22169–22194. https://doi.org/10.1007/s11042-017-4962-9

29. Mohapatra, S., Weisshaar, J.C. Modified Pearson correlation coefficient for two-color imaging in spherocylindrical cells. BMC Bioinformatics 19, 428 (2018). https://doi.org/10.1186/s12859-018-2444-3

30. Hammouche, R., Attia, A., Akhrouf, S., & Akhtar, Z. (2022). Gabor filter bank with deep autoencoder based face recognition system. Expert Systems With Applications, 197, 116743. https://doi.org/10.1016/j.eswa.2022.116743

31. Shu, X., Song, Z., Shi, J., Huang, S., & Wu, X. (2021). Multiple channels local binary pattern for color texture representation and classification. Signal Processing Image Communication, 98, 116392. https://doi.org/10.1016/j.image.2021.116392

32. Zhang, X., Zhang, Q., Li, P., You, J., Sun, J., & Zhou, J. (2024). Multi-Feature Extraction and Selection Method to Diagnose Burn Depth from Burn Images. Electronics, 13(18), 3665. https://doi.org/10.3390/electronics13183665

33. Zhao, Y., Wang, R., Wang, W., & Gao, W. (2016). Local Quantization Code histogram for texture classification. Neurocomputing, 207, 354–364. https://doi.org/10.1016/j.neucom.2016.05.016

34. Li, B., Li, Y., & Wu, Q. J. (2023). A completed parted region local neighborhood energy pattern for texture classification. Digital Signal Processing, 137, 104031. https://doi.org/10.1016/j.dsp.2023.104031

35. Choi, W., Lam, K., & Siu, W. (2001). An adaptive active contour model for highly irregular boundaries. Pattern Recognition, 34(2), 323–331. https://doi.org/10.1016/s0031-3203(99)00231-9

36. Xue, P., & Niu, S. (2024b). A novel active contour model based on features for image segmentation. Pattern Recognition, 155, 110673. https://doi.org/10.1016/j.patcog.2024.110673

37. Yan, T., Liu, Y., Wei, D., Sun, X., & Liu, Q. (2023). Shape analysis of sand particles based on Fourier descriptors. Environmental Science and Pollution Research, 30(22), 62803–62814. https://doi.org/10.1007/s11356-023-26388-5

38. Lowe, D. G. & Computer Science Department, University of British Columbia. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 2004.

39. Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. International Conference on Computer Vision, 2564–2571. https://doi.org/10.1109/iccv.2011.6126544

40. Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10), 1615–1630. https://doi.org/10.1109/tpami.2005.188

41. Shakhovska, N., Melnykova, N., Pobereiko, P., & Zakharchuk, M. (2023). Investigating Methods of Searching for Key Frames in Video Flow with the Use of Neural Networks for Search Systems. International Journal of Computing, 455–461. https://doi.org/10.47839/ijc.22.4.3352

42. Chuquirachi, F., Siow, C., Chin, W. H., & Kubota, N. (2022, September). Human action repetition counting based on similarity of convolutional neural networks features of frames. Paper presented at the 30th Symposium on Fuzzy, Artificial Intelligence, Neural Networks and Computational Intelligence (FAN 2022), Kobe, Japan.

43. P. Rohan, "Euclidean Distance and Normalizing a Vector", Kaggle, 2020. [Online]. Available: https://www.kaggle.com/code/paulrohan2020/euclidean-distance-and-normalizing-a-vector

44. Melnykova, N., Shakhovska, N., Pobereiko, P., & Cichoń, D. (2024). Advancing video search capabilities: Integrating feedforward neural networks for efficient Fragment-Based retrieval. In Lecture notes on data engineering and communications technologies (pp. 181–197). https://doi.org/10.1007/978-3-031-62213-7_9
45. Fletcher, S., & Islam, M. Z. (2018). Comparing sets of patterns with the Jaccard index. AJIS. Australasian Journal of Information Systems/AJIS. Australian Journal of Information Systems/Australian Journal of Information Systems, 22. https://doi.org/10.3127/ajis.v22i0.1538
46. Hassan, E. (2021). Learning video actions in two stream recurrent neural network. Pattern Recognition Letters, 151, 200–208. https://doi.org/10.1016/j.patrec.2021.08.017
47. Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2017). Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. IEEE Access, 6, 1155–1166. https://doi.org/10.1109/access.2017.2778011
48. Yin, M., Liao, S., Liu, X., Wang, X., & Yuan, B. (2021). Towards Extremely Compact RNNs for Video Recognition with Fully Decomposed Hierarchical Tucker Structure. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2104.05758
49. Niu, Z., Zhong, G., Yue, G., Wang, L., Yu, H., Ling, X., & Dong, J. (2022). Recurrent attention unit: A new gated recurrent unit for long-term memory of important parts in sequential data. Neurocomputing, 517, 1–9. https://doi.org/10.1016/j.neucom.2022.10.050
50. Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 3634–3640. https://doi.org/10.24963/ijcai.2018/505
51. Zeghina, A., Leborgne, A., Ber, F. L., & Vacavant, A. (2024). Deep learning on spatiotemporal graphs: A systematic review, methodological landscape, and

research opportunities. Neurocomputing, 594, 127861. https://doi.org/10.1016/j.neucom.2024.127861

52. Melnykova, N., & Pobereiko, P. (2022b). Research of methods of searching key frames in video flow with the use of neural networks for search systems. Herald of Khmelnytskyi National University Technical Sciences, 309(3), 55–60. https://doi.org/10.31891/2307-5732-2022-309-3-55-60
53. Yi, M., & Li, M. (2024). Efficient Video to Audio Mapper with Visual Scene Detection. (Cornell University). https://doi.org/10.48550/arxiv.2409.09823
54. Mohammad, D., Aljarrah, I., & Jarrah, M. (2021). Searching surveillance video contents using convolutional neural network. International Journal of Power Electronics and Drive Systems/International Journal of Electrical and Computer Engineering, 11(2), 1656. https://doi.org/10.11591/ijece.v11i2.pp1656-1665
55. Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2018). SlowFast networks for video recognition. (Cornell University).
56. Sun, D., Yang, X., Liu, M., & Kautz, J. (2019). Models matter, so does training: An Empirical study of CNNs for Optical flow estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(6), 1408–1423. https://doi.org/10.1109/tpami.2019.2894353
57. Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal Relational Reasoning in Videos. Proceedings of the European Conference on Computer Vision (ECCV), 803–818.
58. Song, H., Wang, Y., & Zhang, L. (2024). Layer Reuse Strategies for Efficient Scene and Keyframe Detection. IEEE Transactions on Multimedia, 26(5), 1024–1035.
59. Łęcki, J., Hering, M., Jabłoński, M., & Karpus, A. (2024). An analysis of the performance of lightweight CNNs in the context of object detection on mobile phones. Proceedings of the International Conference on Information Systems Development. https://doi.org/10.62036/isd.2024.12
60. Basystiuk, O., Melnykova, N., Rybchak, Z., & Lviv Polytechnic National University. (2023). Machine learning methods and tools for facial recognition

based on multimodal approach. In Lviv Polytechnic National University. https://ceur-ws.org/Vol-3426/paper13.pdf

61. Kohut, N., Basystiuk, O., Shakhovska, N., & Melnykova, N. (2024). Detecting plant diseases using machine learning models. Sustainability, 17(1), 132. https://doi.org/10.3390/su17010132
62. Thirunagari, C., & Ferdouse, L. (2024). Enhanced Public Safety: Real-Time Crime Detection with CNN-LSTM in Video Surveillance. In Lecture notes on data engineering and communications technologies (pp. 41–54). https://doi.org/10.1007/978-3-031-80817-3_3
63. Li, D., Yao, T., Duan, L., Mei, T., & Rui, Y. (2018). Unified Spatio-Temporal attention networks for action recognition in videos. IEEE Transactions on Multimedia, 21(2), 416–428. https://doi.org/10.1109/tmm.2018.2862341
64. Laakom, F., Raitoharju, J., Iosifidis, A., & Gabbouj, M. (2024). Reducing redundancy in the bottleneck representation of autoencoders. Pattern Recognition Letters, 178, 202–208. https://doi.org/10.1016/j.patrec.2024.01.013
65. Kanimozhi, T., & Latha, K. (2014). An integrated approach to region based image retrieval using firefly algorithm and support vector machine. Neurocomputing, 151, 1099–1111. https://doi.org/10.1016/j.neucom.2014.07.078
66. Zhang, H., Dong, Y., Li, J., & Xu, D. (2021). Dynamic time warping under product quantization, with applications to Time-Series data similarity search. IEEE Internet of Things Journal, 9(14), 11814–11826. https://doi.org/10.1109/jiot.2021.3132017
67. Alrebdi, N., & Al-Shargabi, A. A. (2024). Bilingual video captioning model for enhanced video retrieval. Journal of Big Data, 11(1). https://doi.org/10.1186/s40537-024-00878-w
68. Mienye, I. D., & Swart, T. G. (2024). A comprehensive review of deep learning: architectures, recent advances, and applications. Information, 15(12), 755. https://doi.org/10.3390/info15120755

69. Bougaham, A., Delchevalerie, V., Adoui, M. E., & Frénay, B. (2024). Industrial and medical anomaly detection through cycle-consistent adversarial networks. Neurocomputing, 128762. https://doi.org/10.1016/j.neucom.2024.128762
70. Hazhir, M. M., & Foroughi, A. A. (2024). A compromise programming approach for cross efficiency measurement in basic two-stage network system. Expert Systems With Applications, 252, 124205. https://doi.org/10.1016/j.eswa.2024.124205
71. Weyns, D., Gheibi, O., Quin, F., & Van Der Donckt, J. (2022). Deep learning for effective and efficient reduction of large adaptation spaces in self-adaptive systems. ACM Transactions on Autonomous and Adaptive Systems, 17(1–2), 1–42. https://doi.org/10.1145/3530192
72. «MongoDB Atlas Stream Processing – mongodb.com» [Онлайновий]. Доступний: https://www.mongodb.com/products/platform/atlas-stream-processing [Дата звернення: 25 03 2025]
73. «Milvus – milvus.io» [Онлайновий]. Доступний: https://milvus.io/ [Дата звернення: 25 03 2025]
74. «Mastering ASP.NET Core with Minimal APIs & CQRS - Part 01: Introduction – medium.com» [Онлайновий]. Доступний: https://sasangaedirisinghe.medium.com/mastering-asp-net-core-with-minimal-apis-cqrs-part-01-introduction-9015cca505c2 [Дата звернення: 25 03 2025]
75. «Wolverine – wolverine.netlify.app» [Онлайновий]. Доступний: https://wolverine.netlify.app/ [Дата звернення: 25 03 2025]
76. «Why use RabbitMQ in a Microservice Architecture? – cloudamqp.com» [Онлайновий]. Доступний: https://www.cloudamqp.com/blog/why-use-rabbitmq-in-a-microservice-architecture.html [Дата звернення: 25 03 2025]
77. Kang, D., Emmons, J., Abuzaid, F., Bailis, P., & Zaharia, M. (2017). NoScope. Proceedings of the VLDB Endowment, 10(11), 1586–1597. https://doi.org/10.14778/3137628.3137664
78. «FFmpeg Documentation – ffmpeg.org» [Онлайновий]. Доступний: https://ffmpeg.org/documentation.html [Дата звернення: 25 03 2025]

79. «gRPC Documentation – grpc.io» [Онлайновий]. Доступний: https://grpc.io/docs/ [Дата звернення: 25 03 2025]
80. «PySceneDetect – github.com» [Онлайновий]. Доступний: https://github.com/Breakthrough/PySceneDetect [Дата звернення: 25 03 2025]
81. Kumar, A., Singh, P., & Verma, R. (2021). Certain homological invariants of bipartite kneser graphs. Journal of Algebra and Its Applications, 21(10).
82. Zhu, J., Kang, D., & Maji, T. (2021). Angularity in DIS at next-to-next-to-leading log accuracy. Journal of High Energy Physics, 2021(11).
83. Guarino, A., & Sterckx, C. (2021). Flat deformations of type IIB S-folds. Journal of High Energy Physics, 2021(11).
84. Wu, W., Zhao, Y., Li, Z., Li, J., Zhou, H., Shou, M. Z., & Bai, X. (2023). A Large Cross-Modal Video Retrieval Dataset with Reading Comprehension. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2305.03347
85. Trojahn, T. H., Kishi, R. M., & Goularte, R. (2018). A new multimodal deep-learning model to video scene segmentation. WebMedia '18: Proceedings of the 24th Brazilian Symposium on Multimedia and the Web. https://doi.org/10.1145/3243082.3243108
86. Vercel Documentation – vercel.com» [Онлайновий]. Доступний: https://vercel.com/docs [Дата звернення: 25 03 2025]
87. «Tailwind CSS Documentation – tailwindcss.com» [Онлайновий]. Доступний: https://tailwindcss.com/docs [Дата звернення: 25 03 2025]
88. Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. Neural Networks, 106, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011
89. Kanoor, S., Fraunholz, D., & Schotten, H. D. (2018). Evaluation of machine learning-based anomaly detection algorithms on an industrial ModBus/TCP data set. Proceedings of the 17th International Conference on Availability, Reliability and Security, 1–9.
90. Wang, X., Ke, B., Li, X., Liu, F., Zhang, M., Liang, X., & Xiao, Q. (2022). Modality-Balanced embedding for video retrieval. Proceedings of the 45th

International ACM SIGIR Conference on Research and Development in Information Retrieval, 2578–2582. https://doi.org/10.1145/3477495.3531899

91. P, H. R., Srivastava, V., Chaurasia, K., Pacheco, R. G., & Couto, R. S. (2024). Early-Exit Deep Neural Network - A comprehensive survey. ACM Computing Surveys, 57(3), 1–37.
92. Melnykova N., Pobereiko P. Enhancing fragment-based video retrieval through the integration of feedforward neural networks // European Science. – 2024. – Issue sge27-03. – P. 126–146.

ДОДАТОК А. ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ МОДУЛЯ СЕГМЕНТАЦІЇ ТА ПОБУДОВИ ВЕКТОРІВ ОЗНАК ВІДЕО

```
import numpy as np
import tensorflow as tf

from data_processing.video_loader import VideoLoader
from feature_extraction.temporal_features import extract_temporal_features
from feature_extraction.spatial_features import extract_spatial_features
from feature_extraction.object_features import ObjectFeaturesExtractor
from integration.feature_integrator import integrate_features
from storage.mongo_connector import MongoConnector
from utils.helpers import ensure_dir, current_timestamp
from config.config import VIDEO_DIR, OUTPUT_DIR
from neural_models.slowfast_feature_extractor.slowfast_model

def main(video_file):
    ensure_dir(OUTPUT_DIR)

    slow_frames = 8
    fast_frames = 32
    height, width = 224, 224
    num_classes = 400

    slowfast_model = create_slowfast_model(
        slow_frames=slow_frames,
        fast_frames=fast_frames,
        height=height,
        width=width,
        num_classes=num_classes,
        fusion_method='concat'
    )
    video_loader = VideoLoader(VIDEO_DIR)
    frames = video_loader.extract_frames(video_file)
    temporal_features = extract_temporal_features(frames)

    object_feature_extractor = ObjectFeaturesExtractor()

    spatial_features_list = []
    object_features_list = []
```

```
    slow_indices = np.round(np.linspace(0, len(frames) - 1, slow_frames)).astype(int)
    fast_indices = np.round(np.linspace(0, len(frames) - 1, fast_frames)).astype(int)

    slow_frames_data = [frames[i] for i in slow_indices]
    fast_frames_data = [frames[i] for i in fast_indices]

    def preprocess_frame(frame):
        frame = frame.astype('float32') / 255.0
        return frame

    slow_frames_batch = np.array([preprocess_frame(f) for f in slow_frames_data],
dtype=np.float32)
    fast_frames_batch = np.array([preprocess_frame(f) for f in fast_frames_data],
dtype=np.float32)

    slow_input_tensor = tf.convert_to_tensor(slow_frames_batch[None, ...])
    fast_input_tensor = tf.convert_to_tensor(fast_frames_batch[None, ...])

    predictions = slowfast_model([slow_input_tensor, fast_input_tensor], training=False)
    predictions = predictions.numpy()[0]

    for idx, frame in enumerate(frames):
        spatial_feat = extract_spatial_features(frame)
        object_feat = object_feature_extractor.extract_features(frame)
        spatial_features_list.append(spatial_feat)
        object_features_list.append(object_feat)

    spatial_features = np.mean(spatial_features_list, axis=0)
    object_features = np.mean(object_features_list, axis=0)

    integral_vector = integrate_features(
        temporal_features,
        spatial_features,
        object_features
    )

    combined_features = np.concatenate([integral_vector, predictions], axis=0)
```

ДОДАТОК Б. ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ МОДУЛЯ ДЕТЕКЦІЇ СЦЕН

```
from tensorflow.keras.layers import Input, concatenate
from tensorflow.keras.models import Model
from neural_models.base.convolutional_layers import conv_block
from neural_models.base.pooling_layers import max_pooling_block,
global_avg_pooling_block
from neural_models.base.fully_connected_layers import fc_block
from neural_models.dcnn_scene_detector.histogram_diff_layer import HistogramDiffLayer
from neural_models.dcnn_scene_detector.pixel_diff_layer import PixelDiffLayer

def create_scene_detector(input_shape=(224, 224, 3)):
    frame1_input = Input(shape=input_shape)
    frame2_input = Input(shape=input_shape)

    def cnn_branch(frame_input):
        x = conv_block(frame_input, 32)
        x = max_pooling_block(x)
        x = conv_block(x, 64)
        x = global_avg_pooling_block(x)
        return x

    frame1_features = cnn_branch(frame1_input)
    frame2_features = cnn_branch(frame2_input)

    hist_diff = HistogramDiffLayer()([frame1_input, frame2_input])
    pix_diff = PixelDiffLayer()([frame1_input, frame2_input])

    combined_features = concatenate([frame1_features, frame2_features, hist_diff,
pix_diff])

    x = fc_block(combined_features, 128)
    output = Dense(2, activation='softmax')(x)  # 2 класи: hard cut, soft cut

    model = Model(inputs=[frame1_input, frame2_input], outputs=output)
    model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

    return model
```

ДОДАТОК В. ПРОГРАМНИЙ КОД РЕАЛІЗАЦІЇ SLOWFAST МЕРЕЖІ

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

def residual_block(x, filters, strides=(1, 1, 1), name=None):
    shortcut = x
    x = layers.Conv3D(filters, kernel_size=(3, 3, 3), strides=strides,
                      padding='same', name=f"{name}_conv1" if name else None)(x)
    x = layers.BatchNormalization(name=f"{name}_bn1" if name else None)(x)
    x = layers.ReLU(name=f"{name}_relu1" if name else None)(x)
    x = layers.Conv3D(filters, kernel_size=(3, 3, 3), padding='same',
                      name=f"{name}_conv2" if name else None)(x)
    x = layers.BatchNormalization(name=f"{name}_bn2" if name else None)(x)
    if strides != (1, 1, 1) or shortcut.shape[-1] != filters:
        shortcut = layers.Conv3D(filters, kernel_size=(1, 1, 1), strides=strides,
                                 padding='same', name=f"{name}_shortcut" if name else
None)(shortcut)
        shortcut = layers.BatchNormalization(name=f"{name}_shortcut_bn" if name else
None)(shortcut)
    x = layers.add([x, shortcut], name=f"{name}_add" if name else None)
    x = layers.ReLU(name=f"{name}_relu_out" if name else None)(x)
    return x

def build_slow_pathway(input_shape, base_filters=64, num_classes=400):
    inputs = keras.Input(shape=input_shape)
    x = layers.Conv3D(base_filters, kernel_size=(1, 7, 7), strides=(1, 2, 2),
                      padding='same', name='slow_conv1')(inputs)
    x = layers.BatchNormalization(name='slow_bn1')(x)
    x = layers.ReLU(name='slow_relu1')(x)
    x = layers.MaxPool3D(pool_size=(1, 3, 3), strides=(1, 2, 2), padding='same',
                         name='slow_pool1')(x)
    x = residual_block(x, base_filters, name='slow_res1_1')
    x = residual_block(x, base_filters, name='slow_res1_2')
    x = residual_block(x, base_filters * 2, strides=(1, 2, 2), name='slow_res2_1')
    x = residual_block(x, base_filters * 2, name='slow_res2_2')
    x = residual_block(x, base_filters * 4, strides=(1, 2, 2), name='slow_res3_1')
    x = residual_block(x, base_filters * 4, name='slow_res3_2')
    x = layers.GlobalAveragePooling3D(name='slow_global_pool')(x)
```

```
    outputs = layers.Dense(num_classes, name='slow_fc')(x)
    return keras.Model(inputs, outputs, name='SlowPathway')

def build_fast_pathway(input_shape, base_filters=8, num_classes=400):
    inputs = keras.Input(shape=input_shape)
    x = layers.Conv3D(base_filters, kernel_size=(5, 7, 7), strides=(1, 2, 2),
                      padding='same', name='fast_conv1')(inputs)
    x = layers.BatchNormalization(name='fast_bn1')(x)
    x = layers.ReLU(name='fast_relu1')(x)
    x = layers.MaxPool3D(pool_size=(1, 3, 3), strides=(1, 2, 2), padding='same',
                         name='fast_pool1')(x)
    x = residual_block(x, base_filters, name='fast_res1_1')
    x = residual_block(x, base_filters, name='fast_res1_2')
    x = residual_block(x, base_filters * 2, strides=(1, 2, 2), name='fast_res2_1')
    x = residual_block(x, base_filters * 2, name='fast_res2_2')
    x = residual_block(x, base_filters * 4, strides=(1, 2, 2), name='fast_res3_1')
    x = residual_block(x, base_filters * 4, name='fast_res3_2')

    x = layers.GlobalAveragePooling3D(name='fast_global_pool')(x)
    outputs = layers.Dense(num_classes, name='fast_fc')(x)
    return keras.Model(inputs, outputs, name='FastPathway')

class SlowFastNetwork(keras.Model):
    def __init__(self, slow_frames=8, fast_frames=32, height=224,
width=224, channels=3, slow_base_filters=64, fast_base_filters=8,
num_classes=400, fusion_method='concat', **kwargs):
        super(SlowFastNetwork, self).__init__(**kwargs)

        self.slow_frames = slow_frames
        self.fast_frames = fast_frames
        self.height = height
        self.width = width
        self.channels = channels
        self.num_classes = num_classes
        self.fusion_method = fusion_method

        self.slow_pathway = build_slow_pathway(
            (slow_frames, height, width, channels),
            base_filters=slow_base_filters,
            num_classes=num_classes)
        self.fast_pathway = build_fast_pathway(
```

```
            (fast_frames, height, width, channels),
            base_filters=fast_base_filters,
            num_classes=num_classes
        )
        if fusion_method == 'concat':
            self.fusion_dense = layers.Dense(num_classes)
        elif fusion_method == 'weighted':
            self.slow_weight = self.add_weight(
                name="slow_weight",
                shape=(),
                initializer=keras.initializers.Constant(0.5),
                trainable=True)
        self.activation = layers.Activation('softmax', name='predictions')

    def call(self, inputs, training=None):
        slow_input, fast_input = inputs
        slow_output = self.slow_pathway(slow_input, training=training)
        fast_output = self.fast_pathway(fast_input, training=training)

        if self.fusion_method == 'concat':
            x = layers.concatenate([slow_output, fast_output], name='concat_fusion')
            x = self.fusion_dense(x)
        elif self.fusion_method == 'add':
            x = layers.add([slow_output, fast_output], name='add_fusion')
        elif self.fusion_method == 'weighted':
            x = self.slow_weight * slow_output + (1 - self.slow_weight) * fast_output
        else:
            raise ValueError(f"Unsupported fusion method: {self.fusion_method}")
        return self.activation(x)

    def build_graph(self):
        slow_input = keras.Input(shape=(self.slow_frames, self.height, self.width,
self.channels))
        fast_input = keras.Input(shape=(self.fast_frames, self.height, self.width,
self.channels))
        model = keras.Model(
            inputs=[slow_input, fast_input],
            outputs=self.call([slow_input, fast_input]))
        return model

    def get_config(self):
```

```
        config = super(SlowFastNetwork, self).get_config()
        config.update({
            "slow_frames": self.slow_frames,
            "fast_frames": self.fast_frames,
            "height": self.height,
            "width": self.width,
            "channels": self.channels,
            "num_classes": self.num_classes,
            "fusion_method": self.fusion_method
        })
        return config

def create_slowfast_model(slow_frames=8, fast_frames=32, height=224, width=224,
                          num_classes=400, fusion_method='concat'):
    model = SlowFastNetwork(
        slow_frames=slow_frames,
        fast_frames=fast_frames,
        height=height,
        width=width,
        channels=3,
        num_classes=num_classes,
        fusion_method=fusion_method)
    model.compile(
        optimizer=keras.optimizers.Adam(learning_rate=0.0001),
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'])
    return model

if __name__ == "__main__":
    batch_size = 2, slow_frames, fast_frames = 8, 32
    height, width, channels = 224, 224, 3

    slow_input = tf.random.normal([batch_size, slow_frames, height, width, channels])
    fast_input = tf.random.normal([batch_size, fast_frames, height, width, channels])

    model = create_slowfast_model(
        slow_frames=slow_frames,
        fast_frames=fast_frames,
        num_classes=400)
        preds = model([slow_input, fast_input])
    model.build_graph().summary()
```

ДОДАТОК Г. АКТИ ВПРОВАДЖЕННЯ

"ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Національного університету
"Львівська політехніка"
_____ Олег ДАВИДЧАК
"15" 04 2025 р.

АКТ
про впровадження в навчальний процес результатів дисертаційної роботи Поберейка Петра Богдановича представленої на здобуття наукового ступеня доктора філософії

Цей акт складено про те, що результати дисертаційної роботи Поберейка Петра Богдановича впроваджено у навчальний процес кафедри «Системи штучного інтелекту» Національного університету «Львівська політехніка».

Впровадження результатів дисертаційної роботи полягає в їхньому використанні при викладанні навчальних дисциплін як окремих розділів лекційних курсів, так і в циклах лабораторних робіт.

Зокрема, у межах викладання дисципліни «Організація баз даних та знань» для студентів освітньо-кваліфікаційного рівня «бакалавр», що навчаються за спеціальністю 122 «Комп'ютерні науки», використано такі результати дисертаційного дослідження:

- Моделі побудови дескрипторів відео та формування векторів ознак;
- Методи індексації даних із використанням комбінованих дескрипторів (об'єктних і часових);
- Принципи оптимізації структури сховища для відеоінформації з урахуванням специфіки контенту;
- Приклад архітектури інформаційної системи, інтегрованої з NoSQL-сховищами.

Директор ІКНІ,
д.т.н., професор — Наталія ШАХОВСЬКА

Завідувач кафедри СШІ,
д.т.н., доцент — Наталія МЕЛЬНИКОВА

"ЗАТВЕРДЖУЮ"
Керівник ПНВП «РЕЗОН»
_____ **Михайло МАРТИНЯК**
«7» квітня 2025 р.

АКТ

про впровадження результатів дисертаційної роботи
аспіранта кафедри «Системи штучного інтелекту»
Національного університету «Львівська політехніка»
Поберейка Петра Богдановича

Цей акт підтверджує, що результати дисертаційної роботи Поберейка П.Б. були використані для створення прототипу системи контентного пошуку відеофрагментів у внутрішньому відеоархіві підприємства ПНВП «РЕЗОН» протягом 2024–2025 рр.

Впровадження дисертаційних досліджень Поберейка П.Б. полягає у наступному:

- Розроблено модуль автоматизованої індексації відеофайлів із використанням алгоритмів виділення сцен, сегментації кадрів та формування дескрипторів на основі глибоких нейронних мереж.
- Побудовано систему контентного пошуку за довільним фрагментом відео, що дозволяє оперативно знаходити схожі відео або їх частини на основі просторово-часових ознак.
- Інтегровано інтерфейс користувача для завантаження відео, попереднього аналізу кадрів та візуалізації результатів пошуку, що використовується для внутрішньої аналітики.

Даний акт не є підставою для взаємних фінансових розрахунків.

Керівник ПНВП «РЕЗОН» _____ **Михайло МАРТИНЯК**

ПРИВАТНЕ НАУКОВО-ВИРОБНИЧЕ ПІДПРИЄМСТВО «РЕЗОН» УКРАЇНА м. ЛЬВІВ № 30203680

"ЗАТВЕРДЖУЮ"
Проректор з наукової роботи
Національного університету
"Львівська політехніка"
_____ Іван ДЕМИДОВ
"___" _____ 2025 р.

АКТ
про використання наукових результатів
дисертаційної роботи Поберейка Петра Богдановича
представленої на здобуття наукового ступеня доктора філософії

Комісія в складі: голови комісії – начальник науково-дослідної частини д.т.н., с.н.с. Небесного Р.В. та членів комісії – завідувача кафедри СШІ Мельникової Н.І., директор ІКНІ Шаховська Н.Б., цим актом підтверджують, що результати дисертаційної роботи Поберейка Петра Богдановича, зокрема:

- удосконалений метод детекції меж сцен (SBD) на основі суміщеного аналізу гістограм у просторі HSV, абсолютних різниць пікселів та механізму уваги;
- модель інтеграції просторових та часових ознак відеофрагментів з використанням архітектури SlowFast і комбінованих дескрипторів (TSIM + OSIM);
- алгоритм формування векторів ознак відео, що поєднує геометричні характеристики об'єктів (MobileNet / EfficientNet) з LSTM-аналізом динаміки;
- модульна архітектура інформаційної системи пошуку подібних відеофрагментів у режимі реального часу,

використано у науково-дослідних роботі, що виконувалась на кафедрі систем штучного інтелекту і включено до звіту за договором «Комплексний догляд для наступного покоління» (C2020/1-8, iCare4Next) виконаної за договором № М/100-2024 від 19.07.2024 р.

Отримані автором результати використано:

при налаштуванні гіперпараметрів глибоких моделей (DCNN + LSTM) для підвищення точності Scene Boundary Detection у відеопотоках;

- при проектуванні та тестуванні модулів сегментації й екстракції ознак для класифікації даних в умовах обмежених обчислювальних ресурсів;
- при розгортанні гібридних ансамблів з метою прискорення контентно-орієнтованого пошуку;
- у процесі інтеграції клієнтського веб-інтерфейсу та серверного модуля глибокого аналізу для демонстраційної системи реального часу.

Голова комісії:
Начальник науково-дослідної частини
д.т.н., с.н.с. Роман НЕБЕСНИЙ

Члени комісії:

Директор ІКНІ Наталія ШАХОВСЬКА

Завідувач кафедри СШІ Наталія МЕЛЬНИКОВА