

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Кваліфікаційна наукова
праця на правах рукопису

ЗДОР КОСТЯНТИН АНДРІЙОВИЧ

УДК 004.94

ДИСЕРТАЦІЯ
МОДЕЛІ ТА ПРОГРАМНІ ЗАСОБИ ПІДВИЩЕННЯ ШВИДКОДІЇ
ВИЗНАЧЕННЯ ВІДЕОАТРИБУТІВ ЗА ДОПОМОГОЮ РОЗБИТТЯ
НА СЦЕНИ

121 – Інженерія програмного забезпечення

12 – Інформаційні технології

Подається на здобуття наукового ступеня доктора філософії.

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

 Здор К.А.

Наукові керівники:

ШАЛДЕНКО Олексій Вікторович, кандидат технічних наук, доцент

НЕДАШКІВСЬКИЙ Олексій Леонідович, доктор технічних наук, доцент

АНОТАЦІЯ

Здор К.А. Моделі та програмні засоби підвищення швидкодії визначення відеоатрибутів за допомогою розбиття на сцени. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії з галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2025.

Дисертаційна робота присвячена розробці науково-методичного апарату обробки відеоконтенту і розробки програмних засобів для визначення атрибутів та розбиття відео на сцени за допомогою засобів машинного навчання.

Паралельно з еволюцією виробництва контенту розвивалися методи аналізу відеоданих. Ранні методи аналізу відео були здебільшого ручними та примітивними, зосереджуючись на таких базових функціях, як виявлення руху та просте розпізнавання об'єктів. З часом ці методи кардинально еволюціонували. Впровадження алгоритмічних підходів уможливило автоматизований розбір відеопотоків на значущі сегменти, заклавши основу для пошуку та індексування на основі контенту. Фундаментальні дослідження, проілюстрували можливість вилучення просторових і часових характеристик з відеоконтенту, тим самим уможлививши більш систематичне розуміння візуальної інформації.

Досягнення в галузі штучного інтелекту та машинного навчання ще більше розвинули сферу аналізу відеоконтенту. Сучасні системи використовують глибокі нейронні мережі для розпізнавання патернів у відеоданих - від поведінкових сигналів до контекстних асоціацій. Інтеграція технологій автоматичного розпізнавання контенту в споживчі пристрої дозволила вимірювати аудиторію в реальному часі та персоналізувати

доставку контенту, що, в свою чергу, змінило стратегії реклами та дистрибуції медіа.

Виявлення сцен є актуальною задачею у сфері аналізу відеоконтенту, оскільки воно забезпечує структурну основу, яка дозволяє виявляти семантично пов'язані сегменти у відеоданих. Сегментування відео на плани і сцени - аналогічно до поділу тексту на абзаци - дає змогу виокремити часові межі сцен та організувати вміст контенту у менші структурні одиниці. Така сегментація має важливе значення для індексування та узагальнення, оскільки дозволяє як автоматизованим системам, так і користувачам ефективно орієнтуватися у великих відеоархівах.

Метою дисертації є підвищення точності та швидкодії розбиття відео на сцени шляхом розробки моделей з використанням візуальних трансформерів для відео та розробка спеціалізованих програмних засобів для зниження обчислювальних витрат при визначенні атрибутів.

Серед методів розбиття відео на сцени можна виділити традиційні, які використовують візуальні характеристики (гістограми, рівень освітлення тощо), та алгоритми на основі виділення ключових точок, як-от SIFT і SURF. Перший підхід демонструє високу ефективність для статичних сцен, але втрачає точність при аналізі динамічного відеоконтенту з короткими сценами та складними переходами. Алгоритми з виділення ключових точок забезпечують вищу точність у визначенні змін, однак їх застосування обмежене через значні обчислювальні витрати.

Сучасні методи сегментації базуються на використанні нейронних мереж, що дозволяє враховувати як візуальний, так і концептуальний контекст кадрів. Використання згорткових, рекурентних нейронних мереж і трансформерів сприяє точному визначенню змін сцен, проте ці підходи вимагають великої кількості навчальних даних і можуть мати високі вимоги до обчислювальних ресурсів. Тому виникає протиріччя, з одного боку математичні методи мають високу швидкість але низьку точність, з іншого

боку методи машинного навчання демонструють вищу точність, але можуть мати високі вимоги до обчислювальних ресурсів.

Для подолання цих недоліків застосовуються методи оптимізації, такі як прунінг, дистиляція знань та квантизація, що дозволяє прискорити роботу моделей при мінімальній втраті точності.

Розробка та вдосконалення методів виявлення сцен на основі методів машинного навчання є пріоритетним напрямком в сфері аналізу відео контенту. Методам виявлення сцен присвячені роботи зарубіжних вчених Del Fabro M., Böszörményi L., Chong-Wah Ngo, Yu-Fei Ma, Hong-Jiang Zhang, Baraldi L., Grana C, Cucchiara R. Прунінгу і оптимізації перед навчанням присвячені роботи Lee N., Ajanthan T., Frankle J., Carbin M. Розробці методів архітектурної оптимізації присвячені роботи Сінькевич О.О., Терейковський І.А., Кудін О.В., Кривохата А.Г., Howard A. G., Zhu M., Hinton G., Dean J. та інші. Дослідженням методів зниження витрат обчислювальних ресурсів займались Рувінська В.М., Тімков Ю.Ю., Струнін І.В., Прогонов Д.О. Liang T., Li B., Kong Z. Tan M., Wang Z., Frankle J., Carbin M. Han S., Pool J., Li H. та інші.

Дисертаційна робота виконана відповідно з поточними та перспективними планами наукової та науково-технічної діяльності Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» для подальшого розвитку інженерії програмного забезпечення. Дослідження тісно пов'язано з результатами науково-дослідницької роботи (НДР), в яких автор приймав особисту участь, а саме: «Методи і алгоритми оптимізації розпізнавання образів на основі методів машинного навчання» №0121U109207, що виконувалась в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» у 2021 – 2024 рр. Особисто автором в НДР запропоновано удосконалений алгоритм розбиття відео на плани використовуючи поєднання математичних алгоритмів, для виявлення

особливостей кадрів, та рекурентних нейронних мереж, для визначення зміни плану, що дозволяє зменшити кількість необхідних даних для аналізу, значно пришвидшуючи розпізнавання образів.

Наукова новизна одержаних результатів полягає в тому, що в дисертаційній роботі:

1. **Вперше розроблено архітектуру** розподіленого програмного забезпечення для визначення атрибутів на відео, **характерною особливістю** якої є оперування відеопотоками для їхнього розбиття відео на плани та сцени, **що дозволило** збільшити швидкість аналізу відеоконтенту мінімум в 2.5-3 рази.

2. **Вперше розроблено метод** для виявлення переходів планів у відеоконтенті на основі поєднання математичних підходів та рекурентних нейронних мереж, **який на відміну від існуючих методів** швидко та ефективно виділяє просторові та часові ознаки кадрів, **що дозволило** збільшити точність влучання та F1-оцінку для знаходження зміни планів досягаючи інноваційних результатів.

3. **Вперше розроблено метод** виявлення зміни сцени для відеоконтенту з використанням нейронної мережі на основі архітектури візуального трансформеру для відео з застосуванням методу прунінгу перед навчанням, **що на відміну від існуючих методів** виділяє контекстуальні особливості сцен, **що дозволило** збільшити F1-оцінку на 5.1% та пришвидшити час виконання на 10%.

4. **Набув подальшого розвитку метод** прунінгу перед навчанням для моделей архітектури візуальних трансформерів для відео, **який на відміну від існуючих методів** враховує важливість механізму «уваги» та дозволяє пришвидшити час виконання моделі на 10%.

Практичне значення одержаних результатів полягає в підвищенні точності та швидкодії аналізу відеоконтенту за допомогою розробленої архітектури розподіленого програмного забезпечення для визначення

атрибутів на відео за допомогою розбиття відео на плани та сцени, що на відміну від існуючих ефективно розподіляє обчислення та реалізує розроблені та удосконалені методи та алгоритмічне забезпечення. Реалізація удосконаленого методу прунінгу для архітектури візуального трансформера на відео дозволила натренувати нейронну мережу, яка на 10% швидша за оригінальну. Реалізація розробленого методу поєднання математичного підходу з рекурентними нейронними мережами дозволила натренувати дві нейронні мережі, які перевищують точність влучання та F1-оцінку відносно підходу AutoShot на 4.3% та 4.4% відповідно. При цьому розроблений підхід має обчислювальні вимоги у розмірі до 500 kFLOPS, що дозволяє використовувати цей підхід для розв'язання задач у реальному часі. Реалізація розробленого методу для визначення зміни сцени в відео на основі візуального трансформера для відео дозволила натренувати нейронну мережу, яка перевищує F1-оцінку відносно підходів оснований на глибоких мультимодальних мережах на 5.43%. Розроблено програмне забезпечення, яке на відміну від існуючих, дозволяє ефективно аналізувати атрибути для відео використовуючи сцени та плани отримані в режимі реального часу.

Методика дослідження та отримані результати можуть також бути використані для створення систем детальної відеоаналітики, фільтрації та пошуку по відеоатрибутах, тим самим розширюючи сучасні підходи до аналізу відеоконтенту. Дослідження може стати основою для розробки нових підходів до розбиття відео на сцени та плани, та внести вклад у зростаючий обсяг літератури з методів відеоаналізу за допомогою нейронних мереж.

Результати досліджень прийняті до впровадження в Товаристві з обмеженою відповідальністю «ВОТЧЕД» (акт від 10.02.2025р.); в навчальному процесі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (акт впровадження від 24.02.2025р.) при викладанні дисципліни «Цифрова обробка

зображень» для студентів освітньо-кваліфікаційного рівня «Магістр» спеціальності 122 «Комп'ютерні науки».

Наукові результати досліджень є внеском у розвиток теоретичних і прикладних основ розробки й дослідження науково-методичного і програмного апарату для аналізу відеоатрибутів з використанням методів машинного навчання.

Наступними перспективними дослідженнями можуть стати дослідження для вдосконалення критеріїв визначення важливості вагів, автоматичне визначення ключових кадрів для планів та сцен під час аналізу, розширення можливостей аналізу візуального трансформеру для відео.

Ключові слова: інженерія програмного забезпечення, програмні засоби, інформаційні технології, швидкодія, оптимізація, нейронна мережа, машинне навчання, штучний інтелект, обробка зображень, аналіз даних, хмарне середовище, комп'ютерна система, архітектура програмної системи, мікросервісна архітектура, виявлення зміни планів, рекурентні нейронні мережі, аналіз відеоконтенту, обробка інформації, довготривала короткочасна пам'ять (LSTM), комп'ютерний зір, сіамські нейронні мережі, розпізнавання зображень, візуальні трансформери для відео, аналіз відео, паралельна обробка, масштабованість.

ANNOTATION

Zdor K.A. Models and software tools for increasing the speed of determining video attributes using scene segmentation. – Qualification scientific work in the form of a manuscript.

Thesis for the degree of Doctor of Philosophy in the field of knowledge 12 Information Technologies in the specialty 121 Software Engineering. – National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv, 2025.

The dissertation is dedicated to the development of a scientific and methodological framework for video content processing and the development of software tools for attribute detection and video scene segmentation using machine learning techniques.

The evolution of video content as a dominant means of communication has fundamentally changed the landscape of information distribution and consumption. Early forms of visual media, from analog film to broadcast television, set the stage for the transformation that culminated in the digital revolution. Over the past few decades, video content has not only increased in quantity and accessibility, but also in its ability to engage audiences in interactive and multifaceted ways.

In parallel with the evolution of content production, video data analysis methods have evolved. Early video analysis methods were mostly manual and primitive, focusing on basic functions such as motion detection and simple object recognition. Over time, these methods have evolved dramatically. The introduction of algorithmic approaches has enabled the automated parsing of video streams into meaningful segments, laying the foundation for content-based search and indexing. Fundamental research has illustrated the possibility of extracting spatial and temporal features from video content, thereby enabling a more systematic understanding of visual information.

Advances in artificial intelligence and machine learning have further revolutionized video content analysis. Modern systems use deep neural networks to recognize patterns in video data - from behavioral signals to contextual associations. The integration of automatic content recognition technologies into consumer devices has enabled real-time audience measurement and personalized content delivery, which, in turn, has changed advertising and media distribution strategies.

Scene detection is relevant to video content analysis, as it provides a structural framework that transforms continuous video streams into discrete, semantically coherent segments. Segmenting video into frames and scenes—analogue to dividing text into paragraphs—allows us to isolate the temporal boundaries of scenes and organize the content into smaller structural units. Such segmentation is important for indexing and summarization, as it allows both automated systems and users to navigate effectively in large video archives.

The dissertation aims to increase the accuracy and speed of video scene segmentation by developing models using visual transformers for video and developing special software tools to reduce computational costs when determining attributes.

Modern segmentation methods are based on neural networks, allowing the frames' visual and conceptual context to be considered. The use of convolutional, recurrent neural networks and transformers contributes to the accurate detection of scene changes. Still, these approaches require a large amount of training data and can have high requirements for computational resources. Therefore, a contradiction arises: on the one hand, mathematical methods have high speed but low accuracy; on the other hand, machine learning methods demonstrate higher accuracy but can have high requirements for computational resources.

Among the methods for dividing video into scenes, one can distinguish traditional ones that use visual characteristics (histograms, lighting level, etc.) and algorithms based on keypoint extraction, such as SIFT and SURF. The first approach demonstrates high efficiency for static scenes but loses accuracy when analyzing

dynamic video content with shorter scenes and complex transitions. Keypoint extraction algorithms provide higher accuracy in detecting changes, but their application is limited due to significant computational costs.

To overcome these shortcomings, optimization methods such as pruning, knowledge distillation, and quantization are used. These methods allow for the speeding up of model operation with minimal loss of accuracy.

Developing and improving scene detection methods based on machine learning methods is a priority area in video content analysis. The works of foreign scientists are devoted to scene detection methods: Del Fabro M., Böszörményi L., Chong-Wah Ngo, Yu-Fei Ma, Hong-Jiang Zhang, Baraldi L., Grana C, Cucchiara R. The works of Lee N., Ajanthan T., Frankle J., and Carbin M. are devoted to pruning and optimization before training. The works of Sinkevich O.O., Tereykovsky I.A., Kudin O.V., Kryvokhata A.G., Howard A. G., Zhu M., Hinton G., Dean J., and others are devoted to developing architectural optimization methods. The research of strategies for reducing the cost of computing resources was carried out by Ruvinska V.M., Timkov Yu.Yu., Strunin I.V., Progonov D.O. Liang T., Li B., Kong Z., Tan M., Wang Z., Frankle J., Carbin M., Han S., Pool J., Li H., et al.

The dissertation work was carried out by the current and prospective plans of scientific and scientific-technical activities of the National Technical University of Ukraine, "Igor Sikorsky Kyiv Polytechnic Institute," for further software engineering development. The study is closely related to the results of scientific and research work (R&D), in which the author personally participated, namely: "Methods and algorithms for optimizing pattern recognition based on machine learning methods" No. 0121U109207, which was carried out at the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" in 2021 - 2024. The author has personally proposed an improved algorithm for segmenting video into shots by combining mathematical algorithms for detecting frame features with recurrent neural networks for identifying shot changes. This approach reduces

the amount of data required for analysis and significantly accelerates image recognition.

The scientific novelty of the results obtained is that in the dissertation work:

1. For the first time, the architecture of distributed software was developed to identify attributes in the video by splitting the video into frames and scenes which, unlike existing architectures, effectively distributes the computing, which allowed to increase the speed of video content analysis.

2. For the first time, a method was developed to detect shot transitions in video content based on a combination of mathematical approaches and recurrent neural networks. Unlike existing methods, this method quickly and effectively distinguishes the spatial and temporal characteristics of frames, allowing the accuracy of hit and F1-score to find a change of shots to increase, achieving innovative results.

3. For the first time, the method of detecting a scene change in video content using a neural network based on a visual transformer architecture for video using a pruning method before training, which, unlike existing approaches, identifies the contextual features of scenes which allowed to increase F1-score by 5.1% and speed up runtime by 10%.

4. The method of pruning before training was first developed for visual transformers architecture models for video that, unlike existing approaches, takes into account the importance of the "attention" mechanism and allows you to speed up the model execution.

The practical significance of the results obtained lies in their application to increase the accuracy and speed of video content analysis. Implementing the improved pruning method for the architecture of the visual transformer for video allowed us to train a neural network that is 10% faster than the original one. Implementing the developed method of combining the mathematical approach with recurrent neural networks allowed us to train two neural networks that exceeded the accuracy of the hit and F1-score relative to the AutoShot approach by 4.3% and

4.4%, respectively. At the same time, the developed approach has computational requirements of up to 500 kFLOPS, which allows us to use this approach to solve problems in real-time. Implementing the developed method for determining a scene change in a video based on the visual transformer for video allowed us to train a neural network that exceeds the F1-score relative to approaches based on deep multimodal networks by 5.43%.

The research methodology and obtained results can also be used for developing systems for detailed video analytics, filtering, and searching by video attributes, thereby expanding modern approaches to video content analysis. This study may serve as a foundation for developing new methods of video scene and shot segmentation and contribute to the growing body of literature on video analysis using neural networks.

The research results have been approved for implementation at WATCHED LLC (implementation act dated 10.02.2025) and in the educational process of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (implementation act dated 24.02.2025) as part of the course "Digital Image Processing" for Master's degree students in the 122 "Computer Science" specialization.

The scientific results of the research contribute to the development of theoretical and applied foundations for the development and research of scientific, methodological, and software tools for analyzing video attributes using machine learning methods.

Future research directions may include improving the criteria for determining the importance of weights, automating the identification of key frames for shots and scenes during analysis, and expanding the capabilities of visual transformers for video analysis.

Keywords: software engineering, software tools, information technology, performance, optimization, neural network, machine learning, artificial intelligence, image processing, data analysis, cloud environment, computer system, software

system architecture, microservices architecture, scene change detection, recurrent neural networks, video content analysis, information processing, long short-term memory (LSTM), computer vision, Siamese neural networks, image recognition, visual transformers for video, video analysis, parallel processing, scalability.

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Здор К. А., Шалденко О. В. Нейро-математичний підхід для виявлення змін планів у відеопослідовностях. Зв'язок. 2024. №6(172). С. 91-97.
2. Melnychenko, A., Zdor K. Incorporating attention score to improve foresight pruning on transformer models. Computer Science and Applied Mathematics, 2023, №2, P.18-22.
3. Melnychenko, A., Zdor, K. Efficiency of supplementary outputs in siamese neural networks. Advanced Information Systems, 2023, Volume 7, №3, P. 49–53.
4. Zdor K., Shaldenko O., Nedashkivskiy O., Melnychenko A., Leveraging ViViT transformers and foresight pruning for scalable scene change detection on distributed architecture, Зв'язок. 2025. №1. Р. 3-8.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

5. Здор К. А., Шалденко О.В. Концепція обробки зображення на основі багатозадачних сіамських нейронних мереж, XX Міжнародна науково-практична конференція молодих вчених і студентів, м. Київ, 25–28 квітня 2023 року, с. 210-211
6. Melnychenko, A., Zdor, K. Applying classification and regression supplementary output in siamese neural network using fashion MNIST and plantvillage datasets, VII Міжнародна науково-практична конференція «Modern problems of science, education and society», 11-13 вересня 2023 Київ, Україна, С. 126-129.

7. Melnychenko, A., & Zdor, K. Applying classification and regression supplementary outputs in siamese neural network using plantvillage dataset, I Міжнародна науково-практична конференція «Current challenges of science and education», 18-20 вересня 2023, Берлін, Німеччина. С. 79-82.

8. Melnychenko A., Zdor K. Applying classification and regression supplementary output in siamese neural network using fashion MNIST and plantvillage datasets, X Міжнародна науково-практична конференція «Innovations and prospects in modern science», 25-27 вересня 2023, Стокгольм, Швеція. С. 87-92.

9. Мельниченко А., Здор К. Збільшення ефективності оптимізації моделей архітектури ViT перед навчанням шляхом включення активацій механізму самоуваги, I міжнародна науково-практична конференція «Сучасні аспекти інженерії програмного забезпечення», 14 грудня 2023, Київ, Україна.

10. Мельниченко А.В., Здор К.А. Врахування механізмів самоуваги при прунінгу моделей нейронних мереж Vision Transformer. Збірник матеріалів III Міжнародної науково-технічної конференції «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку», 30 листопада 2023 року, Київ, Україна. С. 214 – 215.

ЗМІСТ

СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	20
ВСТУП.....	21
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОПТИМІЗАЦІЇ ВИЗНАЧЕННЯ ВІДЕОАТРИБУТІВ.....	29
1.1. Методи визначення атрибутів у відео.....	29
1.1.1. Методи сегментації сцен у відео.....	30
1.1.2. Алгоритми класифікації сцен	33
1.1.3. Методи визначення відеоатрибутів.....	34
1.2. Архітектура моделей для роботи з відео	36
1.2.1. Рекурентні нейронні мережі	36
1.2.2. Згорткові нейронні мережі.....	38
1.2.3. Моделі на основі трансформерів.....	42
1.2.4. Гібридні архітектури	45
1.3. Сучасні підходи до оптимізації моделей для обробки відео	47
1.3.1. Використання прунінгу для оптимізації моделей	48
1.3.2. Квантування моделей	49
1.3.3. Інтеграція дистиляції знань для підвищення швидкодії..	51
1.4. Аналіз вимог до програмного забезпечення та постановка наукового завдання	52
1.5. Висновки до розділу 1	55
РОЗДІЛ 2. ВДОСКОНАЛЕННЯ МЕТОДІВ СЦЕННОГО РОЗБИТТЯ ТА АНАЛІЗУ ВІДЕОАТРИБУТІВ	57
2.1. Алгоритми розбиття відео на сцени.....	57

2.1.2. Використання глибоких нейронних мереж	61
2.1.3. Перспективні напрямки розвитку	62
2.1.4. Критерії вибору архітектури	64
2.2. Архітектура візуальних трансформерів для відео при визначенні зміни сцен	67
2.2.1. Архітектура ViViT	67
2.2.2. Застосуванні візуальних трансформерів для відео при визначенні зміни сцен	71
2.2.3. Ефективність використання додаткових виходів моделі під час навчання моделей.	72
2.2.4. Визначення зміни планів на відео з метою визначення потенційних країв сцен та ключових кадрів для аналізу.	81
2.3. Метод розбиття відео на сцени за допомогою візуального трансформеру для відео	91
2.3.1. Вимоги до передобробки даних та перевірки результатів експериментів	92
2.3.2. Підвищення точності розпізнавання виявлення зміни сцен з використанням візуальних трансформерів для відео.	95
2.4. Висновки до розділу 2	97
РОЗДІЛ 3. ОПТИМІЗАЦІЯ МОДЕЛЕЙ АРХІТЕКТУРИ ВІЗУАЛЬНИХ ТРАНСФОРМЕРІВ ДЛЯ ВІДЕО	99
3.1. Методи оптимізації моделей трансформерів для відео	99
3.1.1. Скорочення складності моделей архітектури трансформерів	99
3.1.2. Прунінг та скорочення параметрів	101

3.1.3. Прунінг архітектур трансформер	104
3.1.4. Вимоги до програмного забезпечення та обчислювальних ресурсів.....	107
3.2. Алгоритм прунінгу	110
3.2.1. Формати зберігання розріджених матриць	112
3.2.2. Напів-структурована розрідженість формату 2:4	114
3.2.3. Алгоритмічне забезпечення оцінки важливості вагів для напів-структурованої розрідженості	117
3.2.4. Експериментальна оцінка удосконаленого методу оптимізації	121
3.3. Висновки до розділу 3	123
РОЗДІЛ 4. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ВІДЕОАТРИБУТІВ	125
4.1. Засоби розроблення для програмного забезпечення визначення відеоатрибутів.....	129
4.2. Програмне забезпечення для реалізації та впровадження нейронних мереж.....	130
4.3. Програмне забезпечення модулю для проведення прунінгу перед тренуванням.....	136
4.4. Програмне забезпечення модулю для розбиття відео на плани	140
4.5. Програмне забезпечення модулю для визначення сцен	143
4.6. Архітектура розподіленого програмного забезпечення для визначення відеоатрибутів.....	145

4.7. Програмне забезпечення модулю збереження та аналізу даних	
.....	150
4.8. Висновки до розділу 4	151
ВИСНОВКИ	152
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	155
Додаток А	171
Додаток Б	173
Додаток В	175

СПИСОК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

Прунінг	– процес обрізки нейронної мережі шляхом видалення частини параметрів
ReLU	– випрямлений лінійний вузол
$L(W)$	- функція втрат
W	- ваги (параметри) нейронної мережі
W_{enc}	- ваги (параметри) кодера
W_{dec}	- ваги (параметри) декодера
Q_i	- вектор запитів
K_i	- вектор запитів
V_i	- вектор значень
Softmax	- нормована експоненційна функція

ВСТУП

Актуальність теми.

Еволюція відеоконтенту, як домінуючого засобу комунікації, докорінно змінила ландшафт поширення та споживання інформації. Ранні форми візуальних медіа, від аналогової кіноплівки до ефірного телебачення, підготували ґрунт для трансформації, кульмінацією якої стала цифрова революція. За останні кілька десятиліть відеоконтент не лише збільшився у кількості та доступності, але й у своїй здатності залучати аудиторію в інтерактивний та багатогранний спосіб.

У своїх ранніх втіленнях відеоконтент вироблявся і поширювався переважно через традиційні канали мовлення, які функціонували як односпрямований потік інформації від кількох централізованих джерел до пасивної аудиторії. З появою цифрових відеотехнологій і таких платформ, як YouTube, виробництво і споживання відеоконтенту трансформувалось, надавши користувачам можливість створювати, поширювати і взаємодіяти з візуальними медіа в безпрецедентних масштабах. Сучасні платформи зараз можуть налічувати мільярди глядачів щомісяця, і ця статистика підтверджує глибоке проникнення відео в повсякденну комунікацію [1].

Паралельно з еволюцією виробництва контенту розвивалися методи аналізу відеоданих. Ранні методи аналізу відео були здебільшого ручними та примітивними, зосереджуючись на таких базових функціях, як виявлення руху та просте розпізнавання об'єктів. З часом ці методи кардинально еволюціонували. Впровадження алгоритмічних підходів уможливило автоматизований розбір відеопотоків на значущі сегменти, заклавши основу для пошуку та індексування на основі контенту. Фундаментальні дослідження, проілюстрували можливість вилучення просторових і часових характеристик

з відеоконтенту, тим самим уможлививши більш систематичне розуміння візуальної інформації [2].

Досягнення в галузі штучного інтелекту та машинного навчання дозволили ще більше вдосконалити методи аналізу відеоконтенту. Сучасні системи використовують глибокі нейронні мережі для розпізнавання патернів у відеоданих - від поведінкових сигналів до контекстних асоціацій [3,4]. Інтеграція технологій автоматичного розпізнавання контенту в споживчі пристрої дозволила вимірювати аудиторію в реальному часі та персоналізувати доставку контенту, що, в свою чергу, змінило стратегії реклами та дистрибуції медіа.

Вплив цих аналітичних досягнень виходить за межі технічної ефективності; вони також змінили саму природу інформаційного простору. Автоматизований відеоаналіз уможливив кураторство і пошук величезних сховищ візуальних даних, сприяючи тим самим новим формам наукової комунікації та залучення громадськості. Дослідження, що вивчає моделі цитування онлайн-відео, продемонструвало, що відеоконтент все частіше визнається легітимним і впливовим засобом в академічному дискурсі [5]. Такі висновки ілюструють, що можливість індексувати, шукати та взаємодіяти з відеоконтентом не лише покращила наше розуміння візуальної інформації, але й спричинила значні зміни у поширенні досліджень та новин.

Виявлення сцен є важливим кроком у сфері аналізу відеоконтенту, оскільки воно забезпечує структурну основу, яка перетворює безперервні відеопотоки на дискретні, семантично зв'язні сегменти. Сегментуючи відео на кадри і сцени - аналогічно до поділу тексту на абзаци - цей процес дає змогу виокремити часові межі сцен та організувати вміст контенту у менші структурні одиниці. Така сегментація має важливе значення для індексування та узагальнення, оскільки дозволяє як автоматизованим системам, так і користувачам ефективно орієнтуватися у великих відеоархівах.

Важливість виявлення сцен поширюється на його роль у полегшенні семантичного аналізу вищого рівня. Коли відео розбите на окремі сцени, алгоритми можуть точніше аналізувати кожен сегмент на предмет контекстуальних підказок, взаємозв'язків між об'єктами і переходів у розповіді, та інших атрибутів.

Дисертаційна робота виконана відповідно з поточними та перспективними планами наукової та науково-технічної діяльності Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» для подальшого розвитку інженерії програмного забезпечення. Дослідження тісно пов'язано з результатами науково-дослідницької роботи (НДР), в яких автор приймав особисту участь, а саме: «Методи і алгоритми оптимізації розпізнавання образів на основі методів машинного навчання» №0121U109207, що виконувалась в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського» у 2021 – 2024 рр. Особисто автором в НДР запропоновано удосконалений алгоритм розбиття відео на плани використовуючи поєднання математичних алгоритмів, для виявлення особливостей кадрів, та рекурентних нейронних мереж, для визначення зміни плану, що дозволяє зменшити кількість необхідних даних для аналізу, значно пришвидшуючи розпізнавання образів.

Тому актуальною є наукова задача розробки моделей та програмних засобів для підвищення швидкодії визначення атрибутів у відео за допомогою розбиття на плани та сцени з використанням візуальних трансформерів та застосуванням розподіленої архітектури.

Мета і завдання дослідження.

Метою дисертації є підвищення точності та швидкодії розбиття відео на сцени шляхом розробки моделей з використанням візуальних трансформерів для відео та розробка спеціальних програмних засобів для зниження обчислювальних витрат при визначенні атрибутів.

Для досягнення мети в дисертації вирішено такі **наукові завдання**:

- Виконано аналіз існуючих методів розбиття відео на плани.
- Виконано аналіз існуючих методів розбиття відео на сцени.
- Досліджено швидкодію існуючих методів.
- Досліджено ефективність існуючих методів при застосуванні їх для сучасного відеоконтенту.
- Розроблено новий метод розбиття відео на плани за допомогою нейронних мереж.
- Розроблено новий метод розбиття відео на сцени використовуючи плани та візуальні трансформери для відео.
- Розроблено архітектуру розподіленого програмного забезпечення для визначення атрибутів на відео за допомогою розбиття відео на плани та сцени, що на відміну від існуючих ефективно розподіляє обчислення, що дозволило збільшити швидкість аналізу відеоконтенту.

Об'єкт досліджень: Процес обробки відеоданих з використанням розбиття відео на сцени за допомогою візуальних трансформерів для відео та оптимізація програмних засобів при визначенні атрибутів у відео.

Предмет досліджень: Методи розбиття відео на сцени за допомогою візуальних трансформерів для відео та підвищення швидкодії програмних засобів з метою зниження обчислювальних витрат при аналізі відеоданих.

Методи дослідження. Для досягнення поставленої в роботі мети використано методи алгоритмічного та порівняльного аналізу (для визначення актуальності та постановки наукового завдання дисертаційної роботи). Для оцінки роботи алгоритмів розбиття на сцени та збільшення швидкодії нейронних мереж були використані методи дослідження на основі математичної статистики і машинного навчання. Для дослідження моделей для вирішення задач комп'ютерного зору було використано методи обробки сигналів і алгоритми ітеративної оптимізації.

Наукова новизна одержаних результатів полягає в тому, що в дисертаційній роботі:

1. **Вперше розроблено архітектуру** розподіленого програмного забезпечення для визначення атрибутів на відео, **характерною особливістю** якої є оперування відеопотоками для їхнього розбиття відео на плани та сцени, **що дозволило** збільшити швидкість аналізу відеоконтенту мінімум в 2.5-3 рази.

2. **Вперше розроблено метод** для виявлення переходів планів у відеоконтенті на основі поєднання математичних підходів та рекурентних нейронних мереж, **який на відміну від існуючих методів** швидко та ефективно виділяє просторові та часові ознаки кадрів, **що дозволило** збільшити точність влучання та F1-оцінку для знаходження зміни планів досягаючи інноваційних результатів.

3. **Вперше розроблено метод** виявлення зміни сцени для відеоконтенту з використанням нейронної мережі на основі архітектури візуального трансформеру для відео з застосуванням методу прунінгу перед навчанням, **що на відміну від існуючих методів** виділяє контекстуальні особливості сцен, **що дозволило** збільшити F1-оцінку на 5.1% та пришвидшити час виконання на 10%.

4. **Набув подальшого розвитку метод** прунінгу перед навчанням для моделей архітектури візуальних трансформерів для відео, **який на відміну від існуючих методів** враховує важливість механізму «уваги» та дозволяє пришвидшити час виконання моделі на 10%.

Практичне значення дисертаційного дослідження полягає в підвищенні точності та швидкодії аналізу відеоконтенту за допомогою розробленої архітектури розподіленого програмного забезпечення для визначення атрибутів на відео за допомогою розбиття відео на плани та сцени, що на відміну від існуючих ефективно розподіляє обчислення та реалізує розроблені та удосконалені методи та алгоритмічне забезпечення.

Реалізація удосконаленого методу прунінгу для архітектури візуального трансформера на відео дозволила натренувати нейронну мережу, яка на 10% швидша за оригінальну. Реалізація розробленого методу поєднання математичного підходу з рекурентними нейронними мережами дозволила натренувати дві нейронні мережі, які перевищують точність влучання та F1-оцінку відносно підходу AutoShot на 4.3% та 4.4% відповідно. При цьому розроблений підхід має обчислювальні вимоги у розмірі до 500 kFLOPS, що дозволяє використовувати цей підхід для розв'язання задач у реальному часі. Реалізація розробленого методу для визначення зміни сцени в відео на основі візуального трансформера для відео дозволила натренувати нейронну мережу, яка перевищує F1-оцінку відносно підходів оснований на глибоких мультимодальних мережах на 5.43%. Розроблено програмне забезпечення, яке на відміну від існуючих, дозволяє ефективно аналізувати атрибути для відео використовуючи сцени та плани отримані в режимі реального часу

Методика дослідження та отриманні результати можуть також бути використані для створення систем детальної відеоаналітики, фільтрації та пошуку по відеоатрибутах, тим самим розширюючи сучасні підходи до аналізу відеоконтенту. Дослідження може стати основою для розробки нових підходів до розбиття відео на сцени та плани, та внести вклад у зростаючий обсяг літератури з відеоаналізу за допомогою нейронних мереж.

Результати досліджень прийняті до впровадження в Товаристві з обмеженою відповідальністю «ВОТЧЕД» (акт від 10.02.2025.); в навчальному процесі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (акт впровадження від 24.02.2025р.) при викладанні дисципліни «Цифрова обробка зображень» для студентів освітньо-кваліфікаційного рівня «Магістр» спеціальності 122 «Комп'ютерні науки».

Особистий внесок здобувача. Дисертаційна робота є самостійно виконаною науковою працею. Всі представлені наукові результати, приклади

та експериментальні розрахунки, викладені у дисертації, одержані здобувачем одноосібно. У наукових роботах, що опубліковані у співавторстві, в дисертаційній роботі використані лише ті результати, які становлять індивідуальний внесок автора. У друкованих працях, опублікованих у співавторстві, здобувачеві належать:

Удосконалений метод прунінгу для моделей архітектури трансформерів. Зазначений метод дозволяє виключати результати уваги моделі та виходів шарів трансформерів до розрахунку важливості вагів. Завдяки цьому, при застосуванні удосконаленого методу прунінгу до моделі архітектури візуального трансформеру для відео вдалося покращити швидкість виконання моделі на 10% при збереженні оригінальної точності моделі.

Виявлено додаткових виходів моделі для архітектури сіамських нейронних мереж, які мають на меті вирішувати частину основної задачі мережі, з метою акцентування уваги моделі на конкретні особливості даних.

Розроблено метод для поєднання математичних підходів та нейронних мереж для задачі визначення зміни планів на відео, що дозволило перевершити точність розпізнавання існуючих підходів, при цьому зберігаючи високу точність обчислень.

Розроблено метод визначення зміни сцен за допомогою моделі архітектури візуального трансформеру для відео який перевищує F1-оцінку відносно підходів оснований на глибоких мультимодальних мережах на 5.43%, та дозволяє ефективно розширяти існуючі набори даних для визначення зміни сцен шляхом застосування розробленого методу для визначення зміни планів.

Апробація матеріалів дисертації. Результати дисертаційного дослідження апробовано на міжнародних науково-практичних конференціях та семінарах:

– XX Міжнародна науково практична конференція молодих вчених і студентів, м. Київ, 25–28 квітня 2023 року, с. 210-211

- VII Міжнародна науково-практична конференція «Modern problems of science, education and society», 11-13 вересня 2023 Київ, Україна, С. 126-129.
- I Міжнародна науково-практична конференція «Current challenges of science and education», 18-20 вересня 2023, Берлін, Німеччина. С. 79-82.
- X Міжнародна науково-практична конференція «Innovations and prospects in modern science», 25-27 вересня 2023, Стокгольм, Швеція. С. 87-92.
- I міжнародна науково-практична конференція «Сучасні аспекти інженерії програмного забезпечення», 14 грудня 2023, Київ, Україна.
- Збірник матеріалів III Міжнародної науково-технічної конференції «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку», 30 листопада 2023 року, Київ, Україна. С. 214 – 215.

Публікації. За результатами досліджень опубліковано 3 наукові праці. Основні наукові положення викладено в 3 наукових статтях [6,7,8,149] у спеціалізованих фахових виданнях України. Із них одна наукова стаття [7] опублікована у періодичному видання, що входить до наукометричної бази SCOPUS. За матеріалами виступів на науково-технічних конференціях опубліковано 6 тез доповідей [143-148].

Структура і обсяг роботи. Дисертація складається зі вступу, 4 розділів, висновків, 3 додатків та списку використаних джерел із 149 найменувань на 15 сторінках. Повний обсяг дисертації складає 178 сторінок, з них 154 сторінок основного тексту.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ОПТИМІЗАЦІЇ ВИЗНАЧЕННЯ ВІДЕОАТРИБУТІВ

1.1. Методи визначення атрибутів у відео

Задача аналізу відеоатрибутів є важливою частиною багатьох систем, пов'язаних з моніторингом, відеоспостереженням, управлінням транспортними засобами та аналізом мультимедійних даних. Так як кількість створюваного відеоконтенту постійно зростає, то виникає потреба в збільшенні ефективності аналізу відеоатрибутів. При цьому відеоаналіз включає в себе багато важливих задач, таких як сегментація сцен, класифікація та сегментація об'єктів, виявлення активності, руху та подібні [9, 10, 11].

Для збільшення ефективності визначення атрибутів використовують різні підходи такі як пропуск кадрів, виділення ключових кадрів, застосування методів квантування, прунінгу та дистиляції знань. При зменшенні кількості кадрів для аналізу розбивають відео на невеликі сцени, використовуючи гістограми, оптичну обробку або прості моделі, засновані на математичному аналізі [12]. Іншим варіантом є пропускання кадрів. При цих підходах виникає проблема потенційної втрати важливої інформації, що може призвести до неточного аналізу і в подальшому невірної аналітики. Розділяти відео на сцени можна також за допомогою нейронних мереж, які довели свою ефективність та точність, але натомість потребують більших обчислювальних потужностей [13]. Також для збільшення ефективності намагаються використовувати оптимізовані моделі для визначення атрибутів. Такі моделі працюють швидше, мають менші вимоги до системи, на яких відбуваються обчислення, але натомість вони втрачають точність, і розробка таких моделей потребує додаткового часу. Також збільшення ефективності розпізнавання можливе за рахунок збільшення обчислювальних можливостей системи, на якій відбуваються розрахунки, що призводить до збільшення витрат на використання такої системи.

1.1.1. Методи сегментації сцен у відео

Методи сегментації сцен у відео відіграють важливу роль при аналізі сучасного відеоконтенту, так як з їх допомогою стає можливим контролювати аналіз відео та, відокремлюючи логічні межі сцен у відео, передавати на аналіз лише важливу інформацію. Використання сцен застосовується в багатьох сферах аналізу відеоконтенту, включаючи відеоспостереження, мультимедійні системи, автоматичне створення метаданих для аналізу і подібне. У випадку з відеоспостереженням це дозволяє ефективно аналізувати потік даних та при зміні сцени викликати моделі, які потребують великих обчислювальних ресурсів, для детального аналізу відеопотоку та виявлення аномалій. У мультимедійних системах це дає змогу сегментувати контент більш структуровано з подальшою можливістю деталізованого аналізу. Наприклад, використовуючи розбиття на сцени у задачах пов'язаних з навчанням, користувач може швидко знаходити важливі для себе фрагменти відео, робити систематизацію контенту, та знаходити інформацію, що всередині певної сцени або набору сцен [14]. Також, при генеруванні метаданих, використання сцен стає неймовірно важливим для деталізованого аналізу, так як кількість контенту, що одночасно завантажується, є надзвичайно великою, і модерація цього контенту стає дуже складною задачею. При розбитті відео на сцени під час завантаження відео на платформу можливо відразу запускати на кожну сцену необхідні моделі для визначення атрибутів, що дозволить додавати розмітку на відео, тегувати кожну сцену, робити виділення головних особливостей відео та на основі цього проводити автоматичну модерацію контенту, що в майбутньому дозволить значно покращити пошук контенту та виявляти особливі моменти для модерування.

Методи сегментації сцен можна поділити на три основні напрямки, а саме: традиційні методи, методи на основі ознак та моделі глибокого навчання.

Традиційні методи фокусуються на отриманні характеристик зображення, таких як гістограми, рівень освітлення та подібне. За допомоги використання даних характеристик відбувається наступне порівняння кадрів для визначення зміни сцени. Цей підхід демонструє гарну ефективність для статичних сцен, але при аналізі складного відеоконтенту продуктивність цього підходу значно зменшується [15]. Аналізуючи тенденції в сучасному відеоконтенті, можна побачити, що переходи між сценами та планами стають більш складними, а тривалість сцен стає меншою, в результаті чого цей підхід стає неефективним для аналізу більшості сучасного відеоконтенту.

Іншим підходом до вирішення цієї задачі є використання алгоритмів на основі виділення ключових точок. Класичними прикладами таких алгоритмів є SIFT та SURF. Вони здатні враховувати локальні особливості кадрів, що забезпечує вищу точність визначення зміни сцен та планів для динамічних відео. Проте ці підходи мають потребу у великій кількості обчислювальних ресурсах, що значно обмежує їх застосування при масштабуванні та обробці великих об'ємів даних [12].

Методи на основі використання нейронних мереж є найсучаснішим підходом до визначення сцен та мають набагато більшу точність у порівнянні з попередніми методами. Основною перевагою застосування нейронних мереж для таких задач є можливість захоплення контексту та порівняння кадрів як на візуальному, так і на концептуальному рівнях. В залежності від архітектури нейронної мережі це може досягатися шляхом визначення ієрархічної структури ознак кадрів за допомогою згорткових нейронних мереж, врахування просторово-часових залежностей у рекурентних нейронних мережах чи використання механізмів уваги в нейронних мережах типу трансформер. Завдяки цьому моделі можуть дуже точно визначати зміни сцен та планів, основувшись на візуальних атрибутах та концептуальному порівнянні. Недоліком використання нейронних мереж є їх швидкодія та потреба в великій кількості навчальних даних [15, 16].

Хоча швидкодія нейронних мереж є значним недоліком цього підходу, існують шляхи для їх прискорення. Такі технології, як прунінг, дистиляція знань чи квантування, дозволяють значно прискорити нейронні мережі з мінімальною втратою точності.

Проблема з великої кількості даних для навчання нейронних мереж є також значним фактором, що сповільнює пошук рішень на основі нейронних мереж. Проте з часом це може стати важливою перевагою при застосуванні нейронних мереж, так як при зростанні кількості доступних даних для аналізу буде з'являтися можливість використовувати глибші моделі з більшою кількістю параметрів, підвищуючи точність [17].

Також при визначенні зміни сцен та планів у відео є проблема зі складними переходами між сценами, зміною умов освітлення, раптовими рухами камери та спалахами, що значно ускладнює процес визначення сцен. Одним із перспективних напрямків для вирішення цієї проблеми є інтеграція аналізу звуку у відео. Додавання звуку до аналізу дозволить аналізувати зміни в музикальній темі, наявність голосів, репліки, що говорять персонажі. Використовуючи цю інформацію разом із візуальною складовою відео, можна буде підвищити ефективність визначення складних переходів між сценами та планами [18, 19].

Також важливо зауважити потребу в розробці ефективної системи метрик для вимірювання точності розбивки відео на сцени та плани, а також розробити уніфікований підхід для визначення сцени. Ця проблема сильно проявляється при аналізі датасету BBC Planet Earth, так як він містить анотації розбивки на сцени від 5-ти різних людей, які значно відрізняються одна від одної. Це дає розуміння, що поняття сцени наразі є доволі суб'єктивним і потребує додаткових досліджень з метою уніфікації цих знань.

1.1.2. Алгоритми класифікації сцен

Класифікація сцен є важливою частиною при визначенні сцен у відео, так як дозволяє автоматично визначати тип та категорію сцени, а також поєднувати сцени, пов'язані між собою за контекстом. Застосування класифікації сцен є критичною задачею в системах відеоспостереження, мультимедійному пошуку, аналізі відеоконтенту та інших застосунках. Цей крок дозволяє автоматично визначати, які моделі для детального аналізу необхідно використовувати, та робити базове групування сцен за загальним контекстом [20].

Класичні методи класифікації сцен можуть використовувати виділення ознак за допомогою дескрипторів, таких як гістограми направлених градієнтів, чи глобальних візуальних характеристик, таких як текстур чи кольорові шаблони. Хоча ці підходи прості у застосуванні і швидкі, вони мають складності з визначенням складних сцен із великою кількістю динамічних об'єктів чи нестандартним освітленням через неможливість визначати контекст [21].

Альтернативним підходом є застосування нейронних мереж, а саме згорткових нейронних мереж та візуальних трансформерів. Архітектури згорткових нейронних мереж, таких як ResNet, EfficientNet та їх подальші модифікації, дозволяють автоматично виділяти та в подальшому аналізувати просторові залежності, що значно підвищує точність класифікації. Також важливою особливістю таких нейронних мереж, що дозволяє ефективно виділяти ознаки, є використання блоків з пропуском з'єднанням (residual blocks), які детально аналізують кадри без втрати інформації, що є важливим при роботі з складними відео [22].

Моделі на основі архітектури трансформер також дозволили покращити процес аналізу зображень та відкрили нові можливості для класифікації сцен. Варіація цієї архітектури для відео дає можливість аналізувати не тільки одне

зображення, як у згорткових нейронних мережах, а й серію кадрів, як у рекурентних нейронних мережах, що дозволяє цій архітектурі одночасно враховувати просторові залежності в кожному кадрі та часові залежності між кадрами завдяки механізму самоуваги, що значно підвищує визначення концепту в серії кадрів та забезпечує більшу точність класифікації. Це робить архітектуру візуального трансформеру для відео ефективною у вирішенні таких задач, як аналіз відеоконтенту, розбиття на сцени, визначення складних залежностей у відео, враховуючи зміни руху об'єктів та фону [23]

1.1.3. Методи визначення відеоатрибутів

Визначення відеоатрибутів дозволяє отримати велику кількість інформації, яка може бути використана для аналізу, та вибір необхідних відеоатрибутів залежить від поставленої задачі. Найбільш розповсюдженими атрибутами у відеоаналізі є атрибути об'єктів на відео. Для них можна виділити ряд підзадач, що включають розпізнавання об'єктів, виявлення їх точного положення та контурів, відстеження їх руху, визначення пози та жестів людей, визначення дій, які виконують люди чи об'єкти [24, 25]. Маючи ці об'єкти, можна отримати велику кількість корисної інформації, яку далі можна використовувати для аналізу, тегування чи прийняття рішень.

Якщо говорити про моделі прийняття рішень, що можуть використовуватись у камерах відеоспостереження чи в системах наведення дронів, то важливим може бути визначення об'єктів, людей, їх поведінки, жестів, нестандартної поведінки. Також важливими атрибутами можуть бути класифікація типу середовища, освітлення та погодних умов для динамічного налаштування камери для кращої якості зображення. Атрибути руху, такі як визначення швидкості руху, напрямку та прогнозування траєкторії, можуть бути критичними задачами для ефективної взаємодії з об'єктом [26,27].

При аналізі відеоконтенту з метою отримання атрибутів для аналізу та тегування відео до більшості попередніх атрибутів можна додати виявлення розпізнавання емоцій, аналіз міміки, взаємодії між людьми та об'єктами [28]. Також визначення атрибутів може допомогти у визначенні соціальних та демографічних характеристик, таких як вік, стать, одяг та інші. Приклад атрибутів для відео продемонстровано на рисунку 1.1.

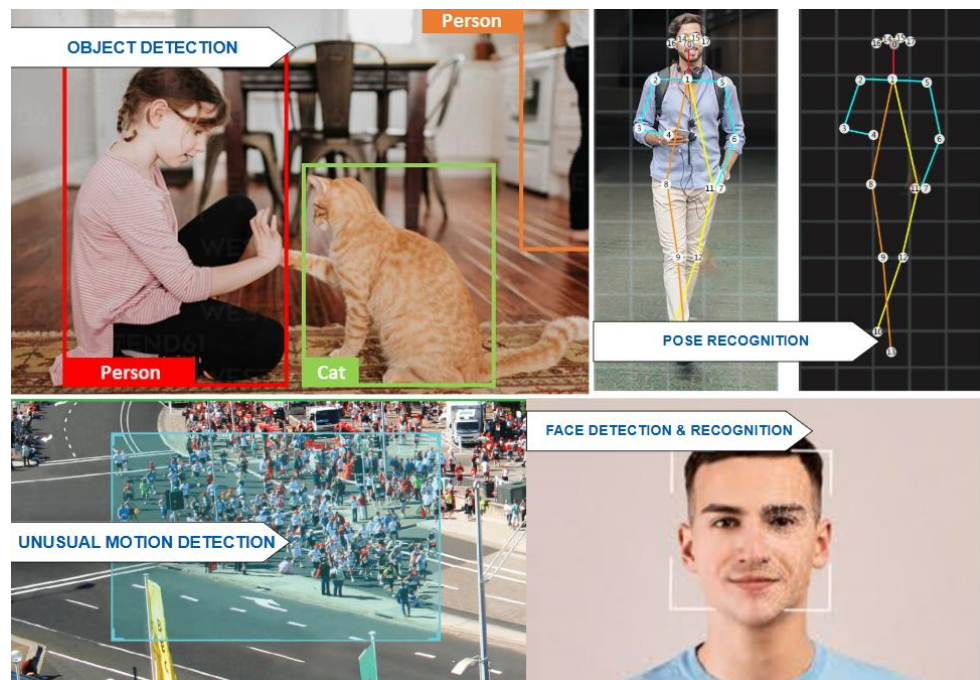


Рисунок 1.1 - Приклад атрибутів для відео

Окремо важливо виділити роботу з текстом та звуком. Визначення цих атрибутів може допомогти визначати наявність номерів, рекламних банерів, тексту на екрані, генерувати автоматичні субтитри та визначати звукові події [29, 30].

Кількість залучених атрибутів одночасно може залежати від задачі, але використання великої кількості моделей значно уповільнює процес визначення атрибутів для цілого відео, тому задача зменшення вимог до обчислювальних ресурсів стає нагальною та потребує покращень.

1.2. Архітектура моделей для роботи з відео

При аналізі відео вибір архітектури нейронної мережі стає дуже важливою задачею, так як, на відміну від інших задач, пов'язаних із комп'ютерним зором, необхідно враховувати послідовну залежність між кадрами у відео. В той же час використання інших підходів, оснований на математичних методах, є неефективним через їх низьку точність та швидкість при аналізі сучасного відеоконтенту через постійну тенденцію до ускладнення структури відеоконтенту.

На противагу цьому, нейронні мережі мають можливість враховувати просторові та часові залежності, що дозволяє ефективно аналізувати контекст кадрів та визначати залежності між ними [30]. Також важливо звернути увагу на останні розробки в сфері нейронних мереж для обробки візуальних даних, а саме візуальні трансформери для відео. Вони дозволяють не тільки використовувати механізм самоуваги, який значно покращує точність моделей, але й аналізувати декілька кадрів одночасно, враховуючи послідовну залежність [23].

1.2.1. Рекурентні нейронні мережі

Рекурентні нейронні мережі є одним із стандартних підходів для вирішення задач з обробки послідовних даних, таких як часові ряди, текст, аудіо чи відео. Особливістю рекурентних нейронних мереж є можливість зберігати внутрішній стан системи, тим самим запам'ятовуючи важливу інформацію, отриману із попередніх кадрів. Далі, при обробці, ця інформація використовується для аналізу та модифікується для подальших обчислень. Ця властивість робить рекурентні нейронні мережі корисними в задачах, де важливий контекст або залежності між елементами в ланцюгу обробки, такими як сигнали чи кадри [31, 32].

Архітектура рекурентних нейронних мереж базується на циклічних зв'язках між вузлами, інформація з яких проходить по вузлах послідовно. Також є модифікації, коли при аналізі інформація може проходити в обидва боки. Це дозволяє більш детально розглядати послідовності та отримувати значно більше інформації. Проте в цьому випадку нейронна мережа має декілька обмежень. Через проходження в обидва боки модель має отримувати відразу всю послідовність, щоб мати можливість запустити аналіз з кінця. Таким чином, модель втрачає можливість працювати в режимі реального часу, так як необхідно отримати достатню кількість елементів для запуску. Також ці моделі обмежені за довжиною, що означає, що після того, як модель відпрацює, треба запускати її заново без можливості перенести попередні стани, утворені при аналізі, на нову модель. На противагу цьому, звичайні рекурентні мережі можуть працювати з будь-якою довжиною, і у випадку надходження нових даних вони можуть продовжити з місця, де завершили попередній аналіз. Також при використанні двосторонніх рекурентних нейронних мереж треба враховувати, що кількість аналізу на таких шарах мережі подвоюється, що призводить до значного збільшення обчислень [33,34].

Завдяки можливості рекурентних нейронних мереж ефективно працювати з числовими рядами їх можна використовувати для аналізу відеоконтенту. вони можуть ефективно себе показувати в задачах визначення зміни планів та сцен. Завдяки зберіганню інформації про попередні стани вони можуть ефективно визначити силу зміни наповнення кадрів та навіть визначати складні переходи, такі як затухання чи повільні переходи між сценами [35].

При навчанні рекурентних нейронних мереж важливо звернути увагу на проблеми з згасаючим градієнтом, які можуть виникати при обробці довгих послідовностей, та можливості зберігати в пам'яті інформацію. З метою вирішення цих проблем були розроблені модифікації рекурентних нейронних

мереж — LSTM та GRU. Вони є різновидом рекурентних нейронних мереж, які дозволяють уникати згасаючого градієнту та зберігати довгострокові залежності завдяки використанню механізму керування інформацією, а саме через додавання спеціальних воріт, які дозволяють видаляти з пам'яті неактуальну інформацію [31, 36]. Це є важливим нововведенням, особливо для задач визначення зміни сцен та планів, так як зберігання інформації про старі сцени та кадри може вплинути на визначення нових переходів.

Під час практичного застосування рекурентних нейронних мереж важливо враховувати, що, не зважаючи на високу точність при обробці залежностей між даними, ці моделі мають високу складність обчислень і сильно залежать від довжини послідовності [37]. Також ця архітектура не має можливості враховувати просторові залежності, що може значно ускладнити процес визначення концепту, закладеного в кадри.

1.2.2. Згорткові нейронні мережі

Згорткові нейронні мережі широко використовуються для обробки візуальних даних завдяки їх здатності виділяти просторові залежності. Особливість цих нейронних мереж полягає у використанні операцій згортки з метою визначення локальних ознак об'єктів на зображенні. Операція згортки — це математична операція, яка базується на використанні фільтрів до вхідних даних. На перших рівнях нейронної мережі операції згортки зазвичай виділяють базові характеристики зображення, такі як краї об'єктів, текстури, контури, переходи кольорів та подібні характеристики. У подальших шарах операції згортки починають виділяти більш складні ознаки, які дозволяють виявляти складні закономірності на зображенні [38].

Особливістю згорткових нейронних мереж є їх можливість самостійно вчитися виділяти ознаки об'єктів, що, на відміну від математичних підходів, дозволяє відмовитися від ручного проектування дескрипторів. Завдяки

використанню великої кількості згорткових шарів модель поступово аналізує вхідні дані та поступово виділяє базові ознаки, а потім з них може виділяти більш складні та змістовні. Таким чином, модель будує ієрархічне представлення даних, що значно покращує точність класифікації. Також важливим фактором під час використання згорткових нейронних мереж є використання спільних фільтрів для згорткових шарів. Так як фільтри використовують спільні ваги, це дозволяє значно зменшити кількість параметрів у нейронній мережі, що, у свою чергу, дозволяє згортковим нейронним мережам мати менший ризик перенавчання [39].

Також важливо зауважити, що згорткові нейронні мережі мають гарну адаптивність до зміщення та масштабування об'єктів. Ця адаптивність досягається шляхом використання пулінгових шарів (*pooling*). Пулінгові шари зменшують кількість даних, при цьому зберігаючи ключову інформацію. Найбільш розповсюдженим варіантом пулінгу даних є вибір найбільшого значення в локальній області (*max pooling*). Це дозволяє робити мережу менш чутливою до невеликих змін у положенні об'єкта. Можливість адаптуватися до зміщення та зміни розміру є дуже важливою при аналізі відеоконтенту, так як це дозволяє краще аналізувати об'єкти, які переміщуються в кадрі [40].

Для покращення продуктивності були розроблені різні модифікації згорткових нейронних мереж, такі як ResNet, EfficientNet та Inception. При розробці були представлені нові механізми для обробки зображень, що дозволили значно підвищити точність та продуктивність моделей. Архітектура згорткової нейронної мережі ResNet представила механізм залишкових зв'язків, який дозволив передавати інформацію між шарами. Використання цього механізму дозволило запобігти проблемам затухаючого градієнту в глибоких нейронних мережах та дозволило розробляти значно глибші моделі для кращого визначення ознак на зображенні. Інші модифікації також були спрямовані на покращення точності та підвищення швидкодії згорткових нейронних мереж [22, 41].

Проте, незважаючи на свої переваги, згорткові нейронні мережі мають складності при роботі з часовими залежностями, що є важливим етапом при обробці відеоконтенту. Для вирішення цієї проблеми згорткові нейронні мережі часто комбінують з іншими типами нейронних мереж, які можуть ефективно працювати з часовими залежностями, такими як рекурентні нейронні мережі чи трансформери. Таким чином, можливості визначення просторових ознак можуть бути поєднані з визначенням часових залежностей для обробки відеоконтенту. Проте недоліком цього підходу є складність моделі та потреба в більших обчислювальних ресурсах [42, 43].

Також для вирішення проблеми з часовими залежностями було розроблено тривимірні згорткові нейронні мережі, які стали основою для обробки відеоконтенту. Використання цього підходу дозволило, на відміну від стандартної двовимірної згортки, оброблювати просторові та часові залежності завдяки застосуванню тривимірної згортки, яка одночасно може працювати з шириною, висотою та глибиною даних. Це дозволяє враховувати зміни які відбуваються між кадрами, чи при роботі з тривимірними даними. Цей підхід може активно застосовуватися при роботі з відеоданими, медичною візуалізацією та іншими сферами, де може бути потрібна робота з тривимірними даними [44, 45].

Важливою характеристикою тривимірних згорткових нейронних мереж є великий розмір ядер згортки, що дозволяє враховувати як локальні особливості, так і глобальні залежності між даними. Завдяки цій характеристиці тривимірні згорткові нейронні мережі можуть кодувати просторово-часові ознаки та на їх основі розпізнавати важливі зміни у часовій послідовності. Це дозволяє ефективно використовувати тривимірні згорткові нейронні мережі для задач моніторингу, розпізнавання дій чи визначення зміни планів та сцен [47].

Недоліком використання тривимірних згорткових нейронних мереж є висока обчислювальна складність та швидкість обробки. Ця проблема

зумовлена потребою збільшення розміру ядра для аналізу та збільшення його розмірності, що в свою чергу значно підвищує кількість параметрів. При збільшенні кількості параметрів виникає не лише проблема зі швидкістю обчислень, але й проблема з навчанням таких моделей. При збільшенні розміру моделі необхідно враховувати потребу в збільшенні кількості навчальних даних, щоб уникнути перенавчання моделі та забезпечити її можливість навчатися узагальнювати знання. Але кількість розмічених тривимірних даних є значно меншою та потребує значно більшої роботи для накопичення цих даних, та потенційної розмітки їх. Також такі моделі потребують спеціальних методів регуляризації з метою ефективного навчання, що впливає на подальшу складність оптимізації таких моделей.

З метою покращення швидкодії таких моделей активно використовують методи оптимізації. Але, крім класичних методів оптимізації, таких як прунінг, квантування чи дистиляція знань, для тривимірних згорткових нейронних мереж також застосовуються методи розрідженої згортки. Застосування методів розрідженої згортки дозволяє обробляти тільки найбільш важливі елементи та особливості, що дозволяє значно зменшити кількість необхідних обчислень. Також застосування просторово-розділених згорток дозволяє мінімізувати витрати на обчислення, при цьому зберігаючи точність розрахунків. Також було розроблено просторові нормалізації для більш ефективної роботи з тривимірними даними з метою вирівнювання розподілу даних [47].

Хоча практичне застосування тривимірної згортки охоплює багато сфер, найбільше вони застосовуються для розпізнавання об'єктів, аналізу поведінки, відстеження рухів та в медичній діагностиці, де застосування тривимірних згорткових нейронних мереж допомагає виявляти патології та аномалії. Також тривимірні згорткові нейронні мережі можуть застосовуватися в автономному водінні завдяки їх можливості розпізнавати об'єкти та перешкоди під час руху [48, 49]. У випадку з визначенням зміни сцен та планів використання

тривимірних згорткових нейронних мереж дозволяє ефективно знаходити зміни сцен та планів, основувшись на об'єктах, присутніх у кадрах, та визначаючи ключові концепти, закладені у відповідні сцени та плани.

1.2.3. Моделі на основі трансформерів

Архітектури на основі трансформерів початково були розроблені для обробки послідовностей, таких як текст, і стали головним інструментом у їх обробці. Ця архітектура змогла значно підвищити ефективність роботи з текстом завдяки введенню механізму самоуваги, який дозволяє моделі одночасно зосереджуватися на різних частинах вхідних даних. Цей механізм забезпечує можливість моделей ефективно виявляти зв'язки між елементами в послідовності, незважаючи на відстань між елементами [50].

Механізм самоуваги працює шляхом обчислення ваг для кожної пари елементів у моделі, таким чином модель може визначати важливість кожного елементу відносно інших елементів. Внаслідок цього, елемент може мати сильні зв'язки з іншими елементами, які можуть знаходитися в іншій частині послідовності. Завдяки цьому моделі архітектури трансформер можуть моделювати довгострокові залежності набагато краще, ніж рекурентні нейронні мережі, яким важко утримувати важливу інформацію для довгих послідовностей, таких як тривалі відео чи великі обсяги тексту [51].

Також при проходженні по моделі архітектури трансформер дані обробляються паралельно, що є великою перевагою порівняно з рекурентними нейронними мережами, які обробляють всі дані послідовно. Завдяки цьому такі моделі набагато швидше навчати, і потім вони набагато швидше працюють, ніж рекурентні нейронні мережі, особливо для моделей, які працюють з великими наборами даних. Також ця архітектура легко адаптується для роботи з різними задачами, що робить її підходящою для аналізу аудіо, зображень та відео [52].

Візуальні трансформери стали адаптацією класичних трансформерів з фокусом на аналіз зображень. Тоді як згорткові нейронні мережі аналізують зображення за допомогою локальних фільтрів, які виділяють ознаки, візуальні трансформери розбивають початкове зображення на невеликі частини – патчі. Далі ці патчі подаються як послідовність токенів, що подібно до стандартних трансформерів, де у ролі токенів були слова. Потім кожен патч кодується у вектор фіксованої довжини, і набір цих векторів подається на вхід моделі [53].

Завдяки такому підходу до аналізу початкових даних візуальні трансформери отримують здатність до моделювання глобальних залежностей між частинами зображення. В результаті, на відміну від згорткових нейронних мереж, модель враховує не тільки локальні ознаки, а й фокусується на зв'язках між віддаленими елементами на зображенні завдяки механізму самоуваги. Це дозволяє значно краще розуміти контекст зображення, що значно підвищує точність у таких задачах, як класифікація сцен, визначення об'єктів, опис зображення [54].

Працюючи з архітектурою візуальних трансформерів, важливо враховувати, що моделі на основі цієї архітектури потребують великих обчислювальних ресурсів та великої кількості даних для навчання. Це викликано квадратичною складністю відносно кількості вхідних токенів у механізмі самоуваги.

Для роботи з відео було розроблено модифікацію візуальних трансформерів, яка отримала назву “візуальні трансформери для відео”. Основною особливістю архітектури візуальних трансформерів для відео є можливість аналізувати одночасно просторові та часові залежності у відео, що дозволяє моделювати складні взаємодії між об'єктами, переміщення об'єктів між кадрами та визначати контекст сцен [23].

Аналогічно до візуальних трансформерів, де вхідне зображення розбивається на патчі, у візуальних трансформерах для відео вхідні дані розглядаються як тривимірний об'єкт. Враховуючи цю особливість, кожен

патч охоплює не лише просторові особливості, а й часові залежності. Це дозволяє моделі враховувати одночасно і просторові характеристики окремих кадрів, і часові залежності між ними.

Механізм уваги у візуальних трансформерах для відео застосовується як до просторових, так і до часових залежностей, що дозволяє моделі звертати увагу на зміну положення об'єктів, які рухаються протягом декількох кадрів, та як змінюється контекст сцен протягом часу. Ця особливість робить архітектуру візуальних трансформерів для відео особливо ефективною при вирішенні задач класифікації дій, прогнозування руху, виявлення аномалій та визначення зміни сцен, основуючись на контексті [55].

Також, на відміну від рекурентних нейронних мереж чи тривимірних згорткових нейронних мереж, моделі на основі архітектури візуального трансформеру для відео не обмежуються локальними залежностями в даних, а й застосовують глобальну увагу до всього відео. Це дозволяє моделі враховувати довгострокові зв'язки між подіями, які можуть відбуватися в різних частинах відео.

Проте, так як механізм самоуваги має квадратичну складність відносно кількості токенів, архітектура візуальних трансформерів для відео має високі вимоги до обчислювальних ресурсів, особливо для довгих відео. Для зменшення обчислювальної складності, крім класичних методів оптимізації моделей, також часто застосовують токенізацію за ключовими моментами, виділяючи найважливіші кадри для зменшення кількості обчислень, розділяють відео приблизним розмежуванням для поступового аналізу або розділяють механізм самоуваги на просторову та часову компоненти для зменшення складності обчислень [55].

Архітектура візуальних трансформерів для відео має гарну адаптивність та підходить для вирішення широкого спектру задач, включаючи детекцію об'єктів, сегментацію відео, прогнозування руху чи визначення зміни сцен та планів. Але важливо враховувати, що ця архітектура чутлива до шумів, таких

як рух камери чи зміни освітлення, через що відео часто потребують додаткової обробки перед аналізом.

1.2.4. Гібридні архітектури

Для вирішення задачі визначення зміни сцен та планів можливо застосовувати гібридні архітектури, які можуть поєднувати необхідні архітектури, досягаючи одночасної та ефективної обробки просторових та часових характеристик. Вибір гібридних архітектур базується на інтеграції сильних сторін різних підходів. Поєднання згорткових нейронних мереж разом з рекурентними нейронними мережами є розповсюдженим підходом для аналізу відео. В основі підходу лежать трансформації вхідних даних, а саме кадрів, за допомогою згорткових нейронних мереж. Під час аналізу кадрів згорткова нейронна мережа виділяє важливі просторові ознаки для кожного кадру у відео, такі як текстури, краї, важливі об'єкти та їх взаємодію. В результаті обробки згортковою нейронною мережею вхідне зображення кодується у вектор, який зберігає в собі важливу для аналізу інформацію. Наступним кроком є передача цих даних до рекурентної нейронної мережі, яка може аналізувати зміни в цих характеристиках відносно часової змінної. Завдяки цьому підходу стає можливим ефективно виділення просторових ознак, використовуючи згорткові нейронні мережі, при чому зберігаючи можливість для подальшого аналізу, використовуючи довгострокові залежності між кадрами. Недоліком цього підходу є складність навчання таких моделей через проблему затухання градієнтів при навчанні довгих послідовностей, а також високі обчислювальні витрати та низька швидкість таких моделей [56].

Також можлива заміна рекурентних нейронних мереж на моделі архітектури трансформер. У цьому випадку стає можливим застосування механізму самоуваги для аналізу глобальних просторових та часових

залежностей. При використанні такого підходу зникає проблема затухаючого градієнту, викликана застосуванням рекурентних нейронних мереж, та покращується точність завдяки здатності аналізувати довгострокові залежності. Проте така модель може мати високі обчислювальні вимоги через квадратичну складність розрахунку механізму самоуваги, особливо на великих наборах даних. Також при використанні архітектури на основі трансформерів потрібна значно більша кількість даних для навчання [57].

Для зменшення кількості обрахунків можливе використання тривимірних згорткових нейронних мереж. У цьому випадку тривимірні згорткові нейронні мережі можуть виділяти одночасно просторові та часові ознаки для певного відрізка відео. В результаті такого аналізу стає можливим значно зменшити кількість вхідних даних до моделі трансформеру. При цьому завдяки використанню моделі трансформеру зберігається можливість до глобального аналізу залежностей у відео [58].

Також можливий гібрид згорткових нейронних мереж, рекурентних згорткових мереж та моделей трансформеру. У цьому випадку кожен кадр спочатку подається до згорткової нейронної мережі для виділення просторових ознак. Далі виділені ознаки передаються до рекурентної нейронної мережі для моделювання часових залежностей. Згенеровані ознаки за допомогою рекурентних нейронних мереж передаються в модель на основі трансформеру з метою додаткового аналізу, використовуючи глобальні залежності. Такий підхід має високу точність завдяки комплексному аналізу просторових, часових та глобальних залежностей. Однак така архітектура потребує дуже високих обчислювальних ресурсів, має низьку швидкодію та потребує великих наборів даних для навчання [59].

1.3. Сучасні підходи до оптимізації моделей для обробки відео

Сучасні нейронні мережі активно застосовуються для аналізу відео та мають високу точність завдяки здатності аналізувати просторові та часові залежності, а також виявляти ключові закономірності на великих масивах даних. Проте застосування таких моделей потребує великої кількості обчислювальних ресурсів, що значно обмежує використання таких моделей для практичного застосування, де важлива висока швидкість виконання, ресурси обмежені, а низьке енергоспоживання є критично важливим.

Через ці обмеження розробники та дослідники приділяють особливу увагу оптимізації моделей. Метою оптимізації моделей є зменшення обчислювальної складності, скорочення розміру моделей та зменшення енергоспоживання, при цьому зберігаючи оригінальну точність моделі або маючи мінімальні втрати по точності. Один з найбільш поширених способів оптимізації моделей — це прунінг. Цей підхід полягає у визначенні нейронів та зв'язків моделі, які мають найменший вплив на отримання кінцевого результату, та подальшому їх видаленні. Застосування прунінгу дозволяє значно зменшити розмір моделі та пришвидшити швидкість обробки [60].

Квантування є іншим методом оптимізації, який полягає в зменшенні числового представлення параметрів моделі. Під час квантування параметри переводяться з 32-бітних чисел до 16-бітних чи навіть 8-бітних представлень. Завдяки цьому значно зменшується необхідний обсяг пам'яті для використання моделі та зменшуються обчислювальні витрати на арифметичні операції, що є важливим для пристроїв з низькою обчислювальною потужністю [61].

Ще одним методом оптимізації нейронних мереж є застосування дистиляції знань. При застосуванні цього підходу розробники мають можливість спочатку навчити велику модель для досягнення максимальної точності, незважаючи на обчислювальні витрати. Далі розробляється менша

модель, яка буде відповідати вимогам щодо обчислювальних ресурсів та розміру, і навчається вона, використовуючи не тільки анотовані дані з датасету, але й результати обчислень оригінальної моделі. Завдяки цьому мала модель може ефективно зберігати ключову інформацію про дані та поведінку оригінальної моделі при значному зменшенні її розміру та обчислювальної складності [62].

При використанні хмарних технологій відкривається додаткова можливість ефективно використовувати паралельні розрахунки з метою значного пришвидшення отримання результатів.

1.3.1. Використання прунінгу для оптимізації моделей

Однією з найбільш ефективних стратегій зменшення обчислювальної складності моделей вважається застосування прунінгу, так як цей підхід дозволяє зберігати високий рівень продуктивності при значному зниженні кількості параметрів моделі, що значно підвищує швидкодію. Механізм прунінгу полягає у видаленні нейронів чи зв'язків між ними, які мають найменший вплив на точність моделі, таким чином зменшуючи їх і створюючи більш компактну та швидку архітектуру нейронної мережі [63].

Зазвичай методи прунінгу поділяють на дві категорії: неструктурований та структурований прунінг. Неструктурований прунінг видаляє окремі ваги в мережі, що значно зменшує розмір моделі, але в процесі створює розріджені матриці. Хоча це і приводить до зменшення обчислювальної складності, це не завжди може забезпечити апаратне прискорення. Зазвичай цей метод добре показує себе, коли кількість параметрів для прунінгу має дуже великий відсоток відносно загальної кількості параметрів моделі, або при розрахунку результатів нейронної мережі за допомогою процесора замість відеокарти. Структурований прунінг, на відміну від неструктурованого, видаляє не окремі зв'язки, а повноцінні структури, такі як фільтри, канали або нейрони. Завдяки

цьому архітектура після структурованого прунінгу залишається більш регулярною та може легко масштабуватися при застосуванні її на сучасних апаратних засобах [64, 65].

При застосуванні методу прунінгу важливо поєднувати його з іншими методами оптимізації, такими як додаткове навчання після прунінгу. Це є важливим кроком після застосування прунінгу, так як дозволяє моделі навчатися додатково, але вже використовуючи нову архітектуру. Завдяки цьому модель після прунінгу може додатково збільшити свою точність, після застосування донавчання, тим самим мінімізуючи зменшення точності, викликане видаленням частини архітектури. Також існують адаптивні підходи, які дозволяють змінювати рівень розрідженості матриць залежно від вхідних даних та задачі, яку виконує модель. Це підтверджується алгоритмами на основі "Lottery Ticket Hypothesis", які довели, що в великих та складних моделях існують менші моделі, які в цих дослідженнях називаються виграшними білетами, які можна тренувати окремо, при цьому зберігаючи ефективність початкової моделі [67, 68].

Ще одним типом прунінгу, який набирає популярність, є прогнозувальний прунінг, який виконується на етапі ініціалізації нейронної мережі. Цей підхід дозволяє уникнути попереднього навчання повної моделі, що значно знижує витрати часу та ресурсів ще на етапі навчання нейронної мережі.

1.3.2. Квантування моделей

Квантування нейронних мереж є простою, але ефективною технологією оптимізації, яка дозволяє значно зменшувати обчислювальні витрати та потребу в пам'яті. При цьому втрата точності при застосуванні квантування є мінімальною. Так як робота з відео потребує великих обчислювальних ресурсів, застосування квантування може бути надзвичайно важливим. Також

застосування цієї технології може дозволити перенести розрахунки на мобільні пристрої та пристрої з обмеженими ресурсами, а також зменшити енергоспоживання при використанні таких моделей. Ідея квантування полягає в зменшенні розрядності даних, тобто переході від 32-бітних чисел з плаваючою комою до чисел з меншою розрядністю, таких як 16-бітні числа з рухомою комою або 8-бітні цілі числа [68, 69].

Залежно від задач та доступних ресурсів можуть використовуватись різні методи квантування. Перший підхід полягає в квантуванні після навчання моделі. Цей підхід є дуже простим та швидким, але недоліком цього підходу може стати втрата точності, особливо якщо переводити параметри моделі до дуже низької розрядності. Для вирішення цієї проблеми застосовують методи коригування помилок, такі як адаптивне округлення даних. Іншим підходом є квантування з урахуванням навчання, під час цього підходу процес квантування є вбудованим у процес навчання моделі. Це дозволяє нейронній мережі адаптуватися до зміни розрядності та зберігати точність при скороченні розрядності параметрів нейронної мережі [70, 71].

При застосуванні квантування розрізняють однорідні та неоднорідні схеми. Однорідна схема квантування має фіксований крок між рівнями, набагато легша в реалізації, особливо на рівні апаратного забезпечення, та широко використовується для великих моделей. Неоднорідне квантування є більш складним через забезпечення відповідності параметрів моделі, що призводить до меншої втрати точності, що може бути критичним для задач, які потребують високої точності [72].

До переваг застосування квантування можна віднести значне зменшення енергоспоживання, зменшення витрат на апаратні ресурси та збільшення швидкодії обробки. Це дозволяє ефективно використовувати нейронні мережі на пристроях з обмеженими ресурсами чи аналізувати дані в режимі реального часу, що може бути важливо в задачах, пов'язаних з обробкою відеоданих. Але при застосуванні квантування важливо враховувати, що точність моделі може

зменшуватись, особливо при переході до низької розрядності параметрів нейронної мережі. Також викликом може стати адаптація квантизованої нейронної мережі до певних апаратних платформ, так як ці платформи можуть не мати оптимізованих операцій для роботи зі значеннями, які мають нестандартну розрядність, що в свою чергу може не пришвидшити час виконання моделі, а в деяких випадках навіть сповільнити її.

1.3.3. Інтеграція дистиляції знань для підвищення швидкодії

Процес дистиляції знань полягає в передачі знань від великих та складних моделей, які називають вчителями, до менших моделей, які називаються студентами, забезпечуючи збереження точності моделі при значному зменшенні обчислювальних витрат. Дистиляція знань часто застосовується для зменшення розміру моделей та їх прискорення [73].

Однією з головних особливостей дистиляції знань є можливість перенесення важливої концептуальної інформації, яка отримується завдяки навчанню великих нейронних мереж. Це відбувається завдяки інтеграції в навчання не лише розмічених даних з датасету, але й з використанням результатів, які повертає модель-вчитель. Таким чином модель-студент отримує можливість більш точно переймати знання з моделі-вчителя. Так як відеодані включають в себе комплексну інформацію, яка має одночасно просторові та часові залежності, дистиляція знань допомагає забезпечити адаптацію менших моделей-студентів до роботи з складними залежностями [74, 75].

Додаткові дослідження показують, що застосування методу дистиляції знань може використовуватися для покращення графових нейронних мереж та мереж на основі архітектури трансформерів, які активно застосовуються в аналізі відеоданих. Такі підходи, як використання м'якої розмітки чи стратегія

багаторівневої дистиляції, дозволяють збалансувати точність та продуктивність моделі [76, 77].

1.4. Аналіз вимог до програмного забезпечення та постановка наукового завдання

Проведений аналіз методів визначення відеоатрибутів показав, що найбільшу ефективність при аналізі відеоконтенту демонструють підходи, засновані на використанні глибоких нейронних мереж. Особливу увагу заслуговують архітектури на основі трансформерів, такі як візуальні трансформери для відео. Традиційні алгоритми, які базуються на використанні математичних підходів чи згорткових нейронних мереж, показують обмежену здатність до ефективного аналізу просторових та часових залежностей у відеоконтенті, що, в свою чергу, призводить до низької точності при визначенні зміни планів та сцен, особливо у випадках сучасного відеоконтенту, який може містити в собі складні переходи, коли сцени змінюються поступово, чи при необхідності розпізнавати складні концепти у взаємодії між об'єктами.

Також важливим викликом у сфері аналізу відеоконтенту є підвищення швидкодії та ефективності обробки відеоконтенту. В той час як використання глибоких нейронних мереж для аналізу відеоконтенту має високу точність, потреба в обчислювальних ресурсах сильно зростає, що значно обмежує застосування таких моделей для задач реального світу. Для вирішення проблеми з потребою у високих обчислювальних ресурсах необхідно застосовувати техніки оптимізації, такі як прунінг, квантування чи дистиляція знань, що може дозволити зменшити розмір моделі та вимоги моделі до обчислювальних ресурсів без значної втрати точності.

Постає задача розробки методів та програмних засобів, що дозволяють точно та швидко визначати зміни планів та сцен, та на основі цього розробити

програмне забезпечення, яке дозволить оптимізувати процес визначення відеоатрибутів, зберігаючи точність розпізнавання. Основна наукова проблема полягає у поєднанні підходів до сегментації сцен за допомогою сучасних методів аналізу відео за допомогою візуальних трансформерів для відео з метою отримання високої точності та мінімізації обчислювальних витрат. Виходячи з поставленої наукової проблеми, сформульовано функціональні вимоги (ФВ), які представлено у вигляді таблиці 1.1.

Таблиця 1.1 Функціональні вимоги до системи розподіленого визначення відеоатрибутів

Код вимоги	Вимога
ФВ1	Забезпечення можливості визначення планів на відео за допомогою поєднання математичних підходів з рекурентними нейронними мережами в режимі реального часу
ФВ2	Забезпечення можливості швидкого визначення сцен на відео за допомогою візуальних трансформерів для відео
ФВ3	Забезпечення зв'язку між модулями за допомогою веб-запитів
ФВ4	Забезпечення розгортання модулів як серверів
ФВ5	Забезпечення можливості додавання нових модулів для розпізнавання відеоатрибутів
ФВ6	Забезпечення автоматичної адаптації системи до навантаження

Нефункціональні вимоги (НФВ), які рекомендовано врахувати при розробці ПЗ даного типу, наведено в таблиці 1.2

Таблиця 1.2 Нефункціональні вимоги до системи розподіленого визначення відеоатрибутів

Код вимоги	Вимога та її опис
НФВ1	Надійність: програмне забезпечення (ПЗ) повинно функціонувати безвідмовно, забезпечуючи консистентність аналізу даних
НФВ2	Сумісність та інтеграція: при розробці ПЗ потрібно передбачити сумісність з іншими системами та інструментами для визначення відеоатрибутів
НФВ3	Продуктивність: ПЗ повинно швидко оброблювати дані та ефективно передавати зображення між модулями

Метою дисертаційного дослідження є підвищення точності та швидкодії розбиття відео на сцени шляхом розробки моделей з використанням візуальних трансформерів для відео та розробка спеціальних програмних засобів для зниження обчислювальних витрат при визначенні атрибутів.

Об'єктом досліджень є процес обробки відеоданих з використанням розбиття відео на сцени за допомогою візуальних трансформерів для відео та оптимізація програмних засобів при визначенні атрибутів у відео.

Предметом дослідження є удосконалення моделей розбиття відео на сцени за допомогою візуальних трансформерів для відео та підвищення швидкодії програмних засобів з метою зниження обчислювальних витрат при аналізі відеоданих.

Для досягнення мети в дисертації вирішено такі наукові завдання:

1. Виконати аналіз існуючих методів розбиття відео на плани.
2. Виконати аналіз існуючих методів розбиття відео на сцени.
3. Дослідити швидкодію існуючих методів.
4. Дослідити ефективність існуючих методів при застосуванні їх для сучасного відеоконтенту.

5. Розробити новий алгоритм розбиття відео на плани за допомогою нейронних мереж.
6. Розробити новий алгоритм розбиття відео на сцени, використовуючи плани та візуальні трансформери для відео.
7. Розробити розподілену архітектуру системи для швидкого визначення атрибутів у відео за допомогою розбиття на плани та сцени на основі нейронних мереж.

1.5. Висновки до розділу 1

Розділ містить аналіз існуючих підходів до визначення відеоатрибутів, що є важливою складовою автоматизованого аналізу відеоконтенту. Було розглянуто традиційні методи визначення зміни сцен та планів у відеоконтенті, які засновані на використанні математичних алгоритмів, а також було розглянуто використання глибоких нейронних мереж для вирішення цієї задачі. Особливої уваги було приділено використанню архітектури візуальних трансформерів для відео, яка демонструє високу точність при аналізі відеоконтенту завдяки здатності ефективно аналізувати просторові та часові залежності у відеопослідовностях.

Було розглянуто методи оптимізації нейронних мереж, такі як прунінг, квантування та дистиляція знань. Аналіз цих методів продемонстрував їх ефективність у зменшенні обчислювальної складності моделей та пришвидшенні їх виконання, що дозволяє ефективно застосовувати їх для задач реального світу, які потребують високої швидкодії при використанні на пристроях з обмеженими обчислювальними ресурсами. Особливої уваги заслуговує метод прунінгу, який дозволяє оптимізувати моделі ще на етапі навчання, що забезпечує додаткову оптимізацію процесів навчання, що може бути критичним при розробці глибоких нейронних мереж.

Аналіз існуючих у відкритому доступі програмних засобів і підходів до визначення зміни планів показав, що ці методи мають недостатньо точність при роботі з сучасним відеоконтентом, що призводить до потреби розробки нових методів для визначення зміни планів. Існуючі програмні засоби і методи визначення зміни сцен показав, що ці методи не мають можливості ефективно використовувати контекст сцен при аналізі, що призводить до потреби розробки нових програмних засобів і методів визначення зміни сцен з врахуванням контексту відеоконтенту. Аналіз методів прунінгу нейронних мереж показав відсутність підходів для прунінгу перед навчанням для моделей архітектури трансформерів для відео. Цей факт дозволяє визначити напрямок подальшого дослідження та вимоги до архітектури програмної системи.

РОЗДІЛ 2. ВДОСКОНАЛЕННЯ МЕТОДІВ СЦЕННОГО РОЗБИТТЯ ТА АНАЛІЗУ ВІДЕОАТРИБУТІВ

2.1. Алгоритми розбиття відео на сцени

У сучасному світі обсяг відеоданих активно зростає, що призводить до потреби вдосконалювати існуючі методи роботи з відеоданими з метою покращення точності та швидкості аналізу. Для оптимізації процесу аналізу відеоконтенту важливою стає розробка методів визначення переходів сцен та планів у відео, оскільки це забезпечує структурованість контенту та полегшує подальші аналітичні завдання завдяки зменшенню необхідної кількості кадрів для аналізу без втрати оригінальної точності.

У результаті застосування підходу розбиття відео на сцени та плани вирішення таких задач, як пошук за змістом, підсумовування відео, визначення ключових моментів у відео, стає набагато швидшим і потребує значно менших обчислювальних потужностей. Проте класичні підходи до вирішення задач розбиття відео на сцени та плани, які базуються на побудові простих математичних моделей чи аналізі гістограм, показують низьку точність аналізу, особливо в умовах сучасного відеоконтенту [78].

Наразі є пряма тенденція до ускладнення наповнення відеоконтенту, що призводить до скорочення довжини сцен та планів, а також до застосування більш складних переходів між планами.

Використання глибоких нейронних мереж для задачі визначення переходів між сценами та планами дозволяє значно покращити точність роботи моделей. Це зумовлено можливістю таких моделей ефективно враховувати просторові та часові залежності у відео, що призводить до більш якісного визначення контексту відео.

Сучасні моделі глибоких нейронних мереж, такі як візуальні трансформери для відео, добре підходять для вирішення подібних задач, оскільки вони можуть не тільки визначати просторові та часові

характеристики відео, але й ефективно аналізувати контекст всього відео за допомогою механізму самоуваги, що є надзвичайно важливим аспектом при виявленні зміни сцен.

2.1.1. Основні підходи до розбиття на сцени

Класичні підходи до визначення зміни сцен у відео зазвичай ґрунтуються на аналізі змін між послідовними кадрами. Здебільшого в них використовується аналіз гістограм, що дозволяє виявляти різкі зміни в послідовності кольорів чи яскравості. Метод порівняння гістограм дозволяє ефективно визначати межі сцен за умови сильних відмінностей між сусідніми кадрами [79].

Проте в сучасному відеоконтенті, де сцени можуть мати високу динаміку, різкі зміни кольорів у межах однієї сцени та складні переходи, цей метод починає показувати низьку точність.

Іншим підходом до вирішення задачі визначення меж сцен та планів є аналіз країв на зображенні. Цей метод виявляє крайові ознаки на послідовних кадрах та порівнює їх. У разі різкої зміни кількості чи розташування країв цей метод сигналізує про можливий перехід між сценами.

При застосуванні такого методу часто використовуються градієнтні методи та аналіз контурів, що дозволяє отримати високу швидкість аналізу. Хоча цей метод не такий чутливий до різких змін кольорів, він все ще погано справляється з високою динамікою у сценах та зі складними переходами між планами та сценами.

Іншим підходом до вирішення проблеми визначення зміни сцен та планів є аналіз часових кореляцій. Цей підхід аналізує кореляцію між пікселями на сусідніх кадрах для виявлення змін у структурі зображення. На основі змін часової кореляції визначаються переходи між сценами та планами.

Цей метод добре підходить для аналізу відеоконтенту, який має повільні переходи між сценами та планами [80].

Також існує підхід, який базується на визначенні ключових кадрів. Хоча в цьому підході не визначаються чіткі межі сцен та планів, цей метод дозволяє ефективно визначати ключові кадри всередині кожного плану. Ключові кадри у цьому підході визначаються шляхом аналізу зміни текстури, кольору або руху між кадрами, що дозволяє спростити подальший аналіз відеоконтенту [81].

Проаналізувавши ці підходи, можна зробити висновок, що вони можуть ефективно визначати зміни планів у простих відео. Проте при роботі з сучасним відеоконтентом ці методи стикаються з проблемами аналізу складних переходів та сцен, які містять багато динамічних елементів. Також недоліками таких підходів є нездатність визначати контекст сцен, що унеможливорює розрізнення переходів між сценами та планами.

Використання глибоких нейронних мереж для вирішення задачі визначення зміни планів та сцен відкриває нові можливості для аналізу. Це зумовлено можливістю глибоких нейронних мереж ефективно визначати просторові та часові характеристики у відео. Також із розвитком архітектур на основі трансформерів відкривається можливість ефективно аналізувати ціле відео одночасно завдяки механізму самоуваги.

При застосуванні глибоких нейронних мереж важливо враховувати їхні можливості та обмеження. Для задачі визначення переходів між сценами важливо, щоб модель мала змогу розуміти контекст як окремих кадрів, так і контекст відео в цілому. Задачу визначення контексту на зображенні добре вирішують згорткові нейронні мережі та візуальні трансформери.

За рахунок використання операцій згортки та пулінгу згорткові нейронні мережі можуть ефективно виділяти особливості на зображенні, починаючи з базових елементів, таких як межі об'єктів та переходи між кольорами, а потім виокремлюючи важливі для аналізу елементи. Візуальні трансформери, у свою

чергу, розбивають зображення на невеликі блоки, кодують їх та потім, на основі механізму самоуваги, шукають важливі зв'язки між цими блоками [82].

Цей підхід дозволяє аналізувати контекст усього зображення одночасно, що призводить до більш якісного визначення контексту зображення загалом. Проте недоліком цих підходів є нездатність визначати часові залежності.

Для визначення часових залежностей часто використовують рекурентні нейронні мережі або звичайні трансформери. Хоча рекурентні нейронні мережі можуть ефективно визначати часові залежності, їхнім недоліком є складність навчання для тривалого контенту через проблеми із затухаючим градієнтом.

Також для рекурентних нейронних мереж важко зберігати контекст протягом тривалого проміжку часу, що часто може призводити до його перезапису останніми даними. Проте можливим є поєднання підходів згорткових та рекурентних нейронних мереж. У цьому випадку зображення спочатку кодуються за допомогою згорткових нейронних мереж, що дозволяє виділити просторові ознаки на кадрах, а потім ці закодовані значення передаються до рекурентної нейронної мережі для визначення часових залежностей [83].

Більш сучасним підходом є заміна рекурентних нейронних мереж моделями архітектури трансформерів. На відміну від рекурентних нейронних мереж, ця архітектура немає проблем із затухаючим градієнтом та, крім того, може ефективно аналізувати часовий контекст відеоконтенту за допомогою механізму самоуваги [84].

Проте недоліком цих підходів є високі обчислювальні витрати, що зумовлено великими розмірами таких моделей. Також це, своєю чергою, призводить до потреби у більшій кількості навчальних даних під час навчання.

Також поєднання просторових та часових характеристик можливе при використанні тривимірних згорткових нейронних мереж та візуальних трансформерів для відео. Тривимірні згорткові нейронні мережі

використовують тривимірні ядра для шарів згортки, що дозволяє одночасно визначати просторові та часові залежності між кадрами. Це дає змогу відстежувати зміни у відеоконтенті та визначати на основі цього зміни сцен та планів.

Проте недоліком цього підходу є обмежений розмір ядра. Через це тривимірні згорткові нейронні мережі не можуть отримувати інформацію з кадрів, які розташовані далеко один від одного. Це призводить до того, що хоча ці моделі можуть ефективно визначати зміни планів, вони не мають достатньо контексту для ефективного визначення змін сцени.

Натомість візуальні трансформери для відео можуть, завдяки механізму самоуваги, ефективно визначати переходи між сценами, порівнюючи фрагменти відео, які можуть бути розташовані далеко один від одного.

2.1.2. Використання глибоких нейронних мереж

Визначення зміни сцен та планів є складною задачею, що потребує вирішення ряду технічних та практичних викликів. Основна складність полягає у складності сучасного відеоконтенту, оскільки він включає різкі зміни освітлення, швидкі рухи камери, об'єкти, які швидко рухаються, та складні композиції сцен. Ця складність робить виявлення зміни сцени дуже важким для методів, які не можуть ефективно аналізувати часові залежності впродовж усього відео і базуються на виявленні змін між сусідніми кадрами.

Також використання глибоких нейронних мереж потребує значних обчислювальних ресурсів, особливо в разі архітектур на основі трансформерів, оскільки їхня складність значно зростає при аналізі тривалих відео чи відео з високою роздільною здатністю. Необхідність у високих обчислювальних ресурсах значно обмежує використання цих моделей у задачах реального світу [85].

Окремою проблемою є обмеженість даних для навчання. Процес анотування відео є вкрай складним та часозатратним, що значно обмежує доступ до великих, різноманітних та добре розмічених наборів даних. Також є проблема суб'єктивності визначення сцен, що може призводити до неточностей ще на етапі навчання.

Цю проблему можна чітко побачити при аналізі датасету BBC Planet Earth, який пропонує п'ять різних розміток для визначення зміни сцен. У результаті моделі, які можуть показувати високу ефективність на певних наборах даних, можуть стикатися із труднощами узагальнення при роботі з новими типами відеоконтенту з інших доменів або відео, створених в іншому стилі. Через це культурні та художні відмінності у стилях відеоконтенту можуть значно знижувати точність визначення сцен [86, 87].

Можливими напрямками розвитку є покращення існуючих датасетів або розробка нових підходів до аналізу, які дозволять ефективніше використовувати наявні дані. Також можливий розвиток мультимодальних підходів, які прийматимуть на вхід не лише фрейми відео, а й аудіоскладову та субтитри.

У цьому випадку модель зможе отримувати додаткову важливу інформацію, що міститься в аудіоскладовій відео. Також важливою є оптимізація розроблених моделей з метою зменшення необхідних обчислювальних ресурсів та підвищення їхньої швидкодії.

Таким чином, застосування методів прунінгу, квантування або дистиляції знань дозволить оптимізувати моделі для ефективного аналізу великої кількості відеоконтенту.

2.1.3. Перспективні напрямки розвитку

Визначення зміни сцен та планів є складною задачею, що потребує вирішення ряду технічних та практичних викликів. Основна складність

полягає у складності сучасного відеоконтенту, оскільки він включає різкі зміни освітлення, швидкі рухи камери, об'єкти, які швидко рухаються, та складні композиції сцен. Ця складність робить виявлення зміни сцени дуже важким для методів, які не можуть ефективно аналізувати часові залежності впродовж усього відео і базуються на виявленні змін між сусідніми кадрами.

Також використання глибоких нейронних мереж потребує значних обчислювальних ресурсів, особливо в разі архітектур на основі трансформерів, оскільки їхня складність значно зростає при аналізі тривалих відео чи відео з високою роздільною здатністю. Необхідність у високих обчислювальних ресурсах значно обмежує використання цих моделей у задачах реального світу [85].

Окремою проблемою є обмеженість даних для навчання. Процес анотування відео є вкрай складним та часозатратним, що значно обмежує доступ до великих, різноманітних та добре розмічених наборів даних. Також є проблема суб'єктивності визначення сцен, що може призводити до неточностей ще на етапі навчання.

Цю проблему можна чітко побачити при аналізі датасету BBC Planet Earth, який пропонує п'ять різних розміток для визначення зміни сцен. У результаті моделі, які можуть показувати високу ефективність на певних наборах даних, можуть стикатися із труднощами узагальнення на нові типи відеоконтенту з інших для моделі доменів або відео, створених в іншому стилі. Через це культурні та художні відмінності у стилях відеоконтенту можуть значно знижувати точність визначення сцен [86, 87].

Можливими напрямками розвитку є покращення існуючих датасетів або розробка нових підходів до аналізу, які дозволять ефективніше використовувати наявні дані. Також можливий розвиток мультимодальних підходів, які прийматимуть на вхід не лише фрейми відео, а й аудіоскладову та субтитри.

У цьому випадку модель зможе отримувати додаткову важливу інформацію, що міститься в аудіоскладовій відео. Також важливою є оптимізація розроблених моделей з метою зменшення необхідних обчислювальних ресурсів та підвищення їхньої швидкодії.

Таким чином, застосування методів прунінгу, квантизації або дистиляції знань дозволить оптимізувати моделі для ефективного аналізу великої кількості відеоконтенту.

2.1.4. Критерії вибору архітектури

Вибір архітектури для вирішення задачі визначення зміни сцен є складним процесом, який включає аналіз вимог до технічних та ресурсних обмежень системи. Також необхідно враховувати здатність моделі до аналізу, зокрема визначення просторових та часових залежностей.

Спочатку слід проаналізувати вихідні дані, які потрібно аналізувати. При аналізі відеоконтенту важливо враховувати як технічні фактори, зокрема тривалість відеоконтенту, частоту кадрів та дозвіл відео, так і концептуальні особливості, такі як складність сцен та переходів між ними.

Доцільним підходом до аналізу відео є уніфікація відеоданих шляхом приведення всіх відео до єдиної частоти кадрів та уніфікованого розміру зображення. Це може значно зменшити потребу в навчанні моделі на відео з різною частотою кадрів. Проте, якщо необхідно аналізувати дані в режимі реального часу, можна застосувати інший підхід, а саме створення варіацій одного й того самого відео з різною частотою кадрів під час тренування.

Різні дозволи відео також можуть бути уніфіковані до спільного розміру як на етапі навчання, так і під час аналізу відеоданих у реальних застосуваннях.

На основі цього аналізу можна зробити висновок, що технічні параметри відео не впливають на вибір архітектури, проте вимагають уваги при

підготовці даних до тренування. Крім того, на етапі використання моделі, відео також слід трансформувати до уніфікованого формату.

Проте, при аналізі концептуальних особливостей відео стає зрозуміло, що підходи, засновані на математичних моделях, згорткові нейронні мережі, рекурентні нейронні мережі та мережі на основі трансформерів, не підходять для вирішення цієї задачі через нездатність одночасно охоплювати просторові та часові особливості у відео.

Проте комбінація цих підходів може дозволити визначати зміни сцен завдяки поєднанню визначення просторових особливостей за допомогою згорткових нейронних мереж та часових залежностей за допомогою рекурентних нейронних мереж чи моделей архітектури трансформерів. Також для вирішення таких задач можна використовувати тривимірні згорткові мережі та візуальні трансформери для відео, оскільки вони здатні одночасно визначати просторові та часові залежності.

Ще одним важливим аспектом є інтеграція такої системи у більш масштабну систему, яка на основі визначення зміни сцен та планів буде виконувати додаткові задачі, такі як визначення атрибутів у відео. У цьому випадку необхідно враховувати, щоб система була достатньо швидкою, і процес визначення сцен не сповільнював роботу до моменту, коли розбиття на сцени стало б недоречним або малоефективним.

Зважаючи на цю особливість задачі, архітектури, що містять рекурентні нейронні мережі, погано підходять для її вирішення через низьку швидкість виконання у порівнянні з іншими глибокими нейронними мережами. При навчанні моделі також важливо враховувати обсяг та якість доступних навчальних даних. Зі збільшенням розміру моделі вона починає потребувати більшої кількості даних для навчання, щоб ефективно навчитися узагальнювати знання.

Гібридні архітектури, які поєднують згорткові нейронні мережі та моделі на основі трансформерів, потребують значно більшої кількості даних

для навчання, що потенційно робить їх неефективними через складність ефективного узагальнення знань. Одним із найважливіших параметрів є рівень точності, якого може досягти архітектура нейронної мережі. Найперспективнішими архітектурами в цьому випадку є тривимірні згорткові нейронні мережі та візуальні трансформери для відео.

Хоча тривимірні згорткові нейронні мережі потребують меншої кількості даних для аналізу та можуть одночасно визначати як просторові, так і часові залежності, вони мають суттєвий недолік — вони не здатні аналізувати загальний контекст відео. Це зумовлено обмеженим розміром ядра згортки, яке дозволяє порівнювати лише частини сусідніх фреймів, що робить порівняння інформації, розташованої далеко одна від одної у відеоконтенті, неможливим.

Візуальні трансформери для відео, на відміну від тривимірних згорткових нейронних мереж, використовують механізм самоуваги, який дозволяє аналізувати різні частини відеоконтенту одночасно.

Ще одним важливим фактором є обмеженість обчислювальних ресурсів. Хоча візуальні трансформери для відео не є найефективнішою архітектурою для аналізу відеоконтенту, вони відповідають усім іншим вимогам, що зумовлює необхідність пошуку нових шляхів оптимізації таких моделей. Перспективним напрямком їхньої оптимізації є розвиток методів прунінгу перед навчанням для візуальних трансформерів для відео.

На основі цього аналізу найбільш підходящою архітектурою для вирішення задачі розбиття відео на сцени є візуальні трансформери для відео, оскільки вони забезпечують високу точність, враховують особливості даних та відповідають вимогам щодо ефективної обробки відеоконтенту.

2.2. Архітектура візуальних трансформерів для відео при визначенні зміни сцен

З розвитком складності сучасного відеоконтенту задача визначення зміни сцен стає щораз більш необхідною та важливою. Класичні підходи до вирішення цієї задачі показують низьку точність та ефективність через нездатність визначати контекст упродовж усього відео, що зумовлює необхідність пошуку нових підходів до її вирішення. Найперспективнішим підходом до визначення зміни сцен і планів є архітектура візуальних трансформерів для відео. Ця архітектура є адаптацією трансформерів, спочатку створених для роботи з текстом, до задач, пов'язаних з аналізом відеоданих. Візуальні трансформери для відео використовують механізм самоуваги, що дозволяє ефективно визначати зв'язки між кадрами, враховуючи просторові та часові залежності [23]. При цьому цей механізм дозволяє виділяти не лише локальні часові залежності, як-от при використанні рекурентних нейронних мереж чи тривимірних згорткових мереж, а й глобальні часові залежності. Це робить візуальні трансформери для відео надзвичайно ефективним інструментом для визначення контексту у відео, що є критично важливим для вирішення задачі знаходження змін сцен у відеоконтенті.

2.2.1. Архітектура ViViT

Архітектура візуального трансформера для відео є розширенням архітектури візуальних трансформерів, адаптованої для роботи з відеоданими за рахунок додавання аналізу часових характеристик. Основний принцип роботи візуальних трансформерів для відео полягає у представленні відео як послідовності зображень-патчів та їхньому подальшому аналізі за допомогою спеціальних шарів трансформерів, таких як Multi-Head Self-Attention (MHSA),

шар нормалізації та feed-forward networks (FFN) [88]. Хоча архітектура візуальних трансформерів для відео має декілька варіацій, найефективнішою для вирішення задачі визначення зміни сцен у відеоконтенті є варіант із використанням Factorized Encoder. Ця варіація дозволяє розділяти аналіз відеоконтенту на дві частини: просторовий трансформер для визначення просторових особливостей усередині кадрів та часовий трансформер для аналізу взаємозв'язків між кадрами та визначення часових особливостей. Це дозволяє ефективно аналізувати просторові та часові характеристики відеоконтенту, здійснювати багаторівневий аналіз та точно визначати контекст кожної сцени [89]. Модель візуального трансформера можна представити як

$$\begin{aligned} X_{\text{spatial}} &= T_{\text{spatial}}(X_{\text{in}}; \theta_{\text{spatial}}) \\ X_{\text{out}} &= T_{\text{temporal}}(X_{\text{spatial}}; \theta_{\text{temporal}}) \end{aligned} \quad (2.1)$$

де T_{spatial} - представленням просторового трансформеру, θ_{spatial} - параметри просторового трансформеру, T_{temporal} - представлення часового трансформеру, θ_{temporal} - його параметри, X_{in} - вхідні патчі сформовані з відео, X_{out} - фінальний представник відео для класифікації.

При аналізі даних відео спочатку розбивається на патчі. Ціле відео перед аналізом можна представити у вигляді тензора:

$$V \in \mathbb{R}^{T \times H \times W \times C} \quad (2.2)$$

, де T це кількість кадрів у відео, H, W представляють висоту та ширину кожного кадру та C є кількість кольорових каналів. Далі відео розбивається на просторово-часові патчі $p \times p \times t$, де p є розміром просторових патчів, а t

представляє кількість кадрів у патчі. Далі за допомогою лінійної проекції кожен патч перетворюється у вектор за допомогою наступної формули:

$$z_i = W_E \cdot x_i + b_E, i = 1, \dots, N \quad (2.3)$$

, де x_i це i -й патч відер, W_E -матриця проекції, b_E – зміщення даних та N це загальна кількість патчів. Далі необхідно додати позиційне кодування, так як механізм самоуваги не враховує позиційний порядок елементів. Додавання позиційного кодування дозволяє додавати інформацію про положення кожного патчу та обраховується за наступною формулою:

$$PE(i, 2j) = \sin\left(\frac{i}{10000^{\frac{2j}{d}}}\right), PE(i, 2j + 1) = \cos\left(\frac{i}{10000^{\frac{2j}{d}}}\right) \quad (2.4)$$

, де i це індекс патча, j представляє координати у векторі та d є розмірністю векторного представлення патча. Далі позиційне кодування додається до векторів патчів за формулою:

$$z'_i = z_i + PE_i \quad (2.5)$$

Далі архітектура візуального трансформеру передає отримані результати до просторового трансформеру і отримані результати передаються до часового трансформеру. Ці трансформери складаються з MHSA, шару нормалізації та FFN. Механізм MHSA дозволяє патчам взаємодіяти між собою утворюючи зв'язки між просторовими та часовими залежностями. Для цього для кожного представлення необхідно обчислити ключі – K , запити – Q , та значення – V , використовуючи відповідні формули:

$$Q = ZW_Q, K = ZW_K, V = ZW_V \quad (2.6)$$

, де $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ є матрицями параметрів. Далі на основі ключів, запитів та значень розраховуються коефіцієнти уваги:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \\ d_k = d/h \quad (2.7)$$

, де h це кількість голів уваги. Далі кожен патч отримує оновлене представлення за формулою

$$\text{head}_i = A_i V_i, i = 1, \dots, h \quad (2.8)$$

, та на основі цього розраховується вихідна матриця самоуваги:

$$Z = \text{concat}(\text{head}_1, \dots, \text{head}_h)W_O \quad (2.9)$$

, де $W_O \in \mathbb{R}^{hd_k \times d}$ є матрицею вихідного перетворення. Далі отримані результати передаються до двох повнозв'язних шарів:

$$FFN(Z) = \max(0, ZW_1 + b_1)W_2 + b_2 \quad (2.10)$$

, де W_1, W_2 є ваговими коефіцієнтами, а b_1, b_2 є відповідними зміщеннями. Далі отримані результати нормалізуються за формулою

$$Z' = \text{LN}(Z + FFN(Z)) \quad (2.11)$$

Отримані результати після аналізу передаються до останнього шару для отримання результатів. У випадку з задачею класифікації останній шар розраховується за формулою:

$$\hat{y} = \text{softmax}(W_C Z_C) \quad (2.12)$$

, де W_C є матрицею класифікації, а Z_C представленням відео. Далі в процесі навчання застосовується модель навчається мінімізувати крос-ентропійну функцію втрати:

$$\mathcal{L} = - \sum_i y_i \log \hat{y}_i \quad (2.12)$$

В результаті роботи архітектури візуальних трансформерів для відео вхідне відео розбивається на патчі, до кожного патча додається позиційне кодування. Далі просторовий трансформер визначає просторові особливості кожного кадру та передає результати аналізу часовому трансформеру, який визначає часові зв'язки між кадрами. Результат аналізу часового трансформера передається до повнозв'язного шару, де відбувається класифікація.

2.2.2. Застосуванні візуальних трансформерів для відео при визначенні зміни сцен

Архітектура візуальних трансформерів для відео відповідає всім вимогам для вирішення задачі визначення зміни сцен, оскільки вона дозволяє ефективно виділяти просторові та часові залежності у відеоконтенті. Проте задача визначення зміни сцен має низку складностей, зокрема визначення контексту зміни сцени, малий обсяг даних для навчання та складність

створення нових наборів даних через трудомісткість процесу та суб'єктивність людини при визначенні меж сцен.

Один із потенційних шляхів вирішення цієї задачі — можливість навчання візуального трансформера для відео визначати не тільки переходи між сценами, а й переходи між планами у відеоконтенті. Зміна сцени є одночасно і зміною плану, проте різниця між зміною планів та зміною сцен полягає в тому, що плани всередині сцени пов'язані між собою концептуально, тоді як концепція сцени загалом відрізняється. На основі цього припущення можна зробити висновок, що модель, яка одночасно визначає зміну планів та сцен, може виділяти якісніший контекст відео та демонструвати кращі результати.

Іншим потенційним шляхом розвитку моделі може бути зменшення обсягу вхідних даних з метою ефективної обробки тривалого відеоконтенту. Це може досягатися за рахунок виділення ключових кадрів та передачі лише їх до моделі візуального трансформера для відео. Складність цього підходу полягає у визначенні ключових кадрів, оскільки при класичному підході після отримання ключових кадрів немає чітких меж, за якими модель могла б визначати зміну сцени.

Для вирішення цієї проблеми може застосовуватися попереднє розбиття відео на плани та використання кадрів із планів з метою визначення зміни сцени. При такому підході, у випадку визначення зміни сцени, буде можливість точно визначити час, коли ця зміна відбулася, оскільки вона відповідатиме зміні плану.

2.2.3. Ефективність використання додаткових виходів моделі під час навчання моделей.

З метою дослідження ефективності використання додаткових виходів моделі було вирішено провести експерименти на архітектурі сіамських

нейронних мереж. Це дозволило швидко провести велику кількість експериментів та протестувати різні гіпотези. Сіамські нейронні мережі — це спеціальний тип нейронних мереж, призначений для вирішення задачі порівняння та виявлення схожості між вхідними даними. Важливою особливістю таких нейронних мереж є їхня здатність навчатися на обмежених обсягах даних. Також цей підхід дозволяє ефективно визначати схожість об'єктів завдяки тому, що схожість оцінюється як відстань між векторами у багатовимірному просторі [90-92]. Найбільш поширеним методом визначення відстані між порівнюваними об'єктами є евклідова відстань, яка розраховується за формулою:

$$D_W(\vec{X}_1, \vec{X}_2) = \|G_W(\vec{X}_1) - G_W(\vec{X}_2)\|_2 \quad (2.13)$$

де \vec{X}_i — вхідні зображення, G_W — функція перетворення, в нашому випадку це нейронна мережа, D_W — відстань між зображеннями. Тут важливо звернути увагу, що гілки сіамської нейронної мережі мають однакову архітектуру та спільні, синхронізовані ваги. Це дозволяє після навчання моделі залишити тільки одну гілку та з її допомогою проводити необхідні розрахунки. При такому використанні сіамської нейронної мережі відкривається можливість викликати модель лише один раз для кожного елемента та зберігати результати розпізнавання для подальшого їх порівняння. Це може бути особливо корисним при задачі класифікації, оскільки дозволяє один раз проаналізувати та зберегти в базі векторне представлення необхідних об'єктів [93]. Проте, оскільки під час навчання використовуються обидві гілки, і результатом навчання є векторні представлення даних, необхідно застосовувати спеціальну функцію втрат. Найбільш поширеною є *contrastive loss*, яка обчислює помилку відносно очікуваного результату, виходячи з отриманої відстані, та розраховується за формулою:

$$L(W, Y, \vec{X}_1, \vec{X}_2) = (1 - Y) \frac{1}{2} (D_W)^2 + (Y) \frac{1}{2} \{\max(0, m - D_W)\}^2 \quad (2.14)$$

де W – параметри системи, Y – очікувана відстань між вхідними даними, m – очікувана відстань між різними вхідними даними [94, 95].

Отже, метою цього дослідження було покращення точності моделі шляхом штучного акцентування уваги моделі на додаткові особливості при порівнянні зображень. Розроблений алгоритм мав покращити результати навчання шляхом додавання додаткових виходів, які вирішували б задачі класифікації, регресії або їхньої комбінації. Це дослідження було мотивоване роботами, присвяченими розробці моделі GoogLeNet, яка використовувала додаткові виходи з класифікацією на різних етапах моделі для покращення точності, пришвидшення навчання та зменшення впливу згасаючого градієнта у глибоких нейронних мережах [96]. Для цих експериментів була розроблена архітектура, яка дозволяє додавати до двох додаткових виходів до моделі, як продемонстровано на рисунку 2.1.

Для навчання таких моделей необхідно модифікувати датасети з метою генерації додаткової розмітки. Для цих експериментів було вирішено, що додаткові виходи будуть вирішувати простішу задачу, ніж основний вихід програми. Також було вирішено провести експерименти для задач різної складності, а саме на датасетах Fashion MNIST та PlantVillage. Для першого експерименту був обраний датасет Fashion MNIST, оскільки він є сучасною альтернативою традиційному датасету для розпізнавання рукописних чисел та широко використовується при тестуванні підходів у галузі комп'ютерного зору.

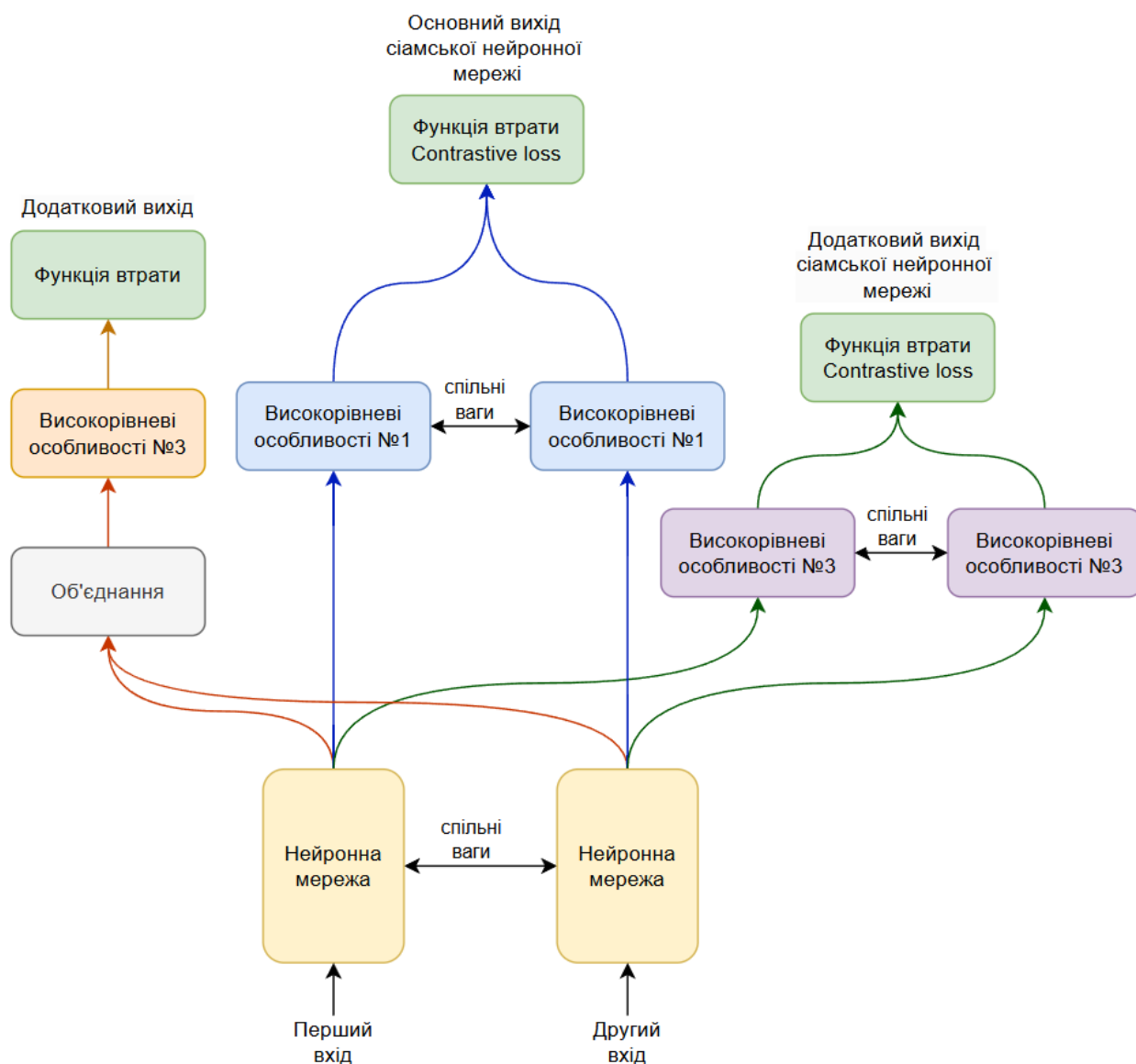


Рисунок 2.1 – Архітектура з додатковими виходами для визначення додаткових особливостей

Цей датасет містить 70 тисяч чорно-білих зображень у десяти різних категоріях одягу, таких як футболки, штани, сукні та інші, як продемонстровано на рисунку 2.2 [97].



Рисунок 2.2 - Класи у датасеті Fashion MNIST.

Для проведення експерименту був розроблений алгоритм, який приймав на вхід датасет Fashion MNIST та його модифіковану версію. Спочатку відбувалося завантаження датасету та генерація пар для сіамської нейронної мережі. Далі на основі цих пар було згенеровано розмітку для додаткових виходів моделі. Після цього датасет розбивався на тренувальну, валідаційну та тестову вибірки. Під час навчання дані, перед надходженням до моделі, проходили базові трансформації, такі як віддзеркалення, часткове обрізання, зміна яскравості зображення. Далі відбувалося тренування звичайної сіамської нейронної мережі, а також версій цієї моделі з додатковим виходом для класифікації, використовуючи різні коефіцієнти для функції втрат. Обидві моделі мали однакову базову архітектуру та спільну функцію оптимізації. В результаті експерименту було встановлено, що оригінальна сіамська нейронна мережа без додаткових виходів показує кращу точність та швидше навчається, досягаючи точності 92,3%.

Таблиця 2.1. Порівняння результатів навчання сіамської нейронної мережі з використанням додаткового виходу класифікації

Додаткові виходи моделі	Точність	Точність відносно оригінальної моделі
Класифікація, ваги функції втрат 100%	90,08%	-2,22%
Класифікація, ваги функції втрат 50%	90,93%	-1,37%
Класифікація, ваги функції втрат 20%	91,75%	-0,55%
Класифікація, ваги функції втрат 10%	92,07%	-0,23%

Результати експериментів показали, що сіамська нейронна мережа без додаткових виходів краще справляється із задачею порівняння, демонструючи вищу точність та швидкість сходження моделі, ніж сіамська нейронна мережа з додатковим виходом. Тому було вирішено провести додаткові експерименти, у яких важливість додаткових виходів зменшували під час обчислення функції помилки. В результаті додаткових експериментів було встановлено, що зі зменшенням впливу додаткових виходів на загальний процес тренування моделей точність та швидкість сходження починали зростати [7].

Для наступного експерименту було вирішено взяти частину датасету PlantVillage. Розмір цього датасету становив 20 тисяч зображень та містив 15 класів для розпізнавання, як показано на рисунку 2.3 [98].

З метою виділення особливостей було вирішено розділити цей датасет на групи, а саме за визначенням класу, до якого належить зображення, та за станом об'єкта на зображенні. Такий розподіл дозволив виділити три класи для задачі розпізнавання, а саме: "Pepper", "Potato" та "Tomato". Також датасет було розділено на дві частини, які включали стан рослини — здорові чи хворі, що стало задачею регресії. Таким чином, додаткові виходи моделі вирішували не повну задачу, а лише частину оригінальної задачі. Таке розділення є дуже

важливим для нашого дослідження, оскільки воно концептуально відповідає задачі одночасного визначення планів та сцен у відеоконтенті.

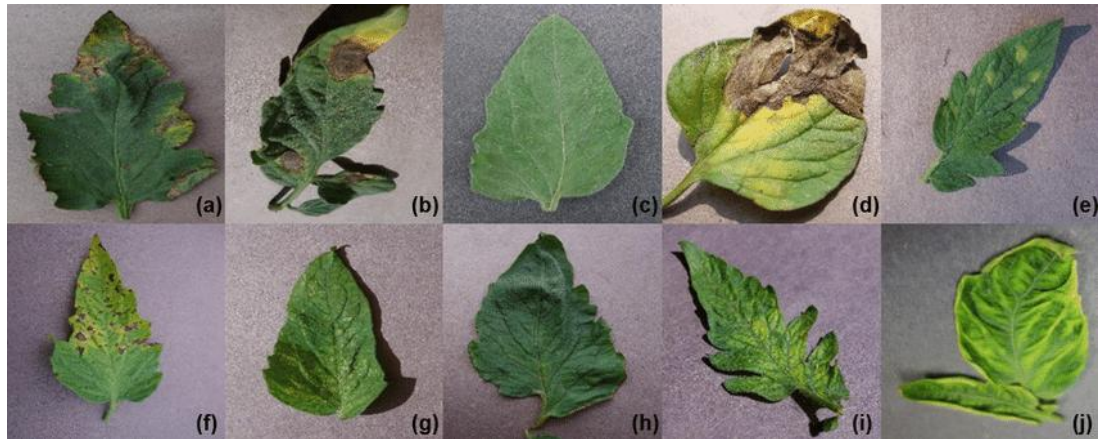


Рисунок 2.3 - Зображення з датасету PlantVillage

Перший алгоритм навчання було модифіковано шляхом додавання додаткового виходу для вирішення задачі регресії. Також було вирішено, замість неглибокої згорткової нейронної мережі, використати модель архітектури ResNet50V2, завдяки її глибокій архітектурі та можливості використовувати попередньо натреновані ваги на датасеті ImageNet, який застосовується для задачі класифікації та містить мільйони розмічених зображень[99-101]. Під час тренування порівнювалася точність при використанні одного додаткового виходу для задач класифікації та регресії, а також двох додаткових виходів одночасно, як продемонстровано на рисунку 2.4.

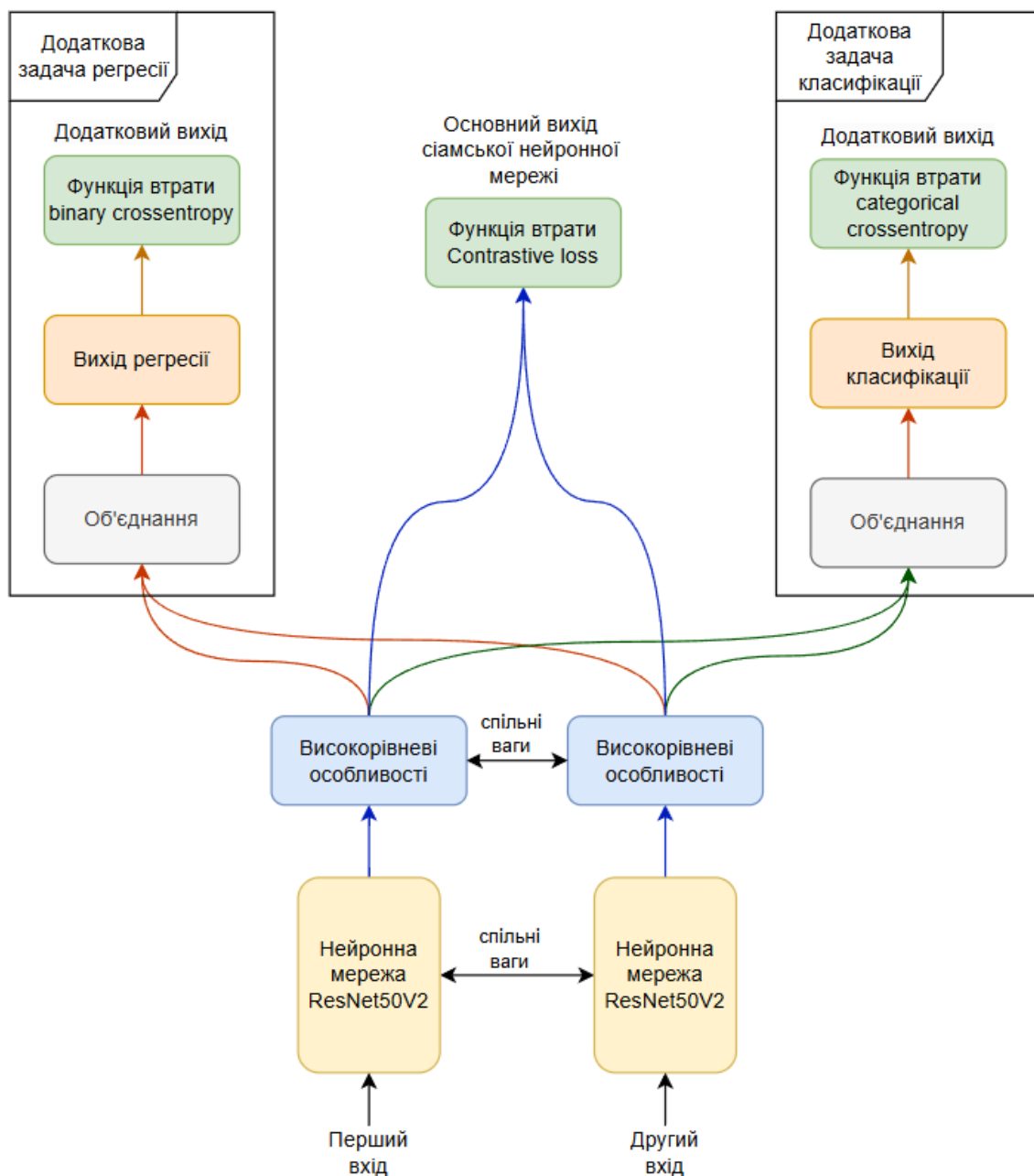


Рисунок 2.4 - Сіамська нейронна мережа з двома додатковим виходами та застосуванням моделі ResNet50V2 як основи сіамської нейронної мережі

Також були застосовані різні ваги для функції втрат. В результаті навчання сіамська нейронна мережа без додаткових виходів показала точність 96.54%.

Таблиця 2.2 - Порівняння результатів навчання сіамської нейронної мережі з використанням додаткових виходів виходу класифікації

Додаткові виходи	Точність	Точність відносно оригінальної моделі
Класифікація та регресія, ваги функції втрат 100%	95,19%	-1,35%
Класифікація та регресія, ваги функції втрат 50%	97,26%	+0,72%
Класифікація та регресія, ваги функції втрат 20%	96,20%	-0,34%
Класифікація та регресія, ваги функції втрат 10%	96,90%	+0,36%
Класифікація, ваги функції втрат 100%	91,00%	-5,54%
Класифікація, ваги функції втрат 50%	94,35%	-2,19%
Класифікація, ваги функції втрат 20%	95,43%	-1,11%
Класифікація, ваги функції втрат 10%	96,24%	-0,30%
Регресія, ваги функції втрат 100%	93,92%	-2,62%
Регресія, ваги функції втрат 50%	94,62%	-1,92%
Регресія, ваги функції втрат 20%	96,61%	+0,07%
Регресія, ваги функції втрат 10%	96,60%	+0,06%

В результаті експериментів було визначено, що найкращі результати було досягнуто при застосуванні двох додаткових виходів моделі зі зменшенням впливу їхніх функцій втрат на 50%. Проте, аналізуючи всі результати, стає зрозуміло, що цей підхід не забезпечує стабільності та потребує великої кількості експериментів для визначення оптимальних параметрів, що призводить лише до незначного збільшення загальної точності моделей. Крім того, цей підхід вимагає значних витрат на підготовку даних та розробку додаткових архітектур для тренування. Також під час навчання великих моделей тестування додаткових виходів може бути надто тривалим і нестабільним, звідси можна зробити висновок, що використання цього методу є недоцільним для вирішення задачі розпізнавання сцен [7].

2.2.4. Визначення зміни планів на відео з метою визначення потенційних країв сцен та ключових кадрів для аналізу.

Визначення планів є важливою задачею при аналізі відеоконтенту та дозволяє ефективніше визначати атрибути для відео шляхом визначення ключових кадрів та зменшення необхідної кількості запитів для розпізнавання атрибутів. Класичні підходи до визначення зміни плану ґрунтуються на застосуванні математичних методів, таких як розрахунок різниці між кадрами чи аналіз гістограм. Хоча ці підходи є ефективними у сценаріях, коли відео містять низьку динаміку або навіть статичні кадри, а також при різких переходах між планами, їх застосування до сучасного відеоконтенту призводить до стрімкого падіння точності [102].

Це викликано зростанням кількості динамічних сцен зі складною композицією та складними візуальними переходами. Можливими шляхами вирішення цієї проблеми є розробка нових математичних підходів до визначення переходів планів у відеоконтенті або розробка нових методів із застосуванням нейронних мереж. Проте при використанні нейронних мереж значно зростає потреба в обчислювальних ресурсах, що може зробити визначення зміни планів у відеоконтенті малоефективним. Це зумовлено необхідністю нейронної мережі одночасно аналізувати просторові та часові залежності.

До таких моделей належать тривимірні згорткові нейронні мережі, візуальні трансформери для відео та поєднання згорткових нейронних мереж із рекурентними нейронними мережами або моделями архітектури трансформерів.

З метою збереження швидкодії було вирішено дослідити поєднання математичних підходів із нейронними мережами. Це поєднання дозволяє швидко визначати візуальні атрибути відео за допомогою математичних

підходів, а за допомогою простих неглибоких рекурентних нейронних мереж класифікувати зміни планів, враховуючи часові зміни.

Для вирішення задачі визначення зміни планів було вирішено спочатку модифікувати існуючі математичні підходи, які можуть ефективно збирати важливу інформацію із зображень для подальшого використання у визначенні зміни плану. Запропонований алгоритм передбачає розбивання зображення на блоки, створення візуальних представлень зображення у різних кольорових просторах та подальше обчислення гістограм.

Нехай L позначає кількість кадрів, B_i позначає i -й блок у кадрі, а C представляє кількість блоків, створених під час процесу поділу, причому кожен блок має однакову форму, як показано на рисунку 2.5 [103].

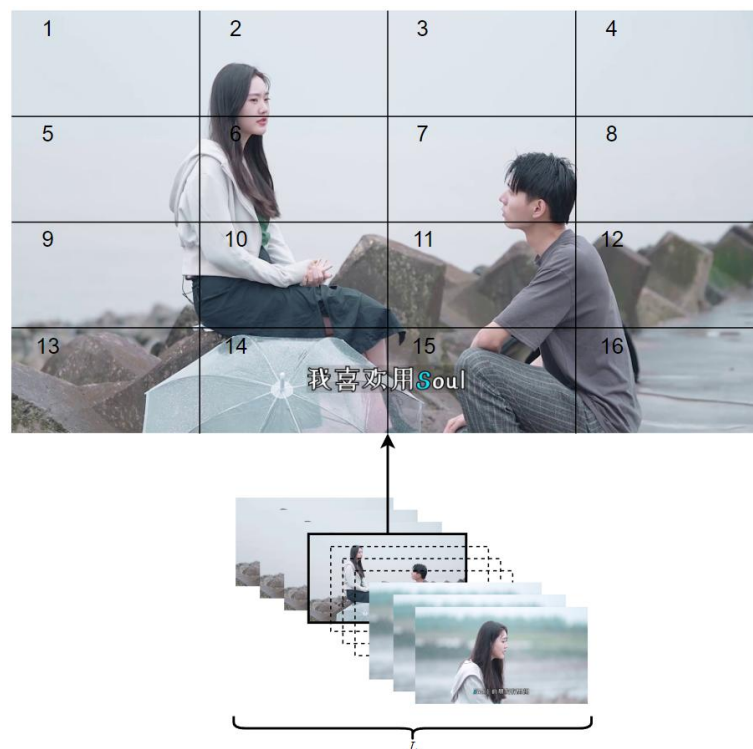


Рисунок 2.5 - Розбиття кадру на блоки

Для кожного блоку B_i у різних кольорних просторах створюються відповідні представлення [104]. Експериментальним шляхом було визначено, що оптимальними кольорними просторами є градації сірого та HSV (Hue,

Saturation, Value). Також було виявлено, що достатньо використовувати лише насиченість і яскравість із спектру HSV. Ці представлення позначаються як B_i^{gray} , $B_i^{saturation}$ та B_i^{value} відповідно [105]. Потім обчислюються гістограми для кожного блоку в кожному представленні даних, позначені як H_i^{gray} , $H_i^{saturation}$ та H_i^{value} . Кожна гістограма стискає кількість даних у діапазон $[0; C_h]$ [106]. Крім того, для кожного B_i^{gray} ми обчислюємо контури за допомогою оператора Собеля-Фельдмана, отримуючи B_i^{sobel} , після чого виконується обчислення гістограми, яке позначається як H_i^{sobel} як показано на рисунку 2.6 [107].



Рисунок 2.6 - Представлення зображення в різних колірний просторах та з виділенням країв

Відстань між гістограмами обчислюється за допомогою формули:

$$d(a, b) = \sqrt{\sum_{j=1}^{c_h} (a_j - b_j)^2} \quad (2.15)$$

де a та b представляють гістограми.

Потім відстані між гістограмами об'єднуються в один список:

$$d_i = d(H_i^{gray}, H_{i+1}^{gray}) \cup d(H_i^{saturation}, H_{i+1}^{saturation}) \cup d(H_i^{value}, H_{i+1}^{value}) \cap d(H_i^{sobel}, H_{i+1}^{sobel}) \quad (2.16)$$

Отже, різницю між відповідними блоками у двох кадрах можна обчислити як:

$$D_i = \frac{1}{4C_h} \sum_{j=1}^{j=4C_h} (d_{ij}) \quad (2.17)$$

отримуючи єдине значення, що позначає відстань між цими блоками [9].

Далі обчислюються відстані між кадрами:

$$D_i^{frame} = \bigcup_{j=1}^C (D_{ij}) \quad (2.18)$$

Відстань між сусідніми кадрами можна обчислити за допомогою наступної формули:

$$D = \bigcup_{i=1}^L (D_i^{frame}) \quad (2.19)$$

Техніки виявлення аномалій застосовуються до гістограм для виявлення відхилень від очікуваних розподілів. Нехай \underline{D} представляє середнє значення D , а σ — стандартне відхилення D . Це дозволяє ідентифікувати блоки між кадрами, які відхиляються від загального розподілу:

$$D_{ij}^{map} = \begin{cases} 1: D_{ij} > \underline{D} + \sigma \\ 0: D_{ij} \leq \underline{D} + \sigma \end{cases} \quad (2.20)$$

Потім відхилення для всіх різниць між кадрами на основі відхилених блоків визначаються як:

$$A = \left\{ D_i \mid \sum_{j=1}^c (D_{i,j}) > \underline{D^{map}} + \sigma^{map} * k \right\} \quad (2.21)$$

де $\underline{D^{map}}$ та σ^{map} це середнє значення та стандартне відхилення для розподілу D^{map} відповідно, а k є коефіцієнтом, який визначає чутливість порогу виявлення аномалій.

Для проведення експерименту було обрано датасет SHOT. Він містить 853 короткі відео, які загалом налічують 960 794 кадри та 6 100 планів [108]. Важливою особливістю цього набору даних є його актуальність, що забезпечує велике розмаїття відео та значну кількість складних переходів, зокрема поступові переходи, як показано на рисунку 2.7.



Рисунок 2.7. Переходи в наборі даних SHOT

При проведенні першого експерименту спочатку було завантажено набір даних, і для кожного відео було вилучено всі кадри. Далі кожен кадр розбивався на блоки, перетворювався у необхідні кольорові простори та обчислювалися контури за допомогою оператора Собеля-Фельдмана. Далі для кожної комбінації блоку та кольорового простору було обчислено гістограму, а потім ці гістограми порівнювалися між сусідніми кадрами. Після цього визначалося відхилення між блоками з метою виявлення аномалій, які відігравали роль змін плану. Після проведення цих операцій визначалися точність (Precision) та показник F1.

Під час проведення експерименту було протестовано різні варіанти кількості блоків, колірні простори, розміри гістограм та коефіцієнти для визначення порогу зміни сцени. Було встановлено, що оптимальною кількістю блоків є 64, які рівномірно розташовані на зображенні без зон перекриття та повністю покривають все зображення. Також було досліджено різні комбінації колірних просторів, у результаті чого встановлено, що використання сірого колірного простору може повністю замінити тривимірний колірний простір RGB. Проте цієї інформації було недостатньо для ефективного визначення зміни планів, тому було додано колірний простір HSV. Під час детального аналізу кожного виміру цього колірного простору встановлено, що канал відтінку (Hue) містив надлишкову інформацію, що дозволило виключити його з розрахунків з метою пришвидшення алгоритму без втрати точності. Також, як додатковий канал, було вирішено використовувати зображення, створене за допомогою визначення країв. Для визначення країв було вирішено використовувати оператор Собеля-Фельдмана через його швидкість та здатність ефективно виявляти межі об'єктів чи регіонів, де інтенсивність змінюється швидко. Під час такого аналізу для кожного пікселя обчислюються горизонтальні та вертикальні градієнти за допомогою ядер Собеля. На основі цих градієнтів розраховується магнітуда градієнта, яка вказує на величину зміни інтенсивності, та напрямок градієнта, який є кутом, під яким змінюється

розрахована інтенсивність. На основі визначених магнітуди та напрямку градієнта будується зображення з визначеними краями. Таке зображення слугує додатковим простором та дозволяє під час аналізу гістограм відстежувати різкі зміни у положенні об'єктів між кадрами.

При розрахунку гістограм було виявлено, що використання стандартного розміру гістограми в 256 бінів містить надлишкову інформацію, що призводить до генерації зайвих шумів при обчисленні відстані між гістограмами за допомогою евклідової відстані. Шляхом експериментів встановлено, що найкращих результатів можна досягти при зменшенні кількості бінів у гістограмі до 64. Проте, при аналізі результатів розробленої математичної моделі було виявлено, що запропонований алгоритм виявився неефективним та ненадійним у порівнянні з методами на основі нейронних мереж.

Таблиця 2.3. Порівняння математичного підходу з моделями TransNetV2 та AutoShot

Метод	F1	Precision
TransNetV2	0.799	0.904
AutoShot@F1	0.841	0.923
AutoShot@Precision	0.826	0.939
Математичний підхід	0.473	0.448

Незважаючи на детальне дослідження різних варіантів набору параметрів, математичний підхід виявився неефективним при аналізі сучасного контенту. Основна проблема полягала у складності розробки адаптивного алгоритму виявлення аномалій при зміні планів математичними методами. Тому було вирішено провести експеримент, у якому детектором зміни планів було обрано нейронну мережу.

Для навчання нейронної мережі всі відео було оброблено, і на основі отриманих даних було згенеровано послідовності для навчання, довжина яких становила 50 кадрів, як показано на рисунку 8. Також, з метою зменшення хибних визначень зміни планів, було згенеровано додаткові послідовності, у яких зміна плану могла знаходитися близько до виходу моделі.

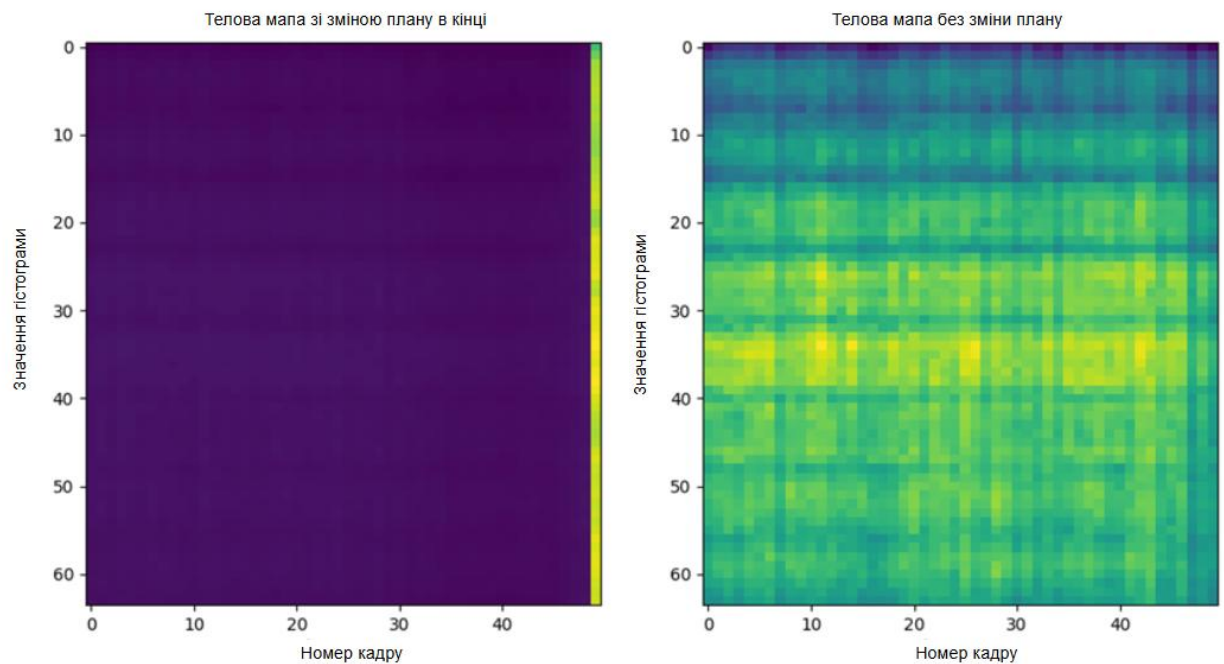


Рисунок 2.8 - Теплові карти тренувальних даних для істинних і хибних міток відповідно

Для визначення часових залежностей було обрано рекурентні нейронні мережі. Перевагою такого вибору була можливість адаптації цієї моделі для роботи в реальному часі та обробки потоків даних. У ході експериментів було вирішено використовувати рекурентні нейронні мережі типу Long Short-Term Memory (LSTM) та Gated Recurrent Unit (GRU) [109]. Модель LSTM — це вид рекурентної нейронної мережі, який був розроблений для кращої роботи з довгостроковими часовими залежностями у послідовних даних. В архітектурі моделі LSTM є три основні компоненти:

- Forget Gate — відповідає за контроль попереднього стану комірки та визначає, яку інформацію необхідно забути.
- Input Gate (вхідні ворота) — відповідає за додавання нової інформації до стану комірки.
- Output Gate (вихідні ворота) — контролює, яку інформацію потрібно передавати на вихід.

Також моделі LSTM мають механізм cell state, який допомагає моделі ефективніше навчатися та зменшує ймовірність виникнення затухаючих і вибухаючих градієнтів [35]. Крім того, було розглянуто використання моделей типу GRU. На відміну від моделей LSTM, моделі GRU мають меншу кількість параметрів та спрощену архітектуру. Для GRU визначають два основні компоненти:

- Reset Gate — визначає, яку частину попереднього стану комірки потрібно забути або виключити при розрахунку нового стану.
- Update Gate — відповідає за керування станом та визначає, яка частина стану має бути оновлена.

На відміну від моделей типу LSTM, моделі GRU не мають cell state, а замість нього використовується один внутрішній стан, що, у свою чергу, спрощує архітектуру та зменшує кількість параметрів. Завдяки цьому моделі типу GRU навчаються швидше, ніж моделі LSTM, проте можуть мати нижчу точність.

При порівнянні LSTM та GRU було встановлено, що при використанні одного рекурентного шару модель на основі GRU має кращу точність сходження та F1-оцінку. Проте, зі збільшенням кількості рекурентних шарів точність LSTM починає зростати та перевершує точність GRU. Також було встановлено, що при збільшенні кількості рекурентних шарів понад два точність починає значно погіршуватися. Кількість параметрів у рекурентних нейронних мережах виявила залежність, за якої при використанні меншої кількості параметрів зростають точність влучання та повнота, але падає F1-

оцінка. З цього можна зробити висновок, що залежно від задачі можна максимізувати точність влучання або повноту, або зосередитися на більш збалансованому підході. Також було протестовано різну кількість повнозв'язних шарів, які йшли після рекурентних шарів. Максимізація точності влучання (98,2%) за рахунок втрати F1-оцінки (83,3%) була досягнута при використанні двох повнозв'язних шарів, що слідували за двома LSTM-шарами. Проте, при збільшенні кількості параметрів у шарах LSTM та трьох повнозв'язних шарах вдалося досягти більш збалансованого результату, при якому точність влучання та F1-оцінка дорівнювали 93,9% та 88,5% відповідно, що продемонстровано в таблиці 4.

Таблиця 2.4 - Порівняння математичного підходу з використанням нейронних мереж з сучасними моделями.

Метод	F1	Precision
TransNetV2	0.799	0.904
AutoShot@F1	0.841	0.923
AutoShot@Precision	0.826	0.939
Математичний підхід	0.473	0.448
Математичний підхід з використанням нейронної мережі(2 LSTM(8) and 2 Layers)	0.833	0.982
Математичний підхід з використанням нейронної мережі (2 LSTM(128) and 3 Layers)	0.885	0.939

У ході експериментів було досягнуто показників, які перевищують точність архітектур TransNetV2 та AutoShot. При цьому цей підхід також має переваги у можливості вибору між підходами, що включають максимізацію точності влучання, повноти або F1-оцінки. Також цей підхід вигідно відрізняється компактним розміром моделі та низькими обчислювальними вимогами. Розроблені нейронні мережі використовують від 6 kFLOPs до 500

kFLOPs на один фрейм, що дозволяє застосовувати цей підхід для розпізнавання в реальному часі [8].

2.3. Метод розбиття відео на сцени за допомогою візуального трансформеру для відео

Основною складністю при аналізі сучасного відеоконтенту з метою визначення сцен є потреба чітко визначати контекст кожної сцени. При використанні візуальних трансформерів для відео виникає проблема: зі збільшенням кількості кадрів, що надходять на вхід моделі, значно зростає обчислювальна складність. Це зумовлено квадратичною складністю механізму уваги у моделях візуальних трансформерів для відео. Проте, якщо кількість таких кадрів буде недостатньою, модель може отримати замало інформації для точного визначення контексту та границь сцени. Альтернативні підходи, такі як застосування рекурентних нейронних мереж чи комбіновані методи, також мають проблеми зі збільшенням вхідної послідовності, оскільки зростають обчислювальні витрати. Крім того, додається проблема затухаючих та вибухаючих градієнтів. З метою вирішення цієї проблеми було вирішено підвищити ефективність передачі даних до моделі шляхом зменшення кількості кадрів при збереженні контексту пропущених кадрів.

Аналіз структури відеоконтенту показав, що сцени складаються з планів, змін положення камери, а також переходи між сценами є зміною плану. Крім того, аналіз показав, що, оскільки плани мають коротку тривалість та не містять великої кількості різких змін у відеопотоці, це дозволяє ефективніше передавати інформацію до моделей візуального трансформера для відео [110]. При такому підході відеоконтент спочатку розбивається на плани, далі з них вибирається необхідна кількість ключових кадрів, і модель візуального трансформера для відео визначає, які переходи між планами є частиною однієї

сцени, а які сигналізують про зміну сцени. Таким чином, необхідний розмір моделі для отримання комплексного представлення відеоконтенту значно зменшується, що дозволяє ефективніше використовувати архітектуру візуального трансформера для відео.

2.3.1. Вимоги до передобробки даних та перевірки результатів експериментів

Для проведення експериментів було поєднано набори даних з різних досліджень з метою максимізації обсягу доступних даних для навчання. Для цього було використано набори даних The Raid, OsVSD та BBC Planet Earth. Приклад переходів між сценами у наборах даних OsVSD та BBC Planet Earth продемонстровано на рисунку 2.9.

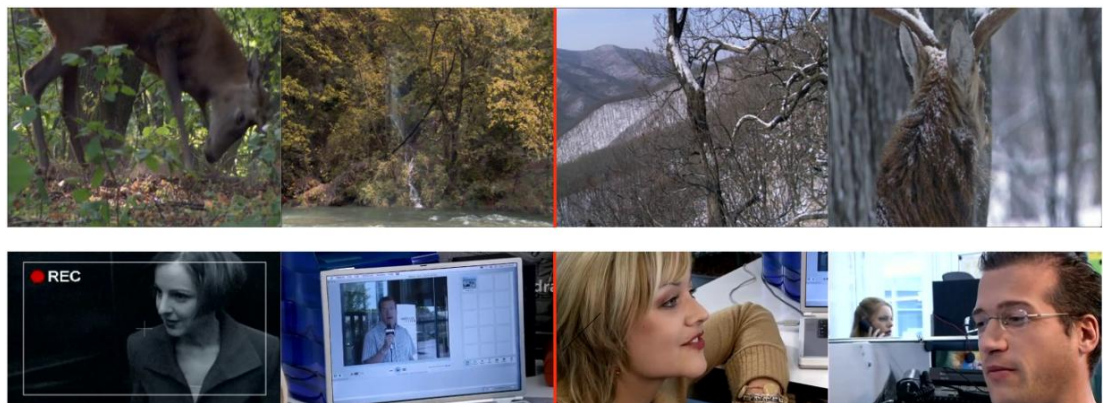


Рисунок 2.9 - Зміни сцен для наборів даних The Raid, OsVsd та BBC Planet Earth

З метою збереження можливості порівняння результатів проведених експериментів із дослідженнями інших учених було вирішено використовувати набір даних BBC Planet Earth як валідаційну вибірку. У результаті поєднання наборів даних The Raid та OsVSD було отримано тренувальний набір, який містив 31 відео з 7856 сценами. Валідаційний набір

даних BBC Planet Earth містив 11 відео з 4844 сценами. Під час підготовки даних до тренування стало зрозуміло, що отримана кількість сцен є недостатньою для ефективного навчання моделі візуального трансформера для відео, через що було вирішено модифікувати оригінальні датасети з метою збільшення обсягу тренувальних даних.

Підготовка датасету для тренування відбувалася у кілька етапів, першим з яких було розбиття кожного відео на плани. Для цього використовувався розроблений метод розбиття на плани, оскільки при його створенні вдалося досягти точності, що перевищує результати інших моделей для визначення зміни планів. Було створено спеціальну розмітку, у якій було зазначено, які плани належать до кожної сцени та які сцени є сусідніми. Це дозволило збільшити варіативність сцен під час тренування завдяки можливості брати різні комбінації планів та розташовувати їх у довільному порядку. Також, маючи таку розмітку, стало можливим змінювати сцени місцями під час генерування даних для тренування. Використання таких даних значно збільшує кількість доступних даних для аналізу, зберігаючи при цьому концепти, закладені в сцени. Для покращення результатів були створені різні комбінації послідовностей даних:

- послідовності, що містили зміну сцени, для навчання моделі виявляти їх;
- послідовності, що містили плани з однієї сцени, щоб модель могла навчитися адаптуватися до довгого контенту, коли зміна сцени може не відбуватися, попри значні зміни планів;
- послідовності, що містили зміну плану поблизу виходу моделі, що необхідно для навчання більш точно визначати момент зміни сцен за допомогою ключових кадрів із плану.

Цей підхід також відкриває можливість використання різної кількості кадрів для кожного плану, що дає змогу контролювати кількість контексту з кожного плану, який отримує модель архітектури візуального трансформера

на вхід. Для використання таких даних у навчанні було створено спеціальний клас, який мав доступ до підготовлених ключових кадрів для кожної сцени та розмітки, що містила інформацію про сцени, сусідні сцени та плани, прив'язані до кожної сцени. У результаті розробки було отримано пайплайн, зображений на рисунку 2.10.

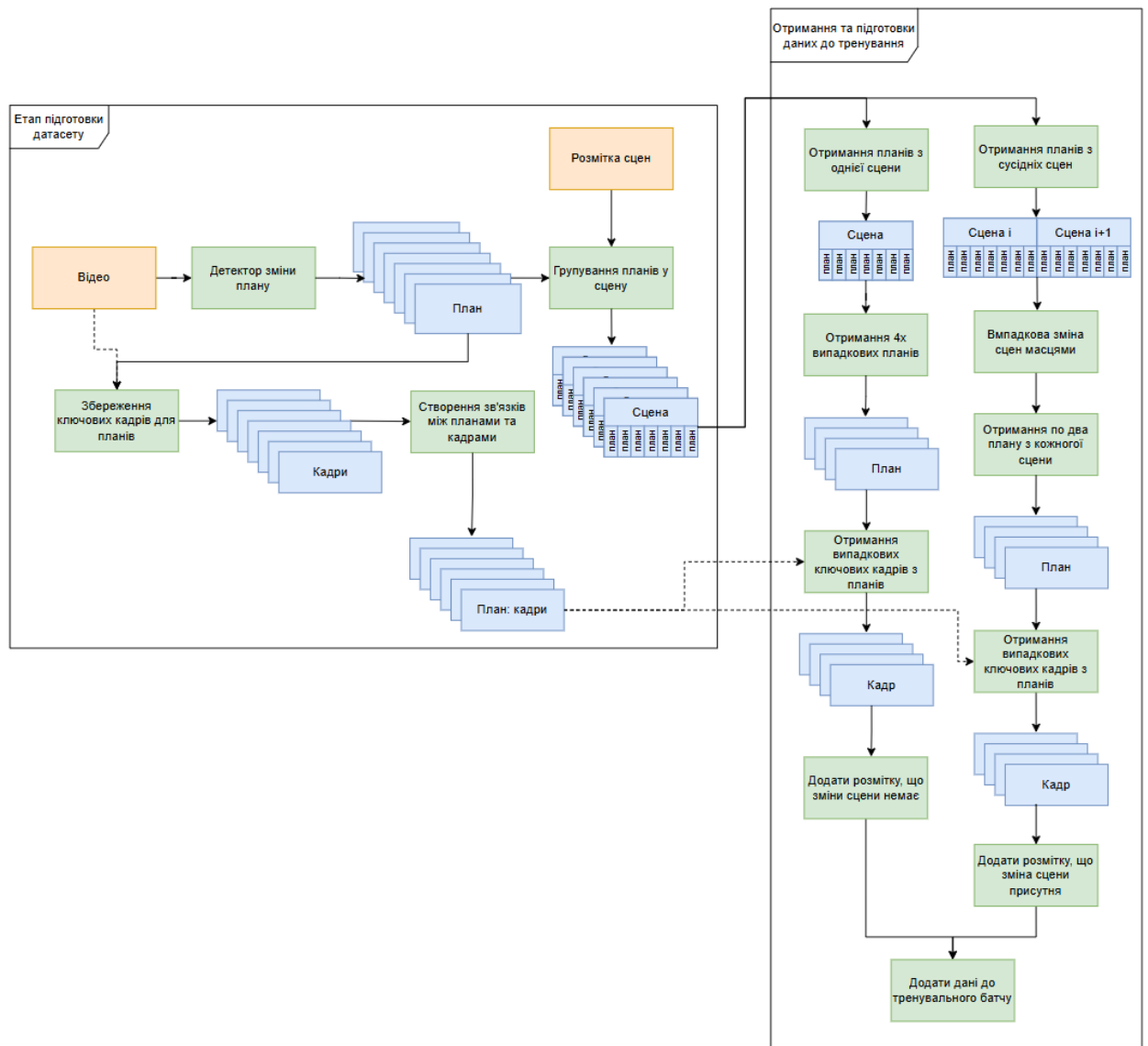


Рисунок 2.10 – Підготовка даних та генерація тренувальних даних.

Розроблений підхід для роботи з даними дозволив ефективно працювати з набором даних і швидко адаптуватися до потреб експериментів шляхом

зміни необхідної кількості ключових кадрів чи кількості планів для сцени. Також після генерації кожного батчу для тренування здійснювалися додаткові трансформації зображень з метою покращення узагальнювальної здатності моделі, зокрема обрізання зображення, випадкова зміна розміру, накладання кольорових фільтрів, застосування гаусового розмиття та інші методи аугментації. Для кожного зображення вибиралися випадкові трансформації та параметри для них, що дозволило зробити вхідні дані більш різноманітними, покращити узагальнювальну здатність розроблюваної моделі та зробити її більш стійкою до шумів.

2.3.2. Підвищення точності розпізнавання виявлення зміни сцен з використанням візуальних трансформерів для відео.

В результаті експериментів було встановлено, що для отримання максимальної точності достатньо використовувати по чотири плани для аналізу, при цьому необхідна лише один ключовий кадр для кожного плану. Умовою переходу між сценами є належність перших двох ключових кадрів з планів які належать першій сцена, та наступні два кадри мають належати наступній сцені. Цієї інформації виявилось достатньо для визначення критично важливої інформації для визначення контексту сцен та визначенню переходу між ними. При цьому при збільшенні кількості ключових кадрів чи збільшенню сцен для аналізу виникали проблеми пов'язані з цією розробленою архітектурою. Першою проблемою ставало значне збільшенні розміру моделей, що потребувало значного збільшення навчальних даних. Навіть з застосуванням розробленого алгоритму для штучного збільшення датасету цього виявилось недостатньо та модель починала стрімко перенавчатись. Проте така поведінка свідчить про те, що при розробці додаткових наборів даних для визначення зміни сцен цей підхід має потенціал для досягнення навіть більшої точності. Іншою проблемою з якою ми

стикнулись це сцени які містили малу кількість планів. Це характерно до сучасного контенту який застосовується в рекламі чи короткі ролики які несуть розважальний характер. Для вирішення цієї проблеми можна розробляти більш досконалий підхід до генерації даних які при включенні більшої кількості планів буде мати можливість мати в послідовності декілька переходів між сценами.

В результаті навчання моделі архітектури візуального трансформеру для відео з застосуванням планів отриманих завдяки розробленому підходу до розбиття відео на плани з використанням поєднання математичного підходу та рекурентних нейронних мереж вдалось розробити модель яка змогла досягнути оцінки F1 72.1%, що на 5.1% краще за Deep Multimodal Networks та перевершити показання інших моделей для визначення зміни сцен, як показано в таблиці 2.5.

Таблиця 2.5 - Порівняння результатів виявлення зміни сцен.

Модель	F1
Розроблена модель	0.721
Deep Multimodal Networks [3]	0.67
STG[4]	0.41
NW[5]	0.33

Також розроблена модель може бути додатково покращена шляхом застосування механізмів оптимізації, особливої уваги при цьому слід приділити прунінгу перед навчанням, так як це дозволить не тільки пришвидшити модель та зменшити її розмір, але також пришвидшить процес навчання такої моделі.

Розробка методів розбиття відео на сцени за допомогою візуальних трансформерів для відео та методів визначення зміни планів за допомогою поєднання математичних підходів з рекурентної нейронною мережею

відкривають можливість для пришвидшення процесу визначення відеоатрибутів шляхом зменшення необхідної кількості кадрів для аналізу. При цьому зберігається висока точність аналізу за рахунок можливості вибирати необхідну кількість ключових кадрів для кожної сцени чи плану. Також детальний розбір відео на складові компоненти дозволяє ефективно аналізувати відеоконтент та будувати детальну аналітику. Поєднання точності розроблених підходів та їх швидкодії дозволяє ефективно використовувати їх для аналізу сучасного відеоконтенту. Наступними кроками є оптимізація підходу до визначення зміни сцен у відеоконтенті за допомогою використання моделі на основі архітектури трансформер та розробка програмного забезпечення яке дозволить ефективно розбити відео на сцени та плани та на основі цього будувати детальну аналітику для відеоконтенту.

2.4. Висновки до розділу 2

У результаті експериментів було встановлено, що для досягнення максимальної точності достатньо використовувати чотири плани для аналізу, при цьому потрібен лише один ключовий кадр для кожного плану. Умовою переходу між сценами є те, що перші два ключові кадри належать планам першої сцени, а наступні два кадри – планам наступної сцени. Цієї інформації виявилось достатньо для визначення критично важливого контексту сцен та ідентифікації переходу між ними. Проте, зі збільшенням кількості ключових кадрів або сцен для аналізу виникали проблеми, пов'язані з архітектурою моделі. Першою проблемою стало значне збільшення розміру моделі, що вимагало суттєвого збільшення обсягу навчальних даних. Застосування розробленого алгоритму штучного розширення датасету виявилось недостатньо, і модель схильна до перенавчання. Така поведінка свідчить про те, що при розробці додаткових наборів даних для визначення зміни сцен цей підхід має потенціал до досягнення ще вищої точності. Іншою проблемою, з

якою довелося зіткнутись, були сцени з малою кількістю планів. Це характерно для сучасного контенту, який використовується в рекламі чи коротких розважальних відеороликах. Для вирішення цієї проблеми можна розробити більш досконалий підхід до генерації даних, який, включаючи більшу кількість планів, дозволить формувати послідовності з декількома переходами між сценами.

Розроблена модель продемонструвала можливість створення алгоритму, який, використовуючи підходи розбиття відео на плани та сцени, дозволяє значно скоротити кількість необхідних викликів моделей для визначення відеоатрибутів, зберігаючи високу точність обчислень. Також застосування цього підходу дозволяє ефективно аналізувати відеоконтент завдяки інформації про чіткі границі планів та сцен. Розроблений підхід має потенціал для подальшої оптимізації шляхом застосування методів оптимізації. Найбільший потенціал має метод прунінгу перед навчанням, оскільки він не лише зменшує розмір моделі та пришвидшує її виконання, але й дозволяє швидше навчати моделі архітектури візуального трансформера.

РОЗДІЛ 3. ОПТИМІЗАЦІЯ МОДЕЛЕЙ АРХІТЕКТУРИ ВІЗУАЛЬНИХ ТРАНСФОРМЕРІВ ДЛЯ ВІДЕО

3.1. Методи оптимізації моделей трансформерів для відео

Постійне зростання кількості створюваних відео ставить перед розробниками завдання розробки швидких та ефективних моделей. Моделі архітектури трансформерів демонструють вищу точність порівняно зі своїми попередниками у задачах, пов'язаних із комп'ютерним зором та обробкою послідовностей. Застосування моделей на основі архітектури трансформерів потребує високих обчислювальних ресурсів, що обмежує їх використання в задачах з обмеженими обчислювальними ресурсами або у випадках, коли потрібна висока швидкість виконання. Також навчання таких моделей займає багато часу через їхню складність. Ці виклики стають критичними при розв'язанні задачі розбиття відео на сцени за допомогою архітектури візуального трансформера для відео та потребують покращення існуючих методів оптимізації з метою підвищення ефективності, зниження витрат на навчання та розширення можливостей для обробки масштабних задач.

3.1.1. Скорочення складності моделей архітектури трансформерів

Одним із напрямків оптимізації трансформерів є зменшення обчислювальної складності з метою подальшого їх використання на платформах з обмеженими обчислювальними ресурсами та підвищення швидкості їх виконання. Основна складність трансформерів полягає у застосуванні механізму самоуваги, який має складність $O(n^2)$, де n — це кількість елементів у послідовності. У результаті цього, при використанні моделей архітектури візуальних трансформерів, зі збільшенням довжини послідовності або роздільної здатності відео, обчислювальна складність

стрімко зростає через необхідність обробки великої кількості просторово-часових патчів [23, 111].

З метою зменшення складності моделей трансформерів було розроблено кілька підходів. Одним із основних підходів є заміна механізму самоуваги на спрощену версію, у якій увага обчислюється лише для частини елементів у послідовності. Одним із механізмів досягнення спрощеного механізму самоуваги є застосування механізму розрідженої уваги, який дозволяє моделі обробляти лише важливі залежності між патчами, виключаючи неважливі. Іншим механізмом спрощення уваги є метод порогового прунінгу, який видаляє компоненти, що мають незначний вплив на кінцеві розрахунки моделі, тим самим зменшуючи обчислювальну складність [88, 112].

Іншим підходом до оптимізації візуальних трансформерів для відео є застосування факторизації уваги. У цьому випадку глобальне обчислення матриці уваги для всіх патчів замінюється на два окремі механізми самоуваги. Перший механізм самоуваги аналізує просторову компоненту для кожного зображення, після чого результати передаються до другого механізму уваги, який аналізує часову компоненту. Застосування цього підходу дозволяє скоротити складність моделі до $O(n \cdot m)$, де n — це кількість елементів у послідовності, а m — кількість патчів для кожного зображення. Факторизація уваги у моделях архітектури візуального трансформера для відео значно зменшує обчислювальні витрати при збереженні точності [23].

З метою зменшення обчислювальних витрат також застосовують лінійні трансформери. Особливістю лінійних трансформерів є використання низькорівневих апроксимацій для обчислення матриць уваги. Це дозволяє спростити складність моделі до $O(n)$, що робить можливим її використання для розв'язання задач у реальному часі, таких як відеоспостереження чи аналіз потокових даних [111].

Результати досліджень показали, що методи зменшення обчислювальної складності моделей є ефективними та дозволяють знизити обчислювальну

складність механізму самоуваги до $O(n)$. На основі цих результатів було вирішено детальніше дослідити методи оптимізації за допомогою прунінгу.

3.1.2. Прунінг та скорочення параметрів

Застосування методу прунінгу є широко розповсюдженою стратегією оптимізації нейронних мереж. Дослідження методів усунення надлишкових параметрів бере свій початок у 1990-х роках із робіт Optimal Brain Damage та Optimal Brain Surgeon, які запровадили концепцію використання інформації про другі похідні для оцінки впливу окремих вагових коефіцієнтів. Початковий підхід використовував апроксимацію матриці Гессе, проте його застосування вимагало значних обчислювальних витрат. Однак ці дослідження довели, що видалення малозначущих параметрів системи має мінімальний вплив на загальну точність моделі, але при цьому може значно підвищити її швидкодію [113, 114].

З розвитком глибоких нейронних мереж техніки прунінгу також удосконалювалися з метою адаптації до складніших та масштабніших архітектур. Дослідження показали, що малі ваги, як правило, мають низький вплив на результат роботи моделі, що призвело до широкого застосування методів прунінгу на основі магнітуди. Ці підходи дозволяли уникнути високої обчислювальної складності при роботі з матрицями Гессе завдяки застосуванню більш евристичної оцінки важливості параметрів. Також було запропоновано використання ітеративного прунінгу, під час якого проріджування параметрів моделі виконувалося кілька разів у процесі навчання. Таким чином, після кожної ітерації прунінгу модель донавчалася, що дозволяло ще більше зменшувати її розмір при збереженні точності. Важливим внеском у розвиток прунінгу стало дослідження "Гіпотеза лотерейного квитка". Це дослідження стверджує, що в глибоких моделях з високою щільністю існують підмережі, які здатні досягати аналогічної

продуктивності при повторному навчанні за умови відповідної ініціалізації початкових параметрів [115, 116].

Також було досліджено підходи до прунінгу, за яких видалення параметрів відбувається структуровано. У цьому випадку замість окремих зв'язків можуть видалятися цілі нейрони, фільтри або канали. Перевагою застосування структурованого прунінгу є не лише зменшення кількості параметрів, але й ефективне скорочення використання пам'яті та пришвидшення роботи моделі на апаратних засобах, оптимізованих для роботи з густими матрицями [117]. Порівнюючи структурований і неструктурований прунінг, важливо враховувати низку ключових факторів. Перш за все, варто звернути увагу на те, що хоча неструктурований прунінг не змінює архітектуру, а лише обнуляє певні параметри, це не завжди призводить до пришвидшення моделі. Це зумовлено необхідністю застосування спеціальних алгоритмів для обчислень розріджених матриць, проте такі обчислення зазвичай недостатньо оптимізовані в стандартному апаратному забезпеченні. Це може призвести не лише до відсутності пришвидшення моделі, а й до її сповільнення. На противагу цьому, структурований прунінг змінює топологію моделі, що дозволяє використовувати оптимізовану модель без зайвих обчислень, а також робить її ефективною на сучасних графічних процесорах та спеціалізованих апаратних прискорювачах, оскільки після структурованого прунінгу модель залишається компактною. Іншим важливим фактором є можливість мінімізувати або повністю уникнути донавчання моделі після неструктурованого прунінгу, тоді як після структурованого прунінгу донавчання є необхідним для відновлення точності моделі [118]. Також можна виділити напівструктурований прунінг, який дозволяє знайти компроміс між структурованим та неструктурованим прунінгом, як це продемонстровано на рисунку 3.1.

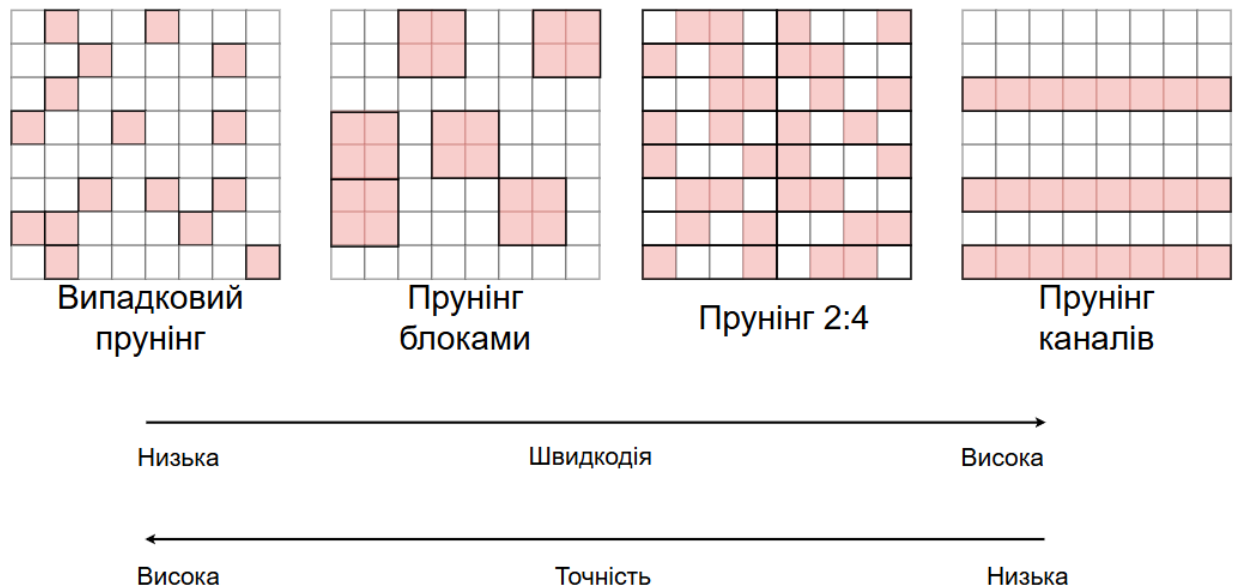


Рисунок 3.1 - Порівняння різних шаблонів прунінгу

Також було розроблено адаптивні методи прунінгу, такі як Dynamic Network Surgery (DNS). Цей метод прунінгу застосовується безпосередньо під час навчання, аналізуючи важливість вагових коефіцієнтів у процесі тренування. Далі, на основі порогових значень, визначаються ваги, які потрібно видалити. Проте, на відміну від класичних підходів до прунінгу, метод DNS пропонує механізм відновлення ваг на основі аналізу градієнтів. Під час тренування метод DNS аналізує градієнти для кожної ваги нейронної мережі, і якщо обнулена вага отримує високий градієнт, вона повертається до моделі. Завдяки цьому метод DNS дозволяє адаптивно змінювати структуру нейронної мережі, мінімізуючи втрати точності [119].

Іншим важливим досягненням стало розроблення методів прунінгу на основі чутливості, таких як Single-shot Network Pruning based on Connection Sensitivity (SNIP). Головною особливістю методу SNIP є його використання перед навчанням, на відміну від інших методів прунінгу, які інтегруються в процес навчання або застосовуються після нього. Алгоритм роботи SNIP полягає в тому, що береться модель нейронної мережі, ініціалізована випадковими вагами, а також вибирається невелика частина датасету, після

чого виконується прохід через нейронну мережу. На основі отриманого результату визначається вплив кожної ваги на функцію втрат. Після оцінки важливості ваг виконується прунінг, а параметри, що залишилися, тренуються стандартним підходом. Хоча при застосуванні цього підходу модель може мати обмежений простір для адаптації, а видалення великої кількості ваг може призвести до зниження точності, цей метод усуває потребу в додатковому навчанні, дозволяє контролювати втрату точності та значно пришвидшує не лише швидкодію моделі, а й сам процес навчання [120].

Аналізуючи існуючі підходи до прунінгу, можна чітко простежити тенденцію, яка вказує на необхідність пошуку компромісу між точністю моделей та їх швидкістю. При цьому дослідження методів прунінгу глибоких нейронних мереж дало поштовх до вивчення роботи розріджених мереж, динаміки навчання нейронних мереж та їх узагальнювальної здатності.

3.1.3. Прунінг архітектур трансформер

Застосування архітектур трансформерів дозволяє значно підвищити точність розпізнавання у задачах, які потребують визначення просторових чи часових залежностей. Проте використання таких моделей вимагає значних обчислювальних ресурсів та пам'яті. Ці обмеження зробили підходи до компресії моделей надзвичайно важливими для підготовки моделей до використання на пристроях з обмеженими ресурсами, а також у задачах, що потребують високої швидкості виконання.

Проте, на відміну від звичайних глибоких нейронних мереж, архітектури трансформерів мають унікальні виклики, пов'язані з оптимізацією механізмів багатоголової самоуваги та позиційно-залежних згорткових шарів. Дослідження механізму уваги показали, що збільшення кількості голів уваги дає мінімальний внесок у загальну точність моделі. При цьому було встановлено, що значну частину голів уваги можна видаляти, зберігаючи

високу точність моделі, що свідчить про надмірну кількість параметрів у шарах уваги [121]. Інше дослідження представило емпіричні докази ефективності скорочення кількості голів уваги у моделях трансформерів зі збереженням високої точності, що підтвердило ефективність використання методів прунінгу для голів уваги в трансформерних моделях [122].

Останні дослідження у сфері прунінгу запропонували, окрім прунінгу голів уваги, також застосовувати динамічні та структуровані методи прунінгу. Одним із методів динамічного прунінгу став Movement Pruning, у якому запропоновано використання траєкторії оновлення ваг моделі під час донавчання для ідентифікації параметрів, що роблять найменший внесок у результати моделі. У цьому підході відбувається адаптація схеми розрідженості матриць під час навчання, що дозволяє моделі ефективно зберігати здатність до точного представлення інформації. Цей підхід було успішно протестовано на попередньо натренованих моделях, таких як BERT, і в результаті його застосування для прунінгу вдалося зменшити кількість параметрів більш ніж на 50% при збереженні точності моделі відповідно до метрик, таких як GLUE [123]. Також важливо зазначити, що емпіричний аналіз впливу методів прунінгу на моделі трансформерів показав не лише зменшення розміру моделі, а й зміну її внутрішніх представлень про дані. Це пояснюється тим, що після донавчання решта голів уваги починають перерозподіляти свої функції, як показано на рисунку 3.2.

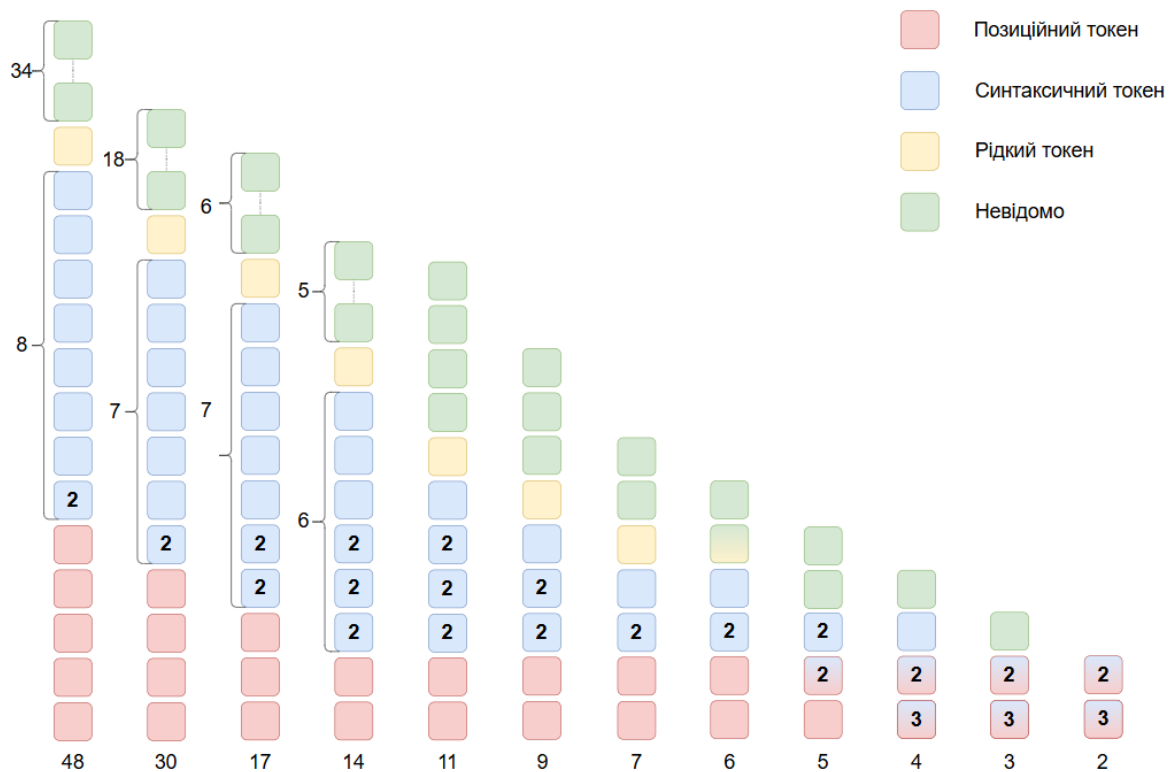


Рисунок 3.2 - Перерозподілення функцій голів уваги при застосуванні різної сили прунінгу

Аналізуючи таку реорганізацію даних, стає очевидним її вплив на здатність моделі до інтерпретованості та стійкості. На основі цього можна зробити висновок, що зі збільшенням кількості параметрів, які видаляються під час прунінгу, швидкість роботи моделі зростає, а використання пам'яті значно зменшується. Проте виникає потреба в детальному аналізі продуктивності та надійності моделі під час розв'язання різних задач.

Також було проведено дослідження прунінгу на рівні шарів для моделей трансформерів. У цьому випадку структурований прунінг дає можливість видаляти цілі компоненти, такі як голови уваги чи шари, що, порівняно з неструктурованими методами прунінгу, дозволяє формувати регулярні шаблони розрідженості. Це, своєю чергою, дає змогу таким моделям після прунінгу значно ефективніше використовувати апаратне прискорення. Важливо враховувати, що, оскільки застосування структурованих підходів

тісно пов'язане з апаратною архітектурою, оптимальний результат можна досягти за умови узгодження схеми розрідженості матриць з обчислювальними можливостями цільової платформи для використання моделі. У результаті аналізу було встановлено, що застосування прунінгу на рівні шарів до архітектури трансформерів потребує особливої уваги. Це зумовлено ієрархічною структурою трансформерів, у якій різні шари кодують різні рівні синтаксичної та семантичної інформації. Емпіричні дослідження показали, що видалення середніх шарів моделі дає змогу підвищити продуктивність за незначної втрати точності, тоді як видалення шарів на початку або в кінці моделі призводить до значної втрати точності, що може стати критичним для використання цього підходу. Ця особливість підкреслює важливість розробки правильної стратегії прунінгу з урахуванням структури моделей архітектури трансформерів.

3.1.4. Вимоги до програмного забезпечення та обчислювальних ресурсів

Для розуміння обчислювальних вимог моделей архітектури трансформер необхідно детальне вивчення основних арифметичних операцій, апаратного забезпечення, яке їх виконує, та існуючих стратегій, що активно використовуються для зменшення обчислювального навантаження на систему.

Основною одиницею обчислень у чисельних алгоритмах є операції з рухомою комою (FLOP). При застосуванні цієї одиниці до нейронних мереж, зокрема трансформерів, кількість FLOP відображає загальну кількість арифметичних операцій, які виконує модель під час прямого проходження даних через неї. Основними операціями для нейронних мереж є додавання та множення матриць. Переважаючими є операції множення матриць, які застосовуються в механізмі самоуваги та повнозв'язних шарах і домінують за

кількістю FLOP. Розрахунок FLOP дозволяє виконувати апаратну діагностику для оцінки теоретичного навантаження під час роботи моделі.

Підрахунок FLOP для моделей архітектури трансформерів має дуже важливе значення, оскільки дозволяє дослідникам оцінити потенційні вимоги до обчислювальних ресурсів та поведінку моделі під час масштабування відповідно до апаратного забезпечення. Ця метрика дає змогу уніфіковано порівнювати моделі та допомагає приймати рішення як під час проєктування системи та моделі, так і в процесі її оптимізації. Проте важливо зауважити, що хоча вимірювання FLOP забезпечує абстрактне розуміння складності моделі, воно не враховує нюансів програмного забезпечення.

Розглядаючи сучасні моделі трансформерів, такі як BERT, GPT та Transformer-XL, можна побачити, що їх широке використання стало можливим не лише завдяки високій точності, яка значно перевершує точність інших архітектур, але й завдяки їх здатності ефективно використовувати паралельні обчислення. Оскільки архітектура трансформерів базується на використанні багатоголової самоуваги та позиційно-залежних шарів, це формує обчислювальні схеми, які складаються з щільних множень матриць [124-126]. Використання сучасних центральних та графічних процесорів, які мають спеціалізовані апаратні прискорювачі, такі як NVIDIA Tensor Cores та векторизовані інструкції AVX-512, дає змогу застосовувати надзвичайно оптимізовані операції для множення щільних матриць. Це дозволяє ефективно працювати з алгоритмічною структурою моделей архітектури трансформерів, використовуючи можливості сучасних апаратних прискорювачів [127, 128].

Моделі архітектури трансформер мають вбудований високий рівень паралелізму, що дозволяє ефективно оброблювати цілі послідовності одночасно, уникаючи вузьких місць у системі, які можуть значно уповільнювати розрахунки. На відміну від них, рекурентні нейронні мережі мають обмежений рівень паралелізму, що робить їх менш ефективними для великомасштабних обчислень. Ця перевага стає ще більш вираженою при

використанні оптимізованих бібліотек, які пришвидшують операції лінійної алгебри, таких як cuBLAS для графічного процесора та Math Kernel Library (MKL) для центрального процесора. Також застосування сучасних апаратних прискорювачів дозволяє ефективно виконувати великомасштабні операції з плаваючою комою, що є важливим фактором при роботі з моделями архітектури трансформерів, які містять велику кількість параметрів. Таким чином, висока продуктивність моделей архітектури трансформерів досягається не лише завдяки алгоритмічному дизайну, а й завдяки розвитку апаратного забезпечення, яке адаптується до викликів, пов'язаних із розрахунками математичних операцій для щільних матриць [129,130].

Важливим фактором при аналізі ефективності виконання певної кількості FLOP є врахування типу процесора, на якому проходитимуть розрахунки. Центральний процесор (CPU) має невелику кількість потужних ядер, які оптимізовані для роботи з різними типами даних та характеризуються низькою затримкою при виконанні операцій. Особливістю центральних процесорів є наявність складної логіки керування та висока тактова частота, завдяки чому вони можуть швидко виконувати задачі різних типів. Спеціальні інструкції для роботи з векторними наборами даних, такі як AVX-512, дають змогу процесорам паралельно обробляти дані розміром до 512 біт, проте загальна кількість ядер для обробки інформації залишається обмеженою. Унаслідок цього, незважаючи на можливість центральних процесорів демонструвати високу швидкість FLOP на окреме ядро за оптимальних умов, їхній рівень паралелізму обмежений через архітектурні особливості CPU.

На противагу центральним процесорам, графічні процесори складаються з тисяч простіших ядер, що робить їх більш ефективними при застосуванні масового паралелізму. Особливої ефективності графічні процесори досягають під час розв'язання задач, що потребують обробки однорідних, але водночас інтенсивних обчислень, що є характерним для моделей архітектури трансформерів. Також графічні процесори можуть

містити спеціалізовані блоки, такі як NVIDIA Tensor Cores, які, в свою чергу, розширюють їхні можливості та дозволяють виконувати математичні операції змішаної точності при дуже високій пропускній здатності. Виходячи з особливостей центральних та графічних процесорів, навіть за однакової кількості FLOP графічні процесори зазвичай демонструють значно більшу ефективність, генеруючи більше результатів за секунду порівняно з центральними процесорами. Таким чином, можна зробити висновок, що хоча метрика FLOP є незмінним показником навантаження на систему, продуктивність моделі значною мірою залежить від можливостей апаратного забезпечення, зокрема від рівня паралелізму та ефективності арифметичних обчислень.

3.2. Алгоритм прунінгу

Використання апаратного забезпечення з високим рівнем паралелізму, такого як графічні процесори, дозволяє значно підвищити ефективність роботи моделей архітектури трансформерів. Проте розмір таких моделей все одно вимагає надзвичайно великої кількості FLOP, яка зазвичай вимірюється в мільярдах. Унаслідок цього використання таких моделей призводить до значних обчислювальних та енергетичних витрат. Однією з ефективних стратегій для суттєвого зменшення обчислювальних та енергетичних вимог є використання методів прунінгу, які дають змогу видаляти параметри моделі, що найменше впливають на кінцевий результат, мінімізуючи втрату точності.

Принцип роботи методу прунінгу полягає у виявленні та подальшому видаленні ваг, нейронів чи цілих голів уваги, які є найменш важливими. Після застосування методів прунінгу кількість FLOP моделі зменшується, оскільки для розрахунку результатів під час прямого проходу зменшується кількість арифметичних операцій. Внаслідок зменшення кількості FLOP також скорочується час виконання моделі та енергоспоживання, що особливо

важливо для задач, які потребують високої швидкодії, або в середовищах із обмеженими ресурсами, таких як периферійні пристрої.

Проте важливо зазначити, що архітектура моделей трансформерів переважно складається з множення щільних матриць у багатоголових шарах самоуваги та повнозв'язних шарах. Тому після застосування методів прунінгу до моделей архітектури трансформерів, попри теоретичне зменшення кількості FLOP, практична зміна швидкодії моделі залежатиме від типу розрідженості, яка утворюється після застосування прунінгу. При застосуванні неструктурованого прунінгу модель стає нерегулярною, унаслідок чого, незважаючи на зменшення кількості FLOP, ефективність її роботи на стандартних графічних та центральних процесорах залишається незмінною через відсутність підтримки операцій для роботи з розрідженими матрицями. При використанні методів структурованого прунінгу блоки формують регулярні шаблони, що, своєю чергою, дає змогу краще адаптувати такі моделі для використання ефективних операцій з цільними матрицями та їх аналізу за допомогою спеціалізованих розріджених ядер.

Можливим підходом до оптимізації роботи з обрізаними ваговими матрицями є зберігання лише ненульових елементів та їхніх індексів у форматах CSR або CSC, що дає змогу зменшити вимоги до пропускної здатності пам'яті. Також у випадках, коли структура розрідженості є регулярною, з моделі видаляються цілі стовпці з матриць значень, ключів чи запитів або цілі голови уваги, що дає можливість обробляти решту цільних підматриць за допомогою високооптимізованих методів роботи з матрицями. Дослідження розріджених моделей трансформерів, які утворюються внаслідок застосування структурованого прунінгу, що дає змогу видаляти цілі голови уваги, демонструють здатність моделей архітектури трансформерів значно зменшувати обчислювальну складність, зберігаючи початкову точність, яка була до застосування методів прунінгу.

3.2.1. Формати зберігання розріджених матриць

Одним із найпростіших способів представлення розрідженої матриці є використання списку координат. Принцип зберігання даних у списку координат полягає у збереженні лише ненульових елементів та відповідних індексів рядків і стовпців у трьох окремих масивах. Під час переведення даних із розрідженої матриці до списку координат утворюються кортежі з трьох елементів, де перші два представляють індекси рядка та стовпця, у яких знаходиться значення, а третій елемент містить саме це значення. Цей формат зберігання даних добре справляється із задачами побудови матриці чи інкрементних оновлень, оскільки дозволяє ефективно вміщувати повторювані індекси і не потребує визначення структури заздалегідь. Проте при використанні списку координат доводиться виконувати неефективні арифметичні операції та операції зрізу, через що застосовують модифіковані варіанти цього формату.

Формат стиснення розріджених рядків (CSR) є покращеною версією списку координат, оскільки використовує стиснення індексів рядків. Основна відмінність цього методу полягає у збереженні масиву вказівників на початок та кінець даних для кожного рядка в масиві ненульових значень, замість збереження індексу рядка для кожного ненульового елемента. Окрім збереження ненульових значень розрідженої матриці та відповідних індексів стовпців, матриця CSR містить додатковий масив вказівників для рядків, розмір якого на один більший за кількість рядків, як продемонстровано на рисунку 3.3. Застосування цього підходу дозволяє ефективно виділяти рядки та швидко виконувати множення матриці на вектор завдяки швидкому доступу до всіх ненульових елементів у кожному рядку. Формат зберігання даних CSR широко застосовується в бібліотеках, які використовуються для прискорення операцій, що обробляють дані по рядках, завдяки використанню оптимізованих операцій лінійної алгебри [131].



Рисунок 3.3 - Зберігання даних у форматі CSR.

Альтернативним методом стиснення даних є формат compressed sparse column (CSC), який, на відміну від CSR, стискає індекси стовпців. При зберіганні даних у форматі CSC ненульові елементи зберігаються по стовпцях, а також створюється масив вказівників на стовпці, що позначає початок даних у кожному стовпці для масиву значень. Застосування формату CSC є ефективним для використання в алгоритмах, які вимагають розбиття даних на стовпці, а також в операціях, де доступ до даних відбувається у порядку збільшення стовпців. Важливо зауважити, що велика кількість програмних засобів, зокрема вбудовані функції MATLAB для роботи з розрідженими матрицями, використовують формат CSC для виконання ефективних операцій на основі стовпців [117].

3.2.2. Напів-структурована розрідженість формату 2:4

Під час застосування методів прунінгу для оптимізації нейронних мереж найбільш ефективним є напівструктурований метод прунінгу 2:4. Особливістю методу прунінгу 2:4 є те, що у кожному суміжному блоці з чотирьох вагових компонентів два компоненти обнуляються, тоді як два інші залишаються без змін. Унаслідок застосування методу прунінгу 2:4 матриці набувають фіксованої, регулярної розрідженості, що дає змогу ефективно скорочувати кількість обчислень, водночас зберігаючи репрезентативні властивості моделі, що є критично важливим для великих нейронних мереж, зокрема моделей архітектури трансформерів.

Метод прунінгу 2:4 ділить матриці вагів на групи по чотири послідовних елементи, після чого для кожної групи окремо застосовується прунінг на основі магнітуди. Внаслідок цього зберігаються лише два вагові коефіцієнти, які мають найбільший вплив на систему, тоді як два інші вагові коефіцієнти обнуляються. Такий контрольований прунінг забезпечує фіксовану розрідженість матриці на рівні 50% та створює передбачувану і регулярну структуру матриці в моделі. З метою оптимізації обчислень після застосування прунінгу 2:4 використовується таблиця індексів, яка містить інформацію про позиції збережених ваг для кожного блоку, як показано на рисунку 3.4. Таблиця індексів дозволяє за допомогою виконання матричних операцій швидко зчитувати лише ті ваги, що залишилися після прунінгу, ігноруючи обнулені. При цьому сучасні спеціалізовані обчислювальні блоки, такі як Tensor Cores, дозволяють використовувати таблицю індексів для ефективного виконання розріджених обчислень. За рахунок цього таблиця індексів дає змогу при використанні методу прунінгу 2:4 виконувати швидкі обчислення у розріджених тензорах та ефективно застосовувати програмні пришвидшувачі [132].

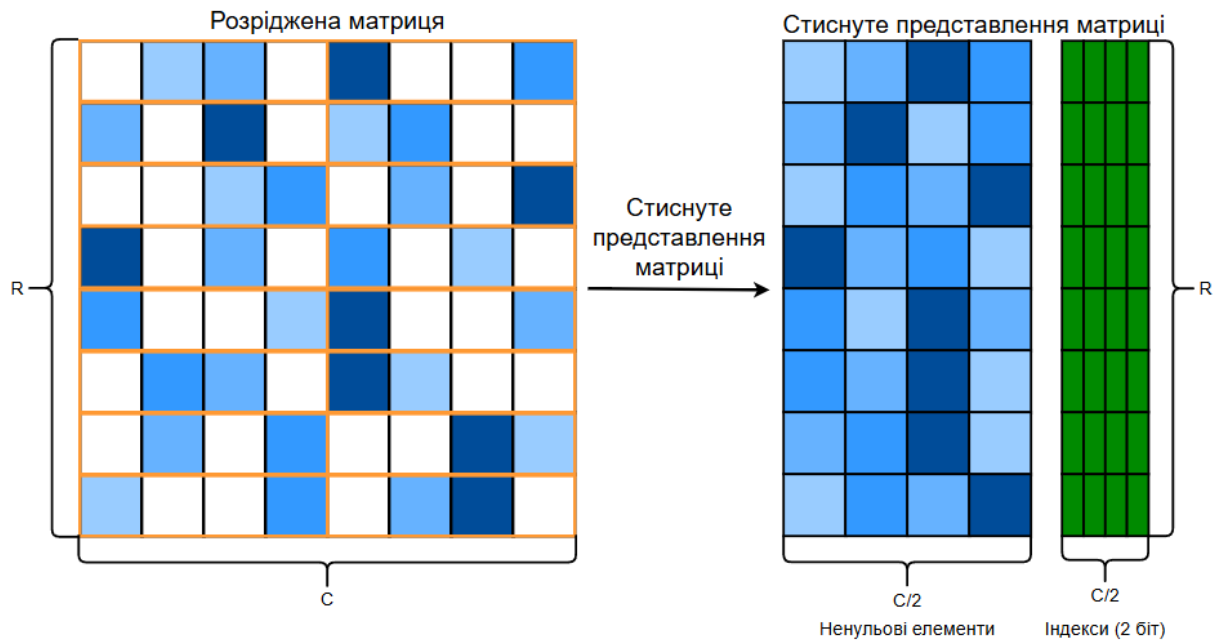


Рисунок 3.4 - Збереження вагів при застосуванні методу прунінгу 2:4

Сучасне апаратне забезпечення дозволяє ефективно використовувати розрідженість матриць завдяки спеціальним ядрам для множення розріджених матриць, оптимізованим для обробки розріджених структур за шаблоном 2:4. Теоретично це дає змогу отримати прискорення вдвічі порівняно з обчисленням повних матриць, проте на практиці прискорення є дещо меншим через витрати часу на завантаження метаданих і запуск спеціалізованих ядер.

Хоча неструктурований прунінг, який видаляє ваги довільно по всій матриці, дозволяє досягати високих рівнів компресії, він створює нерегулярні схеми доступу до пам'яті, що не забезпечує ефективного прискорення на спеціалізованому апаратному забезпеченні з високим рівнем паралелізації, такому як графічні процесори. Завдяки чітко визначеній структурі шаблону 2:4 з'являється можливість розробляти спеціалізовані ядра для роботи з розрідженими матрицями.

Спеціалізовані ядра для роботи з розрідженими матрицями дозволяють зберігати вагову матрицю у компактному вигляді, реєструючи лише ненульові значення та набір метаданих про їхні позиції для кожної групи, а також

виконувати обчислення лише на ненульових елементах матриці. Застосування такого структурованого підходу не лише спрощує апаратну реалізацію, а й дає змогу зберігати високу точність моделі. Емпіричні дослідження, проведені компанією NVIDIA та в межах проєкту Sparse-Llama, продемонстрували майже повне відновлення точності після використання методу прунінгу 2:4 [133].

Оскільки основні елементи моделей архітектури трансформерів, такі як багатоголова самоувага та повнозв'язні шари, містять щільні матриці, застосування методу прунінгу 2:4 для їхньої оптимізації є надзвичайно ефективним. Проте, оскільки застосування методу прунінгу змінює репрезентативні властивості моделей, це може призвести до зниження точності. Тому після використання методу прунінгу 2:4 важливо проводити донавчання моделей архітектури трансформерів для компенсації можливої втрати точності. Дослідження показують, що застосування методу прунінгу 2:4 з подальшим донавчанням моделі дозволяє досягти майже ідентичної точності порівняно з моделями до застосування методу прунінгу, водночас значно зменшуючи використання пам'яті та обчислювальних витрат.

Після застосування методу прунінгу 2:4 вагові матриці зберігаються у стисненому форматі, що містить лише ненульові значення та їхні позиції у кожній групі. Під час роботи з такими моделями матриці у стисненому форматі передаються до спеціалізованих ядер множення розріджених матриць, що використовуються в бібліотеках cuSPARSElt та NVIDIA CUTLASS, які пропускають нульові значення, завдяки чому значно пришвидшується множення матриць [134, 135].

При використанні сучасних бібліотек для навчання глибоких нейронних мереж, таких як PyTorch, можна застосовувати спеціальні функції для перетворення тензорів ваг у напівструктуровані розріджені представлення. Функції, такі як `to_sparse_semi_structured` у бібліотеці PyTorch, дають змогу перетворювати вагові матриці після прунінгу у формат, сумісний із

спеціалізованими ядрами графічних процесорів для роботи з розрідженими матрицями. Інтеграція таких методів значно спрощує використання моделей архітектури трансформерів після прунінгу методом 2:4 і зменшує інженерні витрати на їх розгортання.

3.2.3. Алгоритмічне забезпечення оцінки важливості вагів для напів-структурованої розрідженості

Принцип роботи методу прунінгу SNIP полягає у знаходженні спеціальної підмножини ваг W_{subset} , яка є частиною початкового набору ваг W і має мінімальне значення функції втрат $L(W_{\text{subset}})$ для цієї підмножини. У контексті роботи нейронної мережі W є множиною всіх ваг моделі, а $L(W)$ — функцією втрат, яку нейронна мережа оптимізує під час навчання.

Принцип роботи алгоритму складається з кількох кроків, першим з яких є ініціалізація ваг нейронної мережі випадковими значеннями та створення набору масок для вагових векторів, які приймають значення 1 або 0. Далі відбувається процес тренування моделі на частині тренувальних даних, що дозволяє визначити градієнти функції втрат для кожної маски m_i . Розраховані градієнти вказують на вплив незначних змін у масках на функцію втрат, що дає змогу оцінити важливість кожної ваги, використовуючи формулу:

$$I_i = \left| \frac{\partial L(m_i)}{\partial m_i} \right| \quad (3.1)$$

Важливість кожної ваги w_i , що входить до множини W , обчислюється як абсолютне значення часткової похідної функції втрат відносно відповідних масок m_i . Після отримання важливостей ваг вони нормалізуються з метою визначення їхньої відносної важливості за формулою:

$$\hat{I}_i = \frac{I_i}{\sum_j I_j} \quad (3.2)$$

Після цього нормалізовані показники важливості сортуються за зростанням, а найменш значущі ваги, що відповідають заданій частці ρ , обнуляються. Таким чином формується нова підмножина ваг \mathcal{W}_s , яка містить лише найбільш значущі параметри:

$$\mathcal{W}_s = \{w_i \in \mathcal{W} : \hat{s}_i \geq \rho\} \quad (3.3)$$

Кількість ваг, що видаляються, визначається параметром ρ , значення якого може змінюватися від 0, коли прунінг не виконується, до 1, коли видаляються всі можливі ваги. Таким чином, параметр ρ визначає відсоткове співвідношення ваг, які потрібно видалити, і може бути представлений наступною формулою:

$$p = \frac{N_{\text{pruned}}}{N_{\text{total}}} \quad (3.4)$$

Після застосування методу прунінгу модель навчають, використовуючи стандартні підходи. Однак, для забезпечення ефективності цього процесу важливо зберігати однакову дисперсію для всіх шарів. Для цього використовують ініціалізацію Ксав'є, яка забезпечує оптимальний розподіл ваг нейронної мережі та розраховується за такою формулою:

$$W \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right) \quad (3.5)$$

Запропонований алгоритм прунінгу є модифікацією методу прунінгу SNIP, яка враховує показники уваги моделей архітектури трансформерів під час оцінки важливості ваг.

На відміну від класичного застосування методу SNIP, який визначав розрідженість підмножини ваг W_s , де $W_s \subseteq W$, з метою мінімізації функції втрат, новий підхід включає активацію уваги та вектори виходу моделі, що формалізуються такою формулою:

$$L' = L(W) + \sum_j A_j + O \quad (3.6)$$

, де A_j позначає сумарні виходи для j -го шару уваги, а O представляє суму тензора виходів, що, своєю чергою, дає змогу розраховувати важливість ваг за розширеною формулою:

$$s_i = \left| \frac{\partial L'(c_i)}{\partial c_i} \right| \quad (3.7)$$

Застосування цієї модифікації дає змогу під час зворотного проходження збільшувати значення градієнтів відповідно до чутливості кожної окремої ваги у шарах уваги. Враховуючи важливість механізму уваги в моделях архітектури трансформерів, включення цих параметрів до критеріїв прунінгу дозволяє значно підвищити ефективність запропонованого методу.

Особливістю оригінального методу прунінгу SNIP є створення нерегулярних матриць під час обрізання, що зумовлено неструктурованим підходом цього методу. Оскільки в результаті неструктурованого прунінгу виникає нерегулярна розрідженість матриць, швидкість обробки таких моделей може не збільшуватися або зростати мінімально навіть за умови використання сучасних апаратних пришвидшувачів, таких як графічні

процесори. Це пов'язано з тим, що графічні процесори мають високий рівень паралелізації та розроблені для роботи з регулярними шаблонами обчислень.

Тому для вирішення цієї проблеми було вирішено додати до запропонованого методу прунінгу інтеграцію напівструктурованої схеми розрідженості 2:4. У цій схемі вагова матриця розбивається на суміжні групи по чотири елементи, після чого два елементи, що мають найменший вплив на систему, обнуляються, як показано на рисунку 3.5. Однак це накладає обмеження на фіксовану розрідженість матриць у розмірі 50%. Водночас такий підхід дає змогу проводити ефективні обчислення на спеціалізованому апаратному забезпеченні, зокрема на розріджених ядрах у графічних процесорах NVIDIA Ampere.

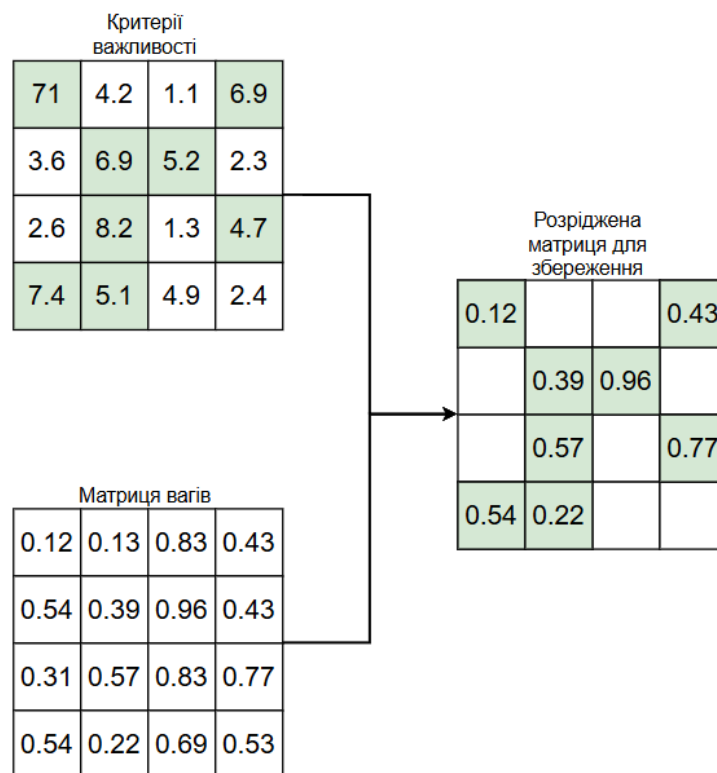


Рисунок 3.5 - Застосування прунінгу 2:4

Поєднання модифікованого методу SNIP із застосуванням розрідженості матриць 2:4 дає змогу ефективно оптимізувати моделі архітектури трансформерів, водночас зберігаючи напівструктурованість матриць. Це,

своєю чергою, дозволяє ефективно використовувати апаратні пришвидшувачі для досягнення значного прискорення не лише під час застосування моделі для вирішення прикладних задач, але й під час її навчання без втрати точності.

3.2.4. Експериментальна оцінка удосконаленого методу оптимізації

З метою перевірки ефективності розробленого методу прунінгу було вирішено оптимізувати модель архітектури візуальних трансформерів для відео, оскільки вона ефективно аналізує просторові та часові залежності й вирішує задачу визначення зміни сцен. Моделі архітектури візуальних трансформерів для відео потребують високих обчислювальних ресурсів як під час виконання, так і в процесі навчання. Тому застосування розробленого методу прунінгу для їх оптимізації має забезпечити значне підвищення ефективності таких моделей.

Під час навчання моделі архітектури візуального трансформера для відео з метою вирішення задачі розпізнавання зміни сцен було вирішено використовувати датасети Raid, OsVsd та BBC Planet Earth. Ці датасети було розділено на тренувальну та валідаційні вибірки, які містили 7 856 сцен загальною тривалістю 1 047 хвилин для тренування моделі та 4 844 сцени загальною тривалістю 539 хвилин для тестування результатів навчання розробленої моделі архітектури візуального трансформера для відео. З метою оптимізації навчання та підвищення точності моделі було вирішено покращити датасет шляхом додаткового визначення планів для кожного відео та створення спеціальної розмітки, яка містила інформацію про сцени, плани, що входять до кожної сцени, а також про сусідні сцени. Завдяки цій модифікації датасету стало можливим використання лише ключових кадрів для кожного плану та зміна порядку сцен і планів усередині сцени з метою штучного розширення датасету при збереженні оригінального контексту сцени. Також це дозволило під час розробки архітектури візуального

трансформера для відео значно зменшити довжину послідовності, яка надходила на вхід моделі, завдяки використанню лише ключових кадрів для кожного плану. Після навчання моделі архітектури візуального трансформера для відео було вирішено застосувати розроблений метод прунінгу з метою порівняння змін у точності та швидкодії моделі до та після його застосування.

В результаті застосування розробленого методу прунінгу було встановлено, що розмір моделі зменшився на 43%, швидкодія моделі зросла на 10%, а її точність покращилася на 0,4%, що продемонстровано в таблиці 3.1.

Таблиця 3.1. Порівняння результатів виявлення зміни сцен

Модель	F1
Розроблена модель	0.721
Розроблена модель після застосування методу прунінгу	0.725
Deep Multimodal Networks [3]	0.67
STG[4]	0.41
NW[5]	0.33

Однак важливо зауважити, що для тестування швидкодії використовувалася відеокарта NVIDIA RTX 3070, яка, хоча й має Tensor Cores, не підтримує роботу з розрідженими матрицями формату 2:4, що утворюються при застосуванні розробленого методу прунінгу. Зважаючи на особливості доступного для тестування апаратного забезпечення, не вдалося досягти максимального рівня пришвидшення. Проте при використанні моделі архітектури візуального трансформера після застосування розробленого методу прунінгу на спеціалізованому апаратному забезпеченні, такому як графічні процесори від NVIDIA з підтримкою обробки розріджених матриць формату 2:4, таких як A100, A30 чи H100, можливо досягти значного підвищення швидкодії моделі.

Отже, дослідження показало, що застосування розробленого методу прунінгу дозволяє значно зменшити розмір моделі, підвищити швидкість її виконання, і при цьому зберігається оригінальна точність. Це робить застосування розробленого методу ефективним у задачах оптимізації нейронних мереж на основі архітектури трансформерів, особливо при подальшому використанні оптимізованих моделей на спеціалізованому апаратному забезпеченні.

3.3. Висновки до розділу 3

В результаті проведених експериментів було встановлено, що розроблений метод прунінгу демонструє високу ефективність при оптимізації моделей архітектури візуальних трансформерів для відео завдяки значному зменшенню розрахункових витрат при збереженні точності моделі. При цьому було досліджено різні підходи до прунінгу глибоких нейронних мереж, що дозволили покращити найбільш важливі елементи прунінгу нейронних мереж на основі архітектури трансформерів з метою одночасного зменшення розміру моделі, підвищення швидкодії та збереження оригінальної точності. Першою частиною модифікованого методу прунінгу стало покращене визначення важливості ваг у методі SNIP за рахунок додавання розрахунку активацій уваги та векторів виходу моделі до загальної функції втрат. Другою частиною покращеного методу прунінгу стала інтеграція розрідженості матриць за шаблоном 2:4, що дозволяє працювати з напіврозрідженими матрицями.

В результаті експериментів було встановлено, що при використанні апаратного забезпечення без підтримки роботи з розрідженими матрицями модель демонструє пришвидшення приблизно на 10%. При цьому моделі на основі архітектури трансформерів мають потенціал значного збільшення швидкодії при використанні спеціалізованого апаратного забезпечення, такого як NVIDIA A100, A30 чи H100. Також важливим аспектом покращеного

методу прунінгу є збереження оригінальної точності моделі, що робить цей метод надзвичайно ефективним для задач, у яких втрата точності може бути критичною. Крім того, застосування цього підходу дозволяє не лише пришвидшити використання моделі, а й значно прискорити процес її навчання.

Проведене дослідження підтверджує ефективність запропонованого методу прунінгу при його застосуванні до моделей архітектури трансформерів. Метод забезпечує підвищення продуктивності, зменшення розміру моделі та збереження її оригінальної точності. Це дозволяє використовувати його для вирішення завдань, що потребують високої швидкодії, зокрема для аналізу відеоконтенту.

РОЗДІЛ 4. ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ВІДЕОАТРИБУТІВ

Для забезпечення виконання вимог до програмного забезпечення для автоматизованого визначення відеоатрибутів із використанням розбиття відео на сцени та плани, що були сформульовані у розділі 1, пропонується універсальна архітектура програмної системи, що зображена на рисунку 4.1.

Компоненти цієї архітектури можна розділити на чотири групи:

- модулі для обробки відео, взаємодії з розподіленими обчислювальними ресурсами, паралельного обчислення відеоатрибутів та керування потоком даних;
- компоненти для прунінгу перед тренуванням, які дозволяють оптимізувати нейронні мережі шляхом усунення малозначущих параметрів, що знижує обчислювальні витрати та підвищує швидкодію моделі без значної втрати точності;
- модулі для автоматизованого розбиття відео на плани та сцени, що використовують візуальні трансформери для точного виявлення змін у відеопотоках;
- система збереження та аналізу отриманих результатів, яка дозволяє зберігати та обробляти метрики, отримані в процесі визначення зміни планів, сцен та відеоатрибутів.

Основу архітектури становлять модулі для завантаження відео, підготовки до аналізу, визначення змін кадрів (планів) і керування розподіленими ресурсами для обробки відеоконтенту. Моделі, які використовуються для визначення змін сцен, планів та аналізу відеоатрибутів, зберігаються в окремому сховищі. Це забезпечує зручний доступ до них, а також можливість оновлення або додавання нових моделей для аналізу.

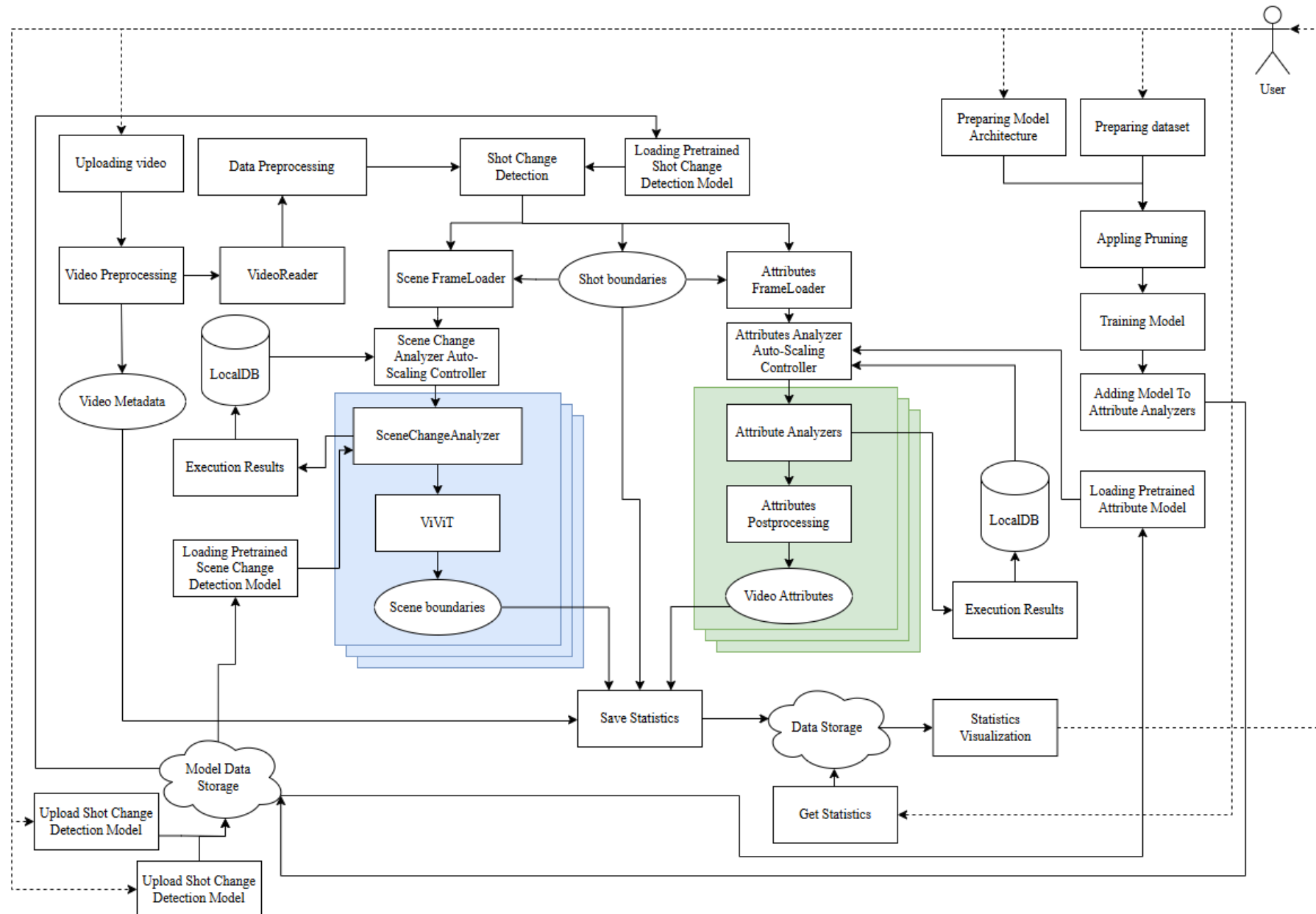


Рисунок 4.1 – Універсальна архітектура програмної системи для розподіленого визначення відеоатрибутів

Під час аналізу відео кадри трансформуються та передаються до модуля визначення змін планів, який детально описано в розділі 4.4. Коли цей модуль виявляє зміну плану, він обирає ключові кадри. При цьому є можливість додаткового налаштування моделі для збільшення кількості ключових кадрів у плані, що дозволяє отримати більше даних для аналізу. Далі ці кадри разом із інформацією про план передаються до модулів, що зберігають даних перед аналізом для подальшого аналізу та керують обчислювальними потоками для обробки відео, а саме потоками для визначення сцен і атрибутів.

Модулі для зберігання даних перед аналізом дозволяють ефективно обробляти інформацію та групувати її відповідно до вимог модулів аналізу. Для аналізу сцен розроблений модуль зберігання даних групує інформацію за групами та передає їх у вигляді наборів зображень, що необхідно для визначення плану. Це дає змогу ефективно аналізувати зміну планів на різних потоках, оскільки кожен потік отримує власну копію зображень, необхідних для аналізу. Для визначення атрибутів модуль зберігання даних перед аналізом дозволяє накопичувати інформацію та передавати її у вигляді наборів зображень, що забезпечує ефективне використання розподілених обчислень, закладених в архітектуру нейронних мереж. При цьому набір даних може бути налаштований залежно від характеристик апаратного забезпечення.

Модулі для обчислювальними потоками мають доступ до відповідних баз даних, у яких зберігається інформація про потоки та результати їхньої роботи. Завдяки цьому система може приймати рішення про зміну кількості активних потоків, а також визначати, які потоки доступні для подальшої обробки інформації. Після завершення аналізу результати передаються до модуля збереження та візуалізації статистики. Користувач має можливість взаємодіяти з цим модулем, отримуючи деталізовану інформацію про оброблене відео, яка включає в себе точні часові мітки зміни планів та сцен, а також атрибутів які були визначені в процесі аналізу, як для кожної сцени та плану, так і узагальнені для всього відео.

Під час визначення атрибутів користувач може використовувати власні методи аналізу. Водночас архітектура підтримує автоматичне навчання моделей із використанням розробленого методу прунінгу перед навчанням, що детально описано в розділі 4.3. Для використання цього модуля користувачеві потрібно передати тренувальний набір та архітектуру нейронної мережі. У результаті система навчить оптимізовану модель, яка буде збережена у сховищі моделей для подальшого використання.

Важливою особливістю розробленої архітектури є її здатність ефективно використовуватися в хмарному середовищі. Асинхронні задачі визначення сцен та атрибутів на відео можуть запускатися у вигляді контейнерів на віртуальних машинах, таких як AWS EC2, що дозволяє обирати оптимальні налаштування залежно від обсягу обчислень та їхньої складності. Крім того, це дає змогу використовувати спеціалізоване апаратне забезпечення, таке як NVIDIA A100, A30 чи H100, для ефективного виконання моделей, оптимізованих методом прунінгу у форматі 2:4. Для керування цими контейнерами можна застосовувати спеціалізовані хмарні технології, такі як Kubernetes. Зберігання даних можна реалізувати шляхом поєднання віртуальних машин чи асинхронних задач, таких як AWS Lambda та Azure Functions, із сховищами для зберігання даних, такими як Amazon DynamoDB.

Розроблена архітектура забезпечує ефективний аналіз відеоконтенту з можливістю гнучкого керування обчислювальними ресурсами та автоматично адаптується до різного рівня навантаження, що особливо важливо при роботі з великими обсягами даних. При цьому архітектура створена з урахуванням можливості інтеграції із сучасними хмарними середовищами, такими як AWS, Azure та Google Cloud. Розроблена архітектура є гнучкою, адаптивною та може використовуватися як для аналізу великих обсягів даних, так і для обробки відео в реальному часі.

4.1. Засоби розроблення для програмного забезпечення визначення відеоатрибутів

Середовищем розробки було обрано інтегроване середовище розробки (IDE) PyCharm, створене спеціально для розробки програмного забезпечення мовою програмування Python. PyCharm має вбудовану систему автоматичного доповнення коду, інструменти для швидкого та ефективного рефакторингу, а також засоби для налагодження коду. Крім того, середовище підтримує використання вбудованих інструментів для підвищення ефективності навчання та роботи з нейронними мережами, таких як Jupyter Notebook, NumPy та TensorFlow.

Для збереження даних було вирішено використовувати нереляційну базу даних MongoDB, розроблену для роботи з великими обсягами неструктурованих або напівструктурованих даних. На відміну від реляційних баз даних, MongoDB використовує документо-орієнтовану модель збереження, що дозволяє гнучко змінювати схему даних. Це робить MongoDB ефективним рішенням для аналітики, особливо у системах керування контентом. Завдяки цьому можна швидко змінювати необхідні атрибути для побудови відеоаналітики, впроваджувати нові рішення та тестувати підходи. Крім того, MongoDB має оптимізовані механізми індексації та агрегації, що забезпечує ефективний пошук і швидку обробку даних [136].

Для керування базою даних було вирішено використовувати DataGrip, оскільки цей інструмент дозволяє ефективно адмініструвати бази даних і підтримує велику кількість СУБД, включаючи MongoDB. DataGrip має вбудовану систему автодоповнення SQL-запитів, можливість візуалізації зв'язків між таблицями, інструменти для оптимізації SQL-запитів, а також вбудований редактор документів у форматі JSON, що є особливо корисним при роботі з MongoDB.

Для роботи з багатовимірними масивами було вирішено використовувати бібліотеку NumPy, яка пропонує широкий набір математичних функцій для роботи з матрицями. NumPy забезпечує високу швидкість виконання математичних операцій завдяки використанню оптимізованих бібліотек, написаних мовами програмування C та Fortran. Це робить цей інструмент надзвичайно корисним для аналізу даних, проведення наукових обчислень та підготовки даних для машинного навчання [137].

Для роботи з зображеннями була обрана бібліотека OpenCV завдяки великій кількості оптимізованих операцій для аналізу та трансформації зображень. OpenCV містить реалізації класичних математичних методів, таких як виділення ключових точок, розпізнавання об'єктів, побудова гістограм, геометричні трансформації та визначення країв, що робить цю бібліотеку надзвичайно корисною для задач, пов'язаних із комп'ютерним зором. Крім того, OpenCV забезпечує високу швидкість обчислень, оскільки реалізована переважно мовою програмування C++ та використовує оптимізовані бібліотеки для прискорення обчислень, такі як Intel IPP, OpenCL та CUDA [138].

Для створення та збереження нейронних мереж було вирішено використовувати бібліотеки PyTorch і TensorFlow, залежно від типу створюваної нейронної мережі.

4.2. Програмне забезпечення для реалізації та впровадження нейронних мереж

Для реалізації нейронних мереж було вирішено використовувати бібліотеку TensorFlow, яка дозволяє ефективно навчати глибокі нейронні мережі та містить інструменти для масштабування їх використання й підготовки до застосування у промислових системах. Зокрема, TensorFlow має

вбудовані інструменти, такі як TensorFlow Serving для ефективного розгортання моделей і TensorFlow Lite для оптимізації моделей для використання на мобільних пристроях. Бібліотека TensorFlow використовує статичні графи для обчислень, що дозволяє підвищити продуктивність та ефективно виконувати операції на різних типах апаратного забезпечення. Для роботи з даними TensorFlow пропонує пакет `tf.data`, який містить функціонал для ефективного завантаження та обробки даних, що значно спрощує навчання моделей при роботі з великими обсягами інформації. Також у TensorFlow є спеціальне розширення для створення повноцінних конвеєрів машинного навчання, які автоматизують процеси від збору та обробки даних до розгортання моделі [136, 139].

Проте під час дослідження можливостей моделей архітектури візуальних трансформерів для відео та оптимізації таких моделей було вирішено використати бібліотеку PyTorch, яка також є спеціалізованим інструментом для навчання глибоких нейронних мереж. Основною особливістю PyTorch є використання динамічного обчислювального графу, що забезпечує більшу гнучкість у розробці нейронних мереж, спрощує відлагодження моделей та дозволяє адаптивно змінювати архітектуру під час навчання. PyTorch також підтримує новітні технології для графічних процесорів, що робить його ефективним для роботи з великими нейронними мережами та обробки даних у реальному часі. Додатковою перевагою є наявність інструменту TorchScript, який дозволяє конвертувати моделі для ефективного розгортання на платформах з обмеженими ресурсами. Це значно розширює можливості використання PyTorch у мобільних і вбудованих системах, а також у хмарних обчисленнях [140].

Ключовим компонентом бібліотеки PyTorch є `torch.Tensor`, який є основною структурою для представлення та обробки даних. Цей об'єкт дозволяє ефективно працювати з багатовимірними масивами, виконуючи операції як на графічному (GPU), так і на центральному процесорі (CPU). Крім

того, `torch.Tensor` має вбудовані функції для автоматичного диференціювання, що робить його надзвичайно корисним для навчання глибоких нейронних мереж. Бібліотека `PyTorch` також містить систему автоматичного обчислення градієнтів – `torch.autograd`, яка дозволяє легко застосовувати алгоритми оптимізації, використовуючи правило ланцюгової похідної. Це забезпечує ефективний процес зворотного поширення помилки (`backpropagation`) та значно спрощує реалізацію навчання моделей. Для створення архітектури нейронних мереж використовується пакет `torch.nn` який містить в собі готову реалізацію різних шарів, таких як `torch.nn.Linear`, для роботи з повноз'язними шарами, `torch.nn.Conv2d`, для використання згорткових шарів, чи `torch.nn.RNN` для рекурентних шарів. Під час створення шарів для них автоматично створюються відповідні параметри (вагові коефіцієнти), які оновлюються під час оптимізації моделі. Крім того, `torch.nn` містить механізми для автоматичного додавання функцій активації, таких як `ReLU`, `Sigmoid` чи `Softmax`, що дозволяє легко інтегрувати нелінійність у структуру нейронних мереж.

Бібліотека `TensorFlow` має аналог `torch.Tensor`, який називається `tf.Tensor`. Він є основною структурою для представлення багатовимірних масивів і підтримує виконання операцій у розподіленому середовищі. Використання статичного графа обчислень у `TensorFlow` дозволяє оптимізувати частину розрахунків ще на етапі компіляції моделей, що значно покращує продуктивність під час роботи з великими нейронними мережами. Механізм автоматичного диференціювання, який дозволяє ефективно обчислювати градієнти під час навчання, у `TensorFlow` реалізований за допомогою класу `tf.GradientTape`. Він забезпечує автоматичне відстеження всіх операцій для подальшого обчислення градієнтів під час зворотного поширення помилки. Реалізація вже готових шарів для навчання знаходиться в пакеті `tf.keras.layers`, яка включає в себе такі шари як `tf.keras.layers.Dense`, `tf.keras.layers.Conv2D` та `tf.keras.layers.LSTM`. Шари можуть бути використані

для побудови моделі за допомогою класів `tf.keras.Sequential`, де вони додаються у цей клас у вигляді масиву, чи `tf.keras.Model`, який приймає як параметри входи і виходи моделі, а шари зв'язуються між собою функціональним стилем програмування.

Основним алгоритмом для навчання нейронних мереж називається зворотнім поширенням помилки, та базується на оновленні вагів моделі нейронної мережі на основі функції втрат. Для обчислення градієнтів функції втрати щодо кожної ваги у кожному шарі моделі застосовується правило ланцюгової похідної. Обчислення градієнтів складається з двох етапів, а саме прямого та зворотного поширення. Під час прямого поширення вхідні дані проходять через всі шари нейронної мережі, де кожен нейрон виконує лінійне перетворення за наступною формулою:

$$z = Wx + b \quad (4.1)$$

та нелінійне активаційне перетворення

$$a = \sigma(z) \quad (4.2)$$

де W є матрицею вагів, b — зміщення, проте важливо зауважити, що цей параметр можна не використовувати в деяких випадках, x це матриця вхідного сигналу, а $\sigma(z)$ — функція активації яка забезпечую нелінійність перетворень, наприклад функції ReLU чи sigmoid. В результаті роботи прямого поширення на шарах виходу утворюється результат аналізу вхідних даних за допомогою нейронної мережі, після чого отриманий результат порівнюється з очікуванням за допомогою функції втрати. Функція втрати залежить від задачі яку вирішує нейронна мережа, при цьому для деяких задач може підходити декілька функцій втрат. Найбільш розповсюдженою є функція втрати яка розраховує середньоквадратичну помилку:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.3)$$

Наступним етапом є застосування зворотнього поширення, де за допомогою градієнтного спуску обчислюються часткові похідні функції втрат для кожного параметру моделі, використовуючи ланцюгове правило, яке можна представити за допомогою формули:

$$\frac{dL}{dW} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{dW} \quad (4.4)$$

Часткові похідні в для кожної вари є силою та напрямком корекції, яку необхідно зробити з метою покращення точності роботи моделі. Проте важливо зазначити, що похідна вказує лише напрямок та силу змін, через що під час оновлення вагів застосовують коефіцієнт η , який представляє швидкість навчання, та дозволяє оптимізувати процес навчання. Таким чином оновлення кожного параметру, градієнтний спуск, відбувається за формулою

$$W^{(t+1)} = W^{(t)} - \eta \frac{dL}{dW} \quad (4.5)$$

Таким чином використання алгоритму зворотнього поширення дозволяє нейронним мережам поступово зменшувати помилку при аналізі вхідних даних завдяки інтервальному оновленню вагів нейронної мережі [141]. Бібліотеки PyTorch та TensorFlow мають стандартні реалізації зворотнього поширення помилки та мають спеціальні пакети з реалізаціями найбільш ефективних алгоритмів градієнтного спуску, таких як стохастичний градієнтний спуск, Adam та RMSprop.

Розглядаючи стохастичний градієнтний спуск важливо зауважити, що в бібліотеках PyTorch та TensorFlow реалізовано розширення стохастичного градієнтного спуску яке дозволяє використовувати принцип інерції. При використанні цього розширення для зміна вагів розраховується за формулою

$$v_t = \mu v_{t-1} + \eta \frac{dL}{dW} \quad (4.6)$$

де v_t є накопиченою швидкістю оновлення, а μ є коефіцієнтом пришвидшення, що дозволяє уникати локальних мінімумів та пришвидшувати навчання. Далі під час оновлення вагів часткова похідна помножена на швидкість навчання моделі замінюється на накопичену швидкість оновлення [142]. Іншими популярними методами градієнтного спуску є Adam, який використовує поєднання інерційного стохастичного градієнтного спуску з адаптивною швидкістю навчання, та RMSprop, в якому використовується контрольоване оновлення швидкості для кожного параметру окремо, що дозволяє уникати проблеми з високою варіативністю градієнтів для частини параметрів.

Отже бібліотеки TensorFlow та PyTorch мають широкий функціонал та дозволяють ефективно будувати та навчати нейронні мережі, проте особливості їх архітектур та розбіжності у функціоналі роблять бібліотеку TensorFlow більш ефективною при використанні її для побудови складних та оптимізованих систем завдяки можливості ефективно оброблювати дані та модифікувати моделі для використання їх на пристроях з обмеженими ресурсами. Бібліотека PyTorch на противагу має більш гнучку систему для створенні та тестування нейронних мереж, що робить її більш ефективною для дослідницьких цілей.

4.3. Програмне забезпечення модулю для проведення прунінгу перед тренуванням

Для реалізації методів оптимізації навчання було вирішено розробити ієрархічну структуру класів, яка буде реалізовувати відповідні методи прунінгу. Відповідно до описаних алгоритмів прунінгу було розроблено функціонал для розрахунку критерії для наборів вагів які містять маскуючі ваги (маски), які можуть містити значення 1 або 0. Також для адаптації архітектур нейронних мереж до застосування методу прунінгу було розроблено функціонал для додавання масок а архітектуру нейронної мережі під час її створення.

Розроблені методи були адаптавані для роботи з бібліотекою PyTorch, так як вона пропонує модульну структуру при якій нейронні мережі складаються з сукупності базових блоків. З метою додавання масок до таких блоків було реалізовано набір класів, які розширяють функціонал найбільш поширених шарів нейронних мереж.

Для реалізації методів прунінгу було створено базовий абстрактний клас `BasePruning` який окреслює фундаментальну структуру та методи які є необхідними для застосування будь-якої стратегії прунінгу. Далі на основі цього класу будують класи які реалізують конкретні стратегії прунінгу, що дозволяє ефективно перевикористовувати код.

При ініціалізації абстрактного класу треба передати в нього список масок для параметрів нейронної мережі, для яких треба застосувати прунінг. Також абстрактний клас має абстрактний метод `score` який використовується для ранжування важливості для кожного параметра та потребує перевизначення у класах наслідниках. Також клас `BasePruning` має методи `update_masks`, який оновлює значення масок основуючись на обчислених критеріях важливості та заданого рівня стиснення для моделі, та методу `get_model_stats`, який

відповідає за визначення кількості параметрів мережі за допомогою підрахунку ненульових елементів.

Для реалізації методу SNIP було створено клас `SNIP_Prining`, який розширює абстрактний клас `BasePruning`, та має реалізацію методу `score` на основі алгоритму SNIP, дозволяючи розраховувати оцінки, відкриває можливість маскам мати градієнти та виконує зворотній прохід для обчислення градієнтів втрат для кожного параметру. Далі ці оцінки нормалізуються з метою надання відносної міри важливості для кожного параметру.

Далі було реалізовано клас `AttentionBasedPruning` який також є розширенням абстрактного класу `BasePruning`, реалізуючи не тільки метод `score`, але і модифікуючи конструктор шляхом додавання до нього посилань на тензори активації уваги та виходи шарів уваги, як показано на рисунку 4.2. Метод `score` у реалізації цього класу враховує не лише функцію втрати, але і включає в себе суму активацій та виходів уваги отримані під час зворотнього проходу. Також цей метод для приведення вагів до розрідженого формату використовує функцію `to_sparse_semi_structured` модулю `torch.sparse`, таким чином реалізуючи запропонований метод прунінгу.

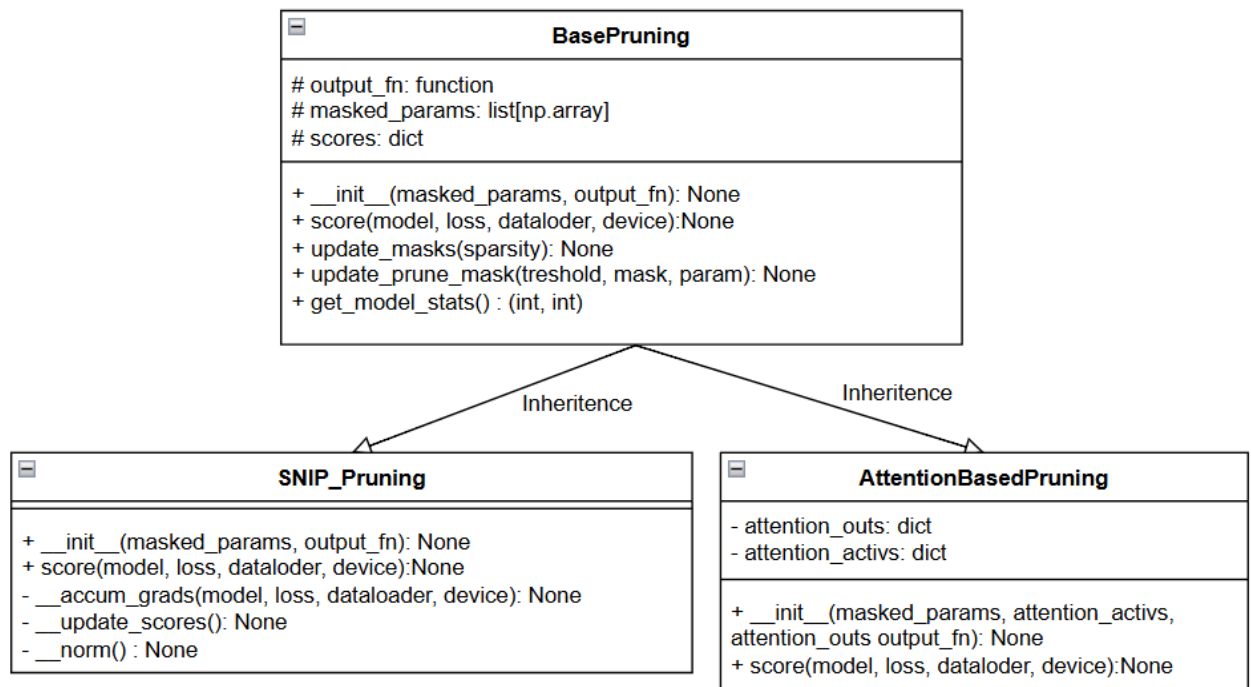


Рисунок 4.2 – Ієрархія класів для прунінгу

Модуль прунінгу було реалізовано як бібліотеку мови програмування python, що дозволяє використовувати її в інших проектах пов'язаних з дослідженням та оптимізацією моделей за допомогою покращеного методу прунінгу. Для використання цієї бібліотеки необхідно мати дані для тренування, архітектуру моделі, опис гіперпараметрів, так додатково можна мати трансформації, які необхідно провести для даних перед передачею їх до моделі. Після використання розробленої бібліотеки користувач може отримати метрики якості моделі, метрики швидкодії, такі як FLOP, ступінь стиснення та час виконання на графічному та центральному процесорах, та модель з розрідженими вагами, у форматі ONNX, як показано на рисунку 4.3

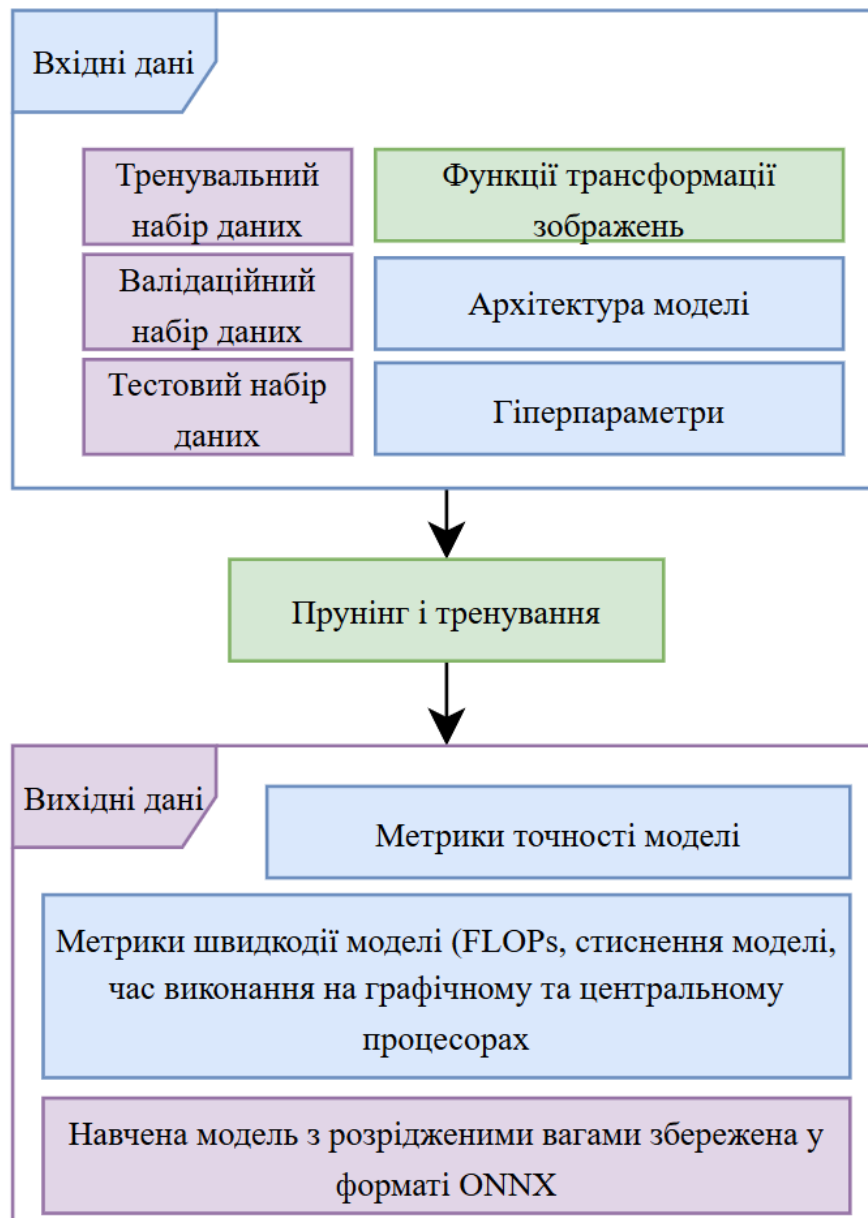


Рисунок 4.3 – Вхідні і вихідні дані при роботі з компонентами бібліотеки.

Для оцінки швидкодії моделі після застосування методу прунінгу було запропоновано проводити оцінку часу виконання на графічному та центральному процесорах. Для оцінки швидкості виконання на центральному процесорі використовується бібліотека `deepsparse`, яка дає можливість запуску моделей нейронних мереж у форматі ONNX [108], при цьому бібліотека

deepsparse містить в собі функціонал для оптимізації обчислень низького рівня, що дозволяють отримати додатковий приріст швидкодії на сучасних центральних процесорах. Для оцінки швидкодії моделі при застосуванні графічного процесору використовується бібліотека PyTorch, так як вона дозволяє використовувати функції для роботи з розрідженими матрицями формату 2:4, що є важливою складовою запропонованого методу прунінгу.

4.4. Програмне забезпечення модулю для розбиття відео на плани

Для реалізації розбиття відео на плани було розроблено спеціальний модуль ShotDetection, який дозволяє передавати на вхід відео та отримувати в результаті границі планів та ключові кадри. Спочатку було створено клас для роботи з відео VideoReader, який дозволяє считувати відеофайли та потоки відео. Конструктор класу VideoReader приймає на вхід шлях до відео та має додатковий параметр resize, при передачі якого після счиутвання кожного кадру його розмір буде змінюватись на заданий в цій змінній. Також для зручності роботи з відео було реалізовано властивості, які можуть забирати з метаданих відео його ширину, висоту, частоту кадрів та загальну кількість кадрів. Також під час счиутвання відео є можливість підвищення контрастності кадрів, яка досягається шляхом переведення зображення у кольорови простір hsv та модифікації частини просторів. Для більш широких можливостей під час счиутвання відео було реалізовано методи які можуть не тільки считувати кадри з відео, але повертати додаткову інформацію, таку як номер кадру чи точний часовий проміжок в якому знаходиться кадр. Також була реалізована можливість зчитувати декілька кадрів одночасно.

Також було творено допоміжний клас ImageComparer який дозволив зручно порівнювати зображення за допомогою бібліотеки OpenCV. Конструктор цього класу приймає на вхід кортеж з кількістю блоків для

розбиття кадру. Також клас `ImageComparer` містить методи для переведу зображень у різні кольорові простори, динамічного виділення частин зображення, обрахунку гістограм для двовимірних матриць, розрахунку оператора Собеля, та дистанцій між гістограмами. Основним методом класу `ImageComparer` є `compare` який приймає на вхід зображення для аналізу, яке далі переводиться у додаткові кольорові простори, розбивається на блоки, визначаються краї зображення. Також результати аналізу цих трансформацій зберігається в спеціальній змінній об'єкту класу, що дозволяє перевикористовувати їх, при аналізі наступного кадру. Таким чином після проведення аналізу кадру та визначення його особливостей, ці особливості порівнюються з особливостями попереднього кадру, повертаючи набір дистанцій для кожного блоку.

Також було створено клас для роботи з нейронною мережею `ShotDetectionRNN`, конструктор якого приймає на вхід шлях до моделі. Також цей клас має методи для завантаження моделі нейронної мережі інструментами бібліотеки `TensorFlow`, оновлення прихованого стану моделі та переведення прихованого стану у початкове положення. Основним методом класу `ShotDetectionRNN` є `process_frames` який приймає на вхід закодовану різницю між зображеннями за допомогою класу `ImageComparer`. Далі у випадку якщо це перший виклик цього методу, то за допомогою методу `update_states` оновлюються стани рекурентних слоїв, для використання контексту попереднього аналізу. Далі відбувається виклик моделі, та зберігається новий прихований стан, для подальших розрахунків. Такі особливості роботи за станами рекурентних слоїв нейронної мережі викликані потребою оптимізувати модель для розрахунків в реальному часі.

Клас `ShotChangeAnalyzer` є ключовим класом, який поєднує використання інших класів, та дозволяє спростити процес аналізу відео для

користувача. Конструктор класу приймає шлях до файлу конфігурації, в якому вказані додаткові характеристики для ініціалізації класів, такі як кількість блоків для розбиття чи шлях до нейронної мережі. Головним методом є `analyze`, який приймає на вхід шлях до відео, метод для зберігання результатів. Цей метод відповідає за поступове зчитування кадрів, передачі їх до класу порівняння зображень, та потім результати порівняння передаються до класу `ShotDetectionRNN`. У випадку визначення зміни сцени викликається метод для зберігання даних. Загальна схема модулю `ShotDetection` продемонстрована на рисунку 4.4.

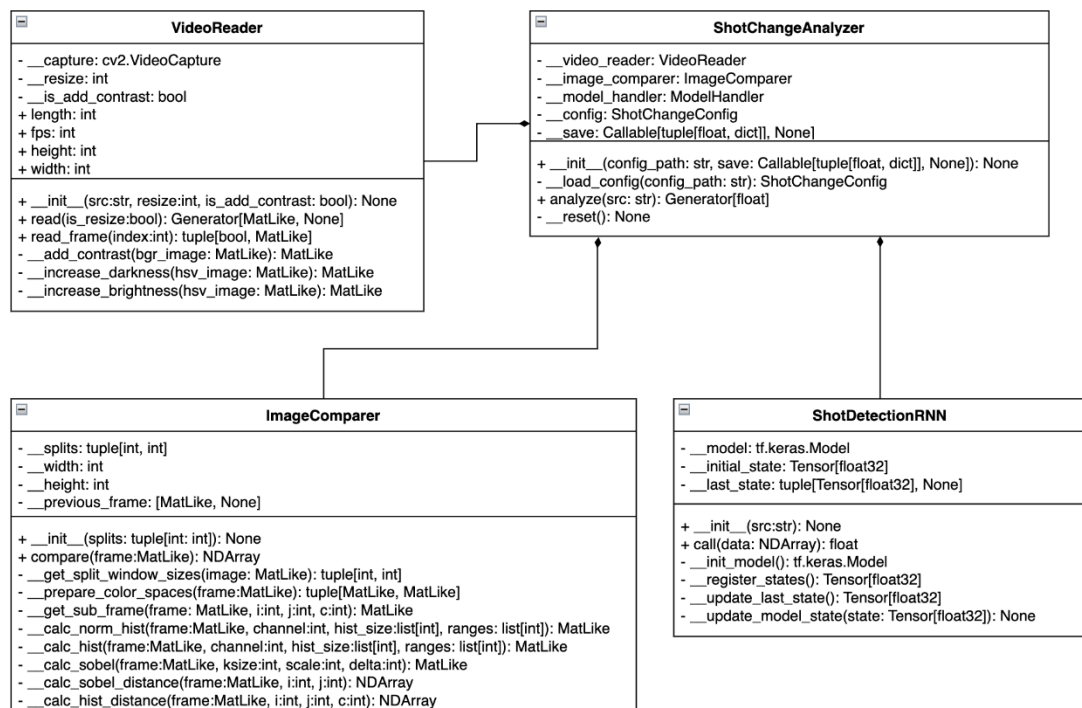


Рисунок 4.4 – Діаграма класів модулю `ShotDetection`

Передача результатів визначення до методу для зберігання даних дозволяє користувачеві застосовувати різні стратегії для обробки інформації, такі як зберігання результатів до файлу, зберігання ключових кадрів чи передача даних до інших систем чи потоків. Також цей модель реалізує спеціальний клас для вилучення ключових кадрів у сценах, що може

використовуватись для передачі отриманих кадрів до методу збереження даних. Це дозволяє не тільки розширити можливості подальшого аналізу, але і може бути корисним при тестуванні системи.

4.5. Програмне забезпечення модулю для визначення сцен

Для визначення зміни сцен на відео було розроблено спеціальний модуль SceneDetection, який розширяє можливості модулю ShotDetection. Для роботи з візуальним трансформером для відео було створено набір спеціалізованих класів для розширення можливостей бібліотеки PyTorch, а саме класи для просторової та часової уваги, нормалізації результатів шарів. Далі на основі цих класів було реалізовано основний клас ViViT, який за допомогою параметрів які передаються до конструктору будує модель візуального трансформеру для відео. Також цей клас має функціонал для створення просторових патчів для зображень та викликає побудовану модель.

Для інтеграції з модулем розпізнавання сцен було розроблено клас FrameLoader, який дозволяє в собі зберігати зображення для аналізу. Цей клас має в собі механізм асинхронної роботи з чергами та спеціальні методи, які дозволяють накопичувати кадри для передачі на аналіз відразу батчу даних. При цьому є можливість для передачі класу асинхронної черги як додатково параметру в конструкторі, що дозволяє використовувати спільну чергу для декількох потоків. Для цього було реалізовано метод збереження `add_frame_to_analyze`, який можна передати до `ShotChangeAnalyzer` як функцію для збереження даних. Метод `add_frame_to_analyze` використовує вбудовану чергу, розмір відповідає кількості зображень, що передаються до моделі архітектури візуального трансформеру для відео, та починаючи з моменту заповнення черги, зображення копіюються до загальної черги, яка зберігає в собі кортежі зображень. Так як цей клас реалізований за допомогою

асинхронного виконання, це дозволяє ефективно отримувати дані паралельно з виконанням задачі розпізнавання зміни сцен.

Також було розроблено клас `SceneChangeAnalyzer` який відповідає за поєднання цих методів та пропонує зручний інтерфейс для роботи користувача. Під час ініціалізації цього класу створюються об'єкти класу `FrameLoader` та `ViViT`. Далі завантажуються ваги для моделі візуального трансформеру для відео, шлях до яких передається як один з параметрів конструктора. Також конструктор приймає файл для конфігурації класу `ViViT` та `FrameLoader`. Основними методами класу `SceneChangeAnalyzer` є `analyze` та `start_analyze`. Метод `analyze` дає змогу викликати модель `ViViT` відразу передавши до неї параметри, після чого повертає результат роботи моделі. Метод `start_analyze` запускає цикл який перевіряє наявність об'єктів в черзі яку надає об'єкт класу `FrameLoader`, та при наявності одного чи більше об'єктів готує їх до аналізу та викликає метод `analyze`. При цьому метод `start_analyze` приймає як параметри функцію для збереження результатів та спеціальний токен для закінчення роботи функції. Загальна схема модулю `SceneDetection` продемонстрована на рисунку 4.5.

Розроблена структура модулю дозволяє легко інтегрувати його до інших модулів чи використовувати його для паралельного визначення зміни сцен у окремих потоках.

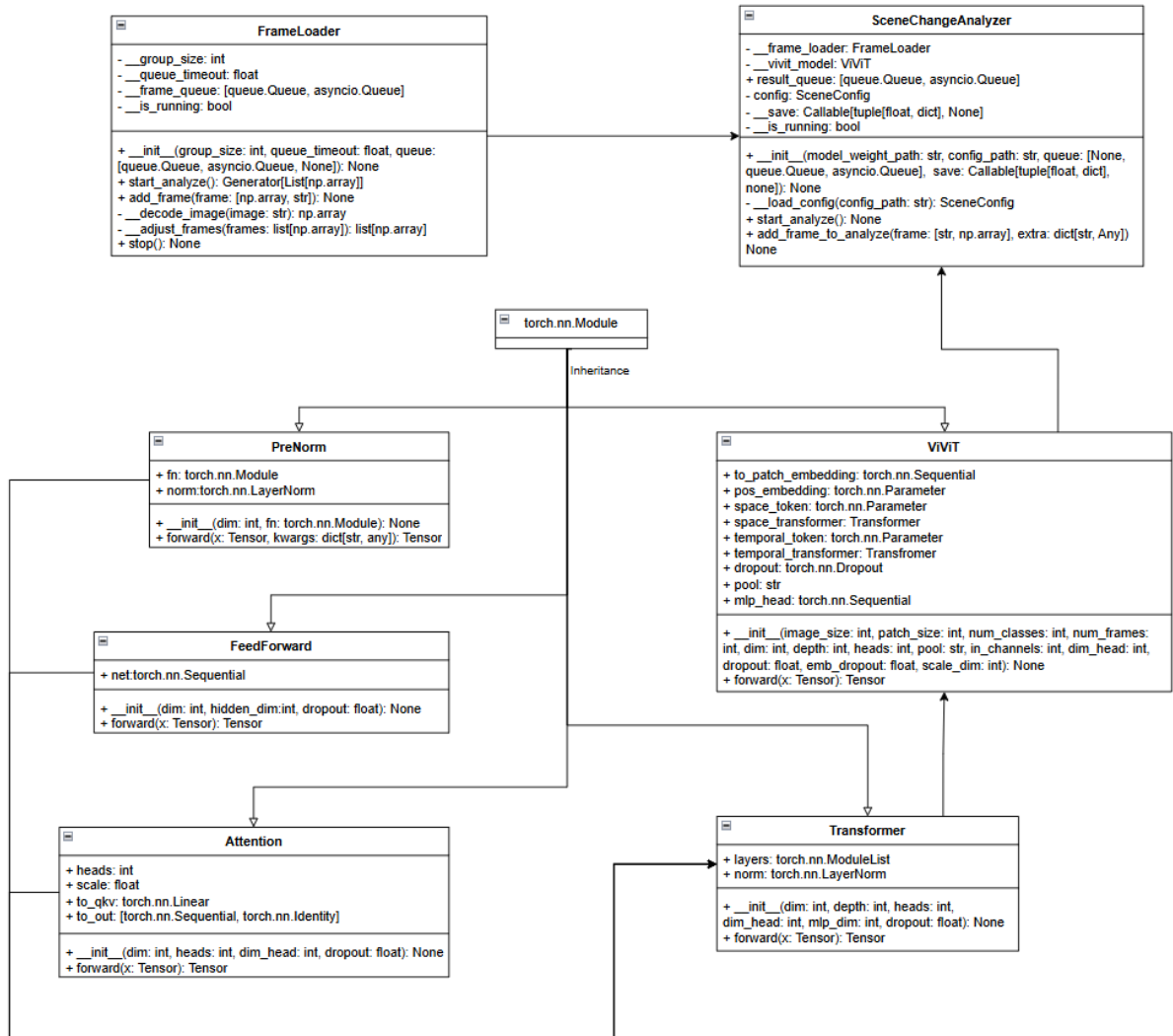


Рисунок 4.5 – Діаграма класів модулю SceneDetection.

4.6. Архітектура розподіленого програмного забезпечення для визначення відеоатрибутів

Для реалізації паралельного обчислення відеоатрибутів було створено спеціальний модуль `AttributeAnalyzerParallel`. Цей модуль складається з трьох частин, а саме визначення зміни плану, сцени та визначення атрибутів. Перша частина відповідає за считування відео, визначення зміни сцени, видалення ключових кадрів та передачі інформації до інших частин програми. Для визначення сцен було розроблено спеціальну обгортку яка дозволяє запускати

клас `SceneChangeAnalyzer` як окремий потік у вигляді серверу. Це дозволило спростити взаємодію з цим класом та використовувати `Docker` для обгортки моделей. Сервер при ініціалізації створює клас `SceneChangeAnalyzer`, та далі за допомогою веб-запитів дозволяє передавати дані на аналіз. Хоча недоліком цього підходу є потреба в кодуванні даних та збільшення часу перенесення даних між модулями, проте завдяки зменшенню необхідної кількості даних для передачі на аналіз цей недолік не є критичним. Також цей підхід дозволяє уникнути проблеми з налаштуванням пакетів для аналізу, яка може виникнути через потребу у застосуванні різних версій одних і тих самих пакетів чи при конфлікті залежностей. Також це дозволяє запускати моделі розроблені на інших мовах програмування, таких як `C++`, що стає особливо критичним при визначенні атрибутів, так як моделі для аналізу відеатрибутів можуть бути написані на різних мовах програмування та з застосуванням різних пакетів. Проте це висуває вимоги до побудови такого контейнеру, а саме він має містити в собі сервер, який буде приймати запити на обробку, проте для таких систем, які написані на мові програмування `Python` було розроблено спеціальний пакет, який дозволяє легко використовувати розроблені моделі за допомогою веб-серверу.

Також було створено клас для контролю за потоками чи контейнерами `SceneWorkerBalancer`, який має аналізувати завантаженість черги, та на її основі створювати додаткові потоки чи контейнери для аналізу або навпаки видаляти потоки та контейнери які не використовуються, як показано на рисунку 4.6. Також цей клас використовується для отримання результатів розпізнавання та подальшого збереження до бази даних.

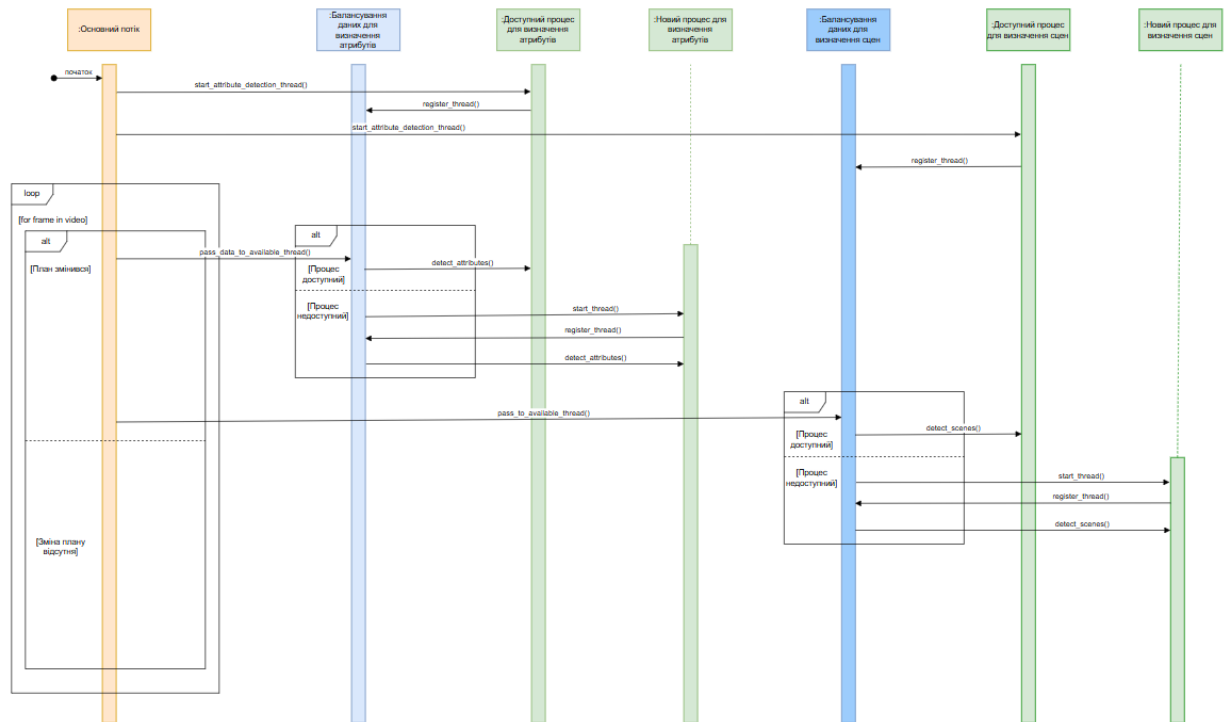


Рисунок 4.6 - Діаграми послідовностей розподіленого аналізу відео.

В результаті розробки модулю `AttrubuteAnalyzerParallel` вдалось побудувати архітектуру яка дозволяє ефективно керувати потоком даних при паралельній обробці інформації, як показано на рисунку 4.7. При цьому розроблений модуль дозволяє використовувати дозволяє використовувати його частини для розподілення цієї архітектури на різні сервери.

Для порівняння ефективності розробленого модулю було вирішено провести експерименти з порівняння швидкодії звичайним методом визначення відеоатрибутів та за допомогою розробленої архітектури. Для визначення відеоатрибутів було вирішено використовувати моделі для знаходження обличчя, визначення ключових точок тіла людини та

знаходження об'єктів на зображенні та модель визначення зміни сцени на відео.

Спочатку було вимірено швидкість обробки кадрів в секунду (FPS) при аналізі кожного кадру, що становило 0.95 FPS. Далі було вирішено розраховувати відеоатрибути та знаходження зміни сцен в окремих потоках, що дозволило пришвидшити роботу систему в 2.17 разів. Наступним кроком було вирішено визначати відеоатрибути та сцени синхрно, проте аналізуючи лише ключові кадри отримані за допомогою детекції зміни планів. Завдяки великій швидкості аналізу зміни сцен та значному зменшенню кількості кадрів для аналізу вдалось пришвидшити отримання результатів та аналізувати відео за швидкістю 10.58 FPS. Далі було застосовану розроблену архітектуру, до дозволило підвищити швидкість аналізу ще більше, досягнувши пришвидшення в 24.21 рази, як показано в таблиці 4.1.

Таблиця 4.1 – Порівняння швидкодії аналізу відео

Підхід	FPS	Пришвидшення
Визначення відеоатрибутів та зміни сцени для кожного кадру	0.95	1x
Визначення відеоатрибутів та зміни сцени для кожного кадру з використанням багатопоточності	1.85	2.17x
Визначення відеоатрибутів та зміни сцени для використовуючи ключові кадри отримані за допомогою знаходження зміни планів	10.58	11.14x
Визначення відеоатрибутів та зміни сцени використовуючи розроблений модуль	23	24.21x

Важливо зауважити, що отримане пришвидшення може змінюватися в залежності від довжини планів, проте розроблений модуль показав можливість значно збільшувати швидкість визначення відеоатрибутів, при цьому додатково визначаючи зміну планів та сцен на відео.

4.7. Програмне забезпечення модулю збереження та аналізу даних

Збереження даних про результати аналізу відео відбувається за допомогою використання бази даних MongoDB, яка дозволяє ефективно працювати зберігати різну аналітику, отриману в результаті розпізнавання відеоатрибутів. Основними колекціями є `users`, `videos`, `shots`, `scenes` та `video_attributes`, в яких зберігається вся інформація про оброблені відео. Базовою колекцією є `videos`, в яких зберігається `id` користувача, заповнене ім'я відео, його роздільна здатність, частота кадрів та тривалість. У колекціях `shots` та `scenes` зберігаються порядкові номери сцен та планів, їх початок і кінець у форматі часу. Колекція `video_attributes` зберігає в собі атрибути отримані при аналізі, як показано на рисунку 4.8.

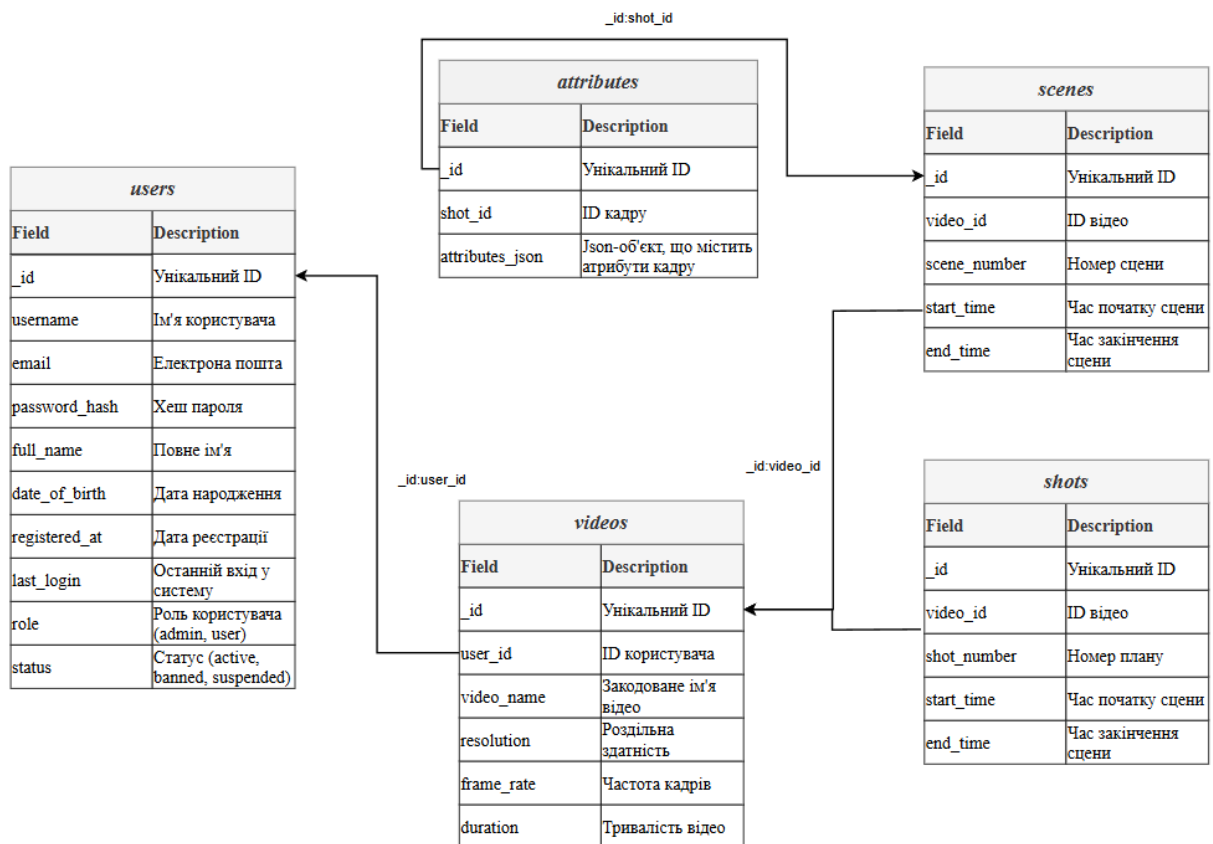


Рисунок 4.8 – Схема бази даних для збереження даних про користувачів та результати розпізнавання

Доступ до бази даних відбувається за допомогою спеціального класу, який має функціонал для підключення до бази даних, перевірки даних на валідність та запису. Така структура бази даних дозволяє ефективно зберігати інформацію та автоматично адаптуватись до потреби визначати різні набори відеоатрибутів. При подальшій роботі з базою даних є можливість будувати детальну статистику чи робити пошук по атрибутах експортуючи дані для подальшого аналізу у Power BI та Tableau.

4.8. Висновки до розділу 4

Розроблена архітектура дозволяє ефективно використовувати розбиття відео на сцени та плани, для значного пришвидшення швидкості визначення відеоатрибутів. Архітектурні рішення дозволяють використовувати розроблену систему як на одній машині, так і на різних, що може бути використано для застосуванні хмарних технологій. Розроблена архітектура має можливість ефективно використовувати Docker для спрощення інтеграції нових підходів для визначення відеоатрибутів. Алгоритм визначення зміни планів реалізовані за допомогою таких інструментів як OpenCV та TensorFlow, а алгоритм визначення зміни плани сцен та прунінгу реалізовані за допомогою бібліотеки PyTorch, та можуть бути розширені для подальших експериментів. Система використовує базу даних MongoDB, яка дозволяє зберігати дані для різного набору відеоатрибутів та надає доступ для подальшого аналізу.

ВИСНОВКИ

У результаті дисертаційного дослідження вирішено актуальне наукове завдання розробки моделей та програмних засобів для підвищення швидкодії визначення атрибутів у відео за допомогою розбиття на плани та сцени з використанням візуальних трансформерів та застосуванням розподіленої архітектури. Дане наукове завдання має суттєве значення для теоретичних основ та програмно-алгоритмічного забезпечення в області глибоких нейронних мереж. Високі показники покращення точності і швидкодії в порівнянні з роботами вітчизняних та закордонних вчених робить результати досліджень пріоритетними.

В дисертації одержані такі основні результати:

1. Продемонстровано нагальну потребу в розробленні універсального програмного забезпечення для визначення змін планів і сцен у відео. Аналіз існуючих методів розбиття відео на плани за допомогою математичних підходів показав низьку точність при високій швидкодії, тоді як використання нейронних мереж підвищує точність, але суттєво знижує швидкодію. Аналіз методів розбиття відео на сцени виявив низьку точність, зумовлену обмеженнями існуючих підходів щодо визначення контексту сцен.

2. Вперше розроблено розподілену архітектуру для швидкого визначення атрибутів у відео шляхом розбиття на плани та сцени на основі нейронних мереж. На відміну від існуючих рішень, вона дозволяє розподіляти аналіз відео між різними системами, ефективно використовуючи обмін даними між серверними компонентами. Завдяки зменшенню кількості викликів методів для визначення атрибутів досягається прискорення щонайменше у 2,5–3 рази.

3. Розроблено новий алгоритм розбиття відео на плани, який, на відміну від існуючих методів, поєднує математичний підхід із рекурентною нейронною мережею. Це дозволило значно підвищити точність розпізнавання,

досягнувши F1-оцінки 85,5% та точності влучення 93,9%, зберігаючи високу швидкодію, що дає змогу аналізувати відео в режимі реального часу.

4. Розроблено новий алгоритм розбиття відео на сцени, який, на відміну від існуючих підходів, що аналізують усі кадри, використовує лише ключові кадри, отримані в результаті розбиття відео на плани. Це дозволило значно підвищити швидкість аналізу відео та точність розпізнавання моделі на основі візуального трансформера, досягнувши F1-оцінки 72,1%, що на 5,1% вище за показник Deep Multimodal Networks.

5. Набув подальшого розвитку метод прунінгу перед навчанням для моделей архітектури візуальних трансформерів для відео, який, на відміну від існуючих підходів, враховує важливість механізму «уваги» та забезпечує прискорення виконання моделі на 10%. При цьому запропонований метод прунінгу використовує напіврозріджені матриці формату 2:4, що дозволяє досягти ще більшого прискорення під час виконання оптимізованої моделі на спеціалізованому апаратному забезпеченні.

6. Практичне використання результатів роботи, розроблених моделей та програмних засобів дозволило підвищити точність і швидкодію аналізу відеоконтенту завдяки розробленій архітектурі розподіленого програмного забезпечення для визначення атрибутів у відео на основі розбиття на плани та сцени. На відміну від існуючих рішень, ця архітектура ефективно розподіляє обчислення та реалізує запропоновані й удосконалені методи та алгоритмічне забезпечення. Реалізація удосконаленого методу прунінгу для архітектури візуального трансформера на відео дозволила натренувати нейронну мережу, яка на 10% швидша за оригінальну. Реалізація розробленого методу поєднання математичного підходу з рекурентними нейронними мережами дозволила натренувати дві нейронні мережі, які перевищують точність влучання та F1-оцінку відносно підходу AutoShot на 4.3% та 4.4% відповідно. При цьому розроблений підхід має обчислювальні вимоги у розмірі до 500 kFLOPS, що дозволяє використовувати цей підхід для розв'язання задач у реальному часі.

Реалізація розробленого методу для визначення зміни сцени в відео на основі візуального трансформера для відео дозволила натренувати нейронну мережу, яка перевищує F1-оцінку відносно підходів оснований на глибоких мультимодальних мережах на 5.43%. Розроблено програмне забезпечення, яке на відміну від існуючих, дозволяє ефективно аналізувати атрибути для відео використовуючи сцени та плани отримані в режимі реального часу

7. Результати досліджень прийняті до впровадження в Товаристві з обмеженою відповідальністю «ВОТЧЕД» (акт від 10.02.2025.); в навчальному процесі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (акт впровадження від 24.02.2025р.) при викладанні дисципліни «Цифрова обробка зображень» для студентів освітньо-кваліфікаційного рівня «Магістр» спеціальності 122 «Комп'ютерні науки».

8. Як можливі напрямки подальших досліджень можна відзначити розробку нових методів визначення зміни сцен із використанням мультимодальних даних, зокрема аудіо та тексту. Інтеграція моделей для розпізнавання та класифікації аудіосигналів дозволить точніше визначати контекст сцен і переходи між ними. Крім того, аналіз аудіосигналів і тексту сприятиме глибшому аналізу відеоконтенту та створенню детальнішої статистики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Khan, Laeeq. Social Media Engagement: What motivates User Participation and Consumption on YouTube?. *Computers in Human Behavior*. 2017. № 66. P. 236–247.
2. N. Dimitrova, Hong-Jiang Zhang, B. Shahraray, I. Sezan, T. Huang, A. Zakhor. Applications of video-content analysis and retrieval. *IEEE MultiMedia*. 2002. vol. 9, № 66. 3. P. 42-55.
3. H. C. Shih. A Survey of Content-Aware Video Analysis for Sports. *IEEE Transactions on Circuits and Systems for Video Technology*. 2018. vol. 28, № 5. P. 1212-1231.
4. S. Wang, Q. Ji. Video Affective Content Analysis: A Survey of State-of-the-Art Methods. *IEEE Transactions on Affective Computing*. 2015. vol. 6, № 4. P. 410-430.
5. Kousha K., Thelwall M., Abdoli M. The role of online videos in research communication: A content analysis of YouTube videos cited in academic publications. *Journal of the American Society for Information Science and Technology*. 2012. № 63. P. 1710-1727.
6. Melnychenko A., Zdor K. Incorporating attention score to improve foresight pruning on transformer models. *Computer Science and Applied Mathematics*. 2023. №2. P. 22-27.
7. Melnychenko A., Zdor K. Efficiency of supplementary outputs in siamese neural networks. *Advanced Information Systems*. 2023. №3 (7). P. 49–53.
8. Здор К. А., Шалденко О. В. . *Зв'язок*. 2024. №6(172). С. 91-97
9. Zhao, Zhong-Qiu, Peng Zheng, Shou-tao Xu, Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*. 2019. № 30(11). P. 3212-3232.
10. Szegedy C., Toshev A., Erhan D. Deep neural networks for object detection. *Advances in neural information processing systems*. 2013. №26.

11. Sung J., Ponce C., Selman B., Saxena A. Unstructured human activity detection from rgb-d images. *IEEE international conference on robotics and automation*. 2012. P. 842-849.
12. Hannane R., Elboushaki A., Afdel K., Naghabhushan P., Javed M. An efficient method for video shot boundary detection and keyframe extraction using SIFT-point distribution histogram. *International Journal of Multimedia Information Retrieval*. 2016. № 5. P.89-104.
13. Kweon I.S., Han S., Yoon, K. A new technique for shot detection and key frames selection in histogram space. *12th Workshop on Image Processing and Image Understanding*. 2000.
14. Koprinska I., Carrato S., Temporal video segmentation: A survey. *Signal processing: Image communication*. 2001. № 16(5). P. 477-500.
15. Xu Y.S., Fu T.J., Yang H.K., Lee C.Y. Dynamic video segmentation network. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. P. 6556-6565.
16. Siam M., Valipour S., Jagersand M., Ray N. Convolutional gated recurrent networks for video segmentation. *IEEE international conference on image processing (ICIP)*. 2017. P. 3090-3094.
17. Abdolrasol M.G., Hussain S.S., Ustun T.S., Sarker M.R., Hannan M.A., Mohamed R., Ali J.A., Mekhilef S., Milad A. Artificial neural networks based optimization techniques: A review. *Electronics*. 2021. № 10(21). P. 2689.
18. Yang Y., Xing Z., Zhu L. Vivim: a video vision mamba for medical video object segmentation. *arXiv preprint arXiv:2401.14168*. 2024.
19. Stocker V., Knieps G. Digitalizing telecommunications: innovation, complexity and diversity in the internet ecosystem. In *A Modern Guide to the Digitalization of Infrastructure*. 2021. P. 59-91.
20. Ni J., Shen K., Chen Y., Cao W., Yang S.X. An improved deep network-based scene classification method for self-driving cars. *IEEE Transactions on Instrumentation and Measurement*. , 2022. № 71. P. 1-14.

21. Zuo Z., Wang G., Shuai B., Zhao L., Yang Q., Jiang X. Learning discriminative and shareable features for scene classification. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland. 2014. Proceedings, Part I* 13 P. 552-568.
22. Koonce B., Koonce B. EfficientNet. *Convolutional neural networks with swift for Tensorflow: image recognition and dataset categorization*. 2021. P. 109-123.
23. Arnab A., Dehghani M., Heigold G., Sun C., Lučić M., Schmid C. Vivit: A video vision transformer. *Proceedings of the IEEE/CVF international conference on computer vision*. 2021. P. 6836-6846.
24. Zou Z., Chen K., Shi Z., Guo Y., Ye J. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*. 2023. №111(3). P. 257-276.
25. Caelles S., Maninis K.K., Pont-Tuset J., Leal-Taixé L., Cremers D., Van Gool L. One-Shot Video Object Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. P. 221-230.
26. Spitz J., Moors P., Wagemans J., Helsen W.F. The impact of video speed on the decision-making process of sports officials. *Cognitive Research: Principles and Implications*. 2018. №3. P. 1-10.
27. Zhao J., Gu Q., Zhao S., Mao J. Effects of video-based training on anticipation and decision-making in football players: A systematic review. *Frontiers in Human Neuroscience*. 2022. №16. P. 945067.
28. Zhao S., Gao Y., Jiang X., Yao H., Chua T.S., Sun X. Exploring principles-of-art features for image emotion recognition. *Proceedings of the 22nd ACM International Conference on Multimedia*. 2014. P. 47-56.
29. Ramani A., Rao A., Vidya V., Prasad V.B. Automatic subtitle generation for videos. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE. 2020. P. 132-135.
30. Eronen A.J., Peltonen V.T., Tuomi J.T., Klapuri A.P., Fagerlund S., Sorsa T., Lorho G., Huopaniemi J. Audio-based context recognition. *IEEE*

Transactions on Audio, Speech, and Language Processing. 2005. №14(1). P. 321-329.

31. Schmidt, R.M. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv preprint arXiv:1912.05911*.

32. Pătrăucean V., He X.O., Heyward J., Zhang C., Sajjadi M.S., Muraru G.C., Zholus A., Karami M., Goroshin R., Chen Y., Osindero S. TRecViT: A Recurrent Video Transformer. *arXiv preprint arXiv:2412.14294*. 2024.

33. Medsker L.R., Jain L. Recurrent neural networks. *Design and Applications*. 2001. №5(64-67). P. 2.

34. Schuster M., Paliwal K. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*. 1997. № 45(11). P.2673-2681.

35. Yu Y., Si X., Hu C., Zhang J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*. 2019. №31(7). P. 1235-1270.

36. Fernández J.G., Keemink S., van Gerven M. Gradient-free training of recurrent neural networks using random perturbations. *Frontiers in Neuroscience*. 2024. №18. P. 1439155.

37. Pascanu R. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*. 2013.

38. O'Shea, K. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*. 2015.

39. Yang J., Li J. Application of deep convolution neural network. *14th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE. 2017. P. 229-232.

40. Lou Y., Wu R., Li J., Wang L., Li X., Chen G. A learning convolutional neural network approach for network robustness prediction. *IEEE Transactions on Cybernetics*. 2022. №53(7). P. 4531-4544.

41. Szegedy C., Ioffe S., Vanhoucke V., Alemi A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2017. №31(1).
42. Dhruv P., Naskar S. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review. *Machine Learning and Information Processing: Proceedings of ICMLIP 2019*. 2020. P. 367-381.
43. Xu Z., Hu J., Deng W. Recurrent convolutional neural network for video classification. *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2016. P. 1-6.
44. Ji S., Xu W., Yang M., Yu K. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012. №35(1). P. 221-231.
45. Alakwaa W., Nassef M., Badr A. Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). *International Journal of Advanced Computer Science and Applications*. 2017. №8(8).
46. Chen Y., Liu J., Zhang X., Qi X., Jia J. LargeKernel3D: Scaling up kernels in 3D sparse CNNs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023. P. 13488-13498.
47. Mittal S. A survey of accelerator architectures for 3D convolution neural networks. *Journal of Systems Architecture*. 2021. №115. P. 102041.
48. Neha F., Bhati D., Shukla, D.K., Amiruzzaman M. From classical techniques to convolution-based models: A review of object detection algorithms. *arXiv preprint arXiv:2412.05252*. 2024.
49. Amjoud A.B., Amrouch M. Object detection using deep learning, CNNs and vision transformers: A review. *IEEE Access*. 2023. №11. P. 35479-35516.
50. De Leon F., Gómez P., Martinez-Velasco J.A., Rioual M. Transformers. *In: Power System Transients*. CRC Press. 2017. P. 177-250.

51. Vaswani A. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017.
52. Sanford C., Hsu D., Telgarsky M. Transformers, parallel computation, and logarithmic depth. *arXiv preprint*.arXiv:2402.09268. 2024.
53. Liu Y., Zhang Y., Wang Y., Hou F., Yuan J., Tian J., Zhang Y., Shi Z., Fan J., He Z. A survey of visual transformers. *IEEE Transactions on Neural Networks and Learning Systems*. 2023.
54. Wu B., Xu C., Dai X., Wan A., Zhang P., Yan Z., Tomizuka M., Gonzalez J., Keutzer K., Vajda P. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint*. arXiv:2006.03677. 2020.
55. Higashi T., Ishibashi R., Meng L. ViViT fall detection and action recognition. *2024 International Conference on Advanced Mechatronic Systems (ICAMechS)*. IEEE. 2024. P. 291-296.
56. Hu Y., Lu X. Learning spatial-temporal features for video copy detection by the combination of CNN and RNN. *Journal of Visual Communication and Image Representation*. 2018. №55. P. 21-29.
57. Khan A., Rauf Z., Sohail A., Khan A.R., Asif H., Asif A., Farooq U. A survey of the vision transformers and their CNN-transformer based variants. *Artificial Intelligence Review*. 2023. №56(Suppl 3). P. 2917-2970.
58. Ma Q., Jiang J., Liu X., Ma J. Learning a 3D-CNN and transformer prior for hyperspectral image super-resolution. *Information Fusion*. 2023. №100. P. 101907.
59. Shan X., Ma T., Gu A., Cai H., Wen Y. TCRNet: Make transformer, CNN and RNN complement each other. *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022. P. 1441-1445.
60. Liu Z., Sun M., Zhou T., Huang G., Darrell T. Rethinking the value of network pruning. *arXiv preprint*.arXiv:1810.05270. 2018.

61. Nagel M., Fournarakis M., Amjad R.A., Bondarenko Y., Van Baalen M., Blankevoort T. A white paper on neural network quantization. *arXiv preprint*. arXiv:2106.08295. 2021.
62. Gou J., Yu B., Maybank S.J., Tao D. Knowledge distillation: A survey. *International Journal of Computer Vision*. 2021. №129(6). P. 1789-1819.
63. Chong J., Gupta M., Chen L. Resource efficient neural networks using Hessian-based pruning. *arXiv preprint*. arXiv:2306.07030. 2023.
64. Huang D., Xiong Y., Xing Z., Zhang Q. Implementation of energy-efficient convolutional neural networks based on kernel-pruned silicon photonics. *Optics Express*. 2023. №31(16). P. 25865-25880.
65. Chakraborty B., Kang B., Kumar H., Mukhopadhyay S. Sparse spiking neural network: Exploiting heterogeneity in timescales for pruning recurrent SNN. *arXiv preprint*. arXiv:2403.03409. 2024.
66. Li Z., Chen T., Li L., Li B., Wang Z. Can pruning improve certified robustness of neural networks?. *arXiv preprint arXiv:2206.07311*. 2022.
67. Frankle J., Carbin M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint*. arXiv:1803.03635. 2018.
68. Banner R., Hubara I., Hoffer E., Soudry D. Scalable methods for 8-bit training of neural networks. *Advances in Neural Information Processing Systems*. 2018. №31.
69. Wang P., Hu Q., Zhang Y., Zhang C., Liu Y., Cheng J. Two-step quantization for low-bit neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018. P. 4376-4384.
70. Hubara I., Courbariaux M., Soudry D., El-Yaniv R., Bengio Y. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*. 2018. №18(187). P. 1-30.
71. Park E., Yoo S., Vajda P. Value-aware quantization for training and inference of neural networks. *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. P. 580-595.

72. Zhou Y., Moosavi-Dezfooli S.M., Cheung N.M., Frossard P. Adaptive quantization for deep neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018. Vol. 32, №1.
73. Gou J., Yu B., Maybank S.J., Tao D. Knowledge distillation: A survey. *International Journal of Computer Vision*. 2021. №129(6). P. 1789-1819.
74. Wang C.C., Xu S., Fu J., Liu Y., Wang B. KD-FixMatch: Knowledge distillation siamese neural networks. *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2023. P. 341-345.
75. Shin S., Boo Y., Sung W. Knowledge distillation for optimization of quantized deep neural networks. *2020 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE. 2020. P. 1-6.
76. He H., Wang J., Zhang Z., Wu F. Compressing deep graph neural networks via adversarial knowledge distillation. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022. P. 534-544.
77. Li C., Cheng D., Zhang G., Li Y., Zhang S. Toward fair graph neural networks via dual-teacher knowledge distillation. *arXiv preprint*. arXiv:2412.00382. 2024.
78. Stringa E., Regazzoni C.S. Real-time video-shot detection for scene surveillance applications. *IEEE Transactions on Image Processing*. 2000. №9(1). P. 69-79.
79. Küçüktunç O., Güdükbay U., Ulusoy Ö. Fuzzy color histogram-based video segmentation. *Computer Vision and Image Understanding*. 2010. №114(1). P. 125-134.
80. Cheng S.C., Su J.Y., Hsiao K.F., Rashvand H.F. Latent semantic learning with time-series cross-correlation analysis for video scene detection and classification. *Multimedia Tools and Applications*. 2016. №75. P. 12919-12940.
81. Ren J., Shen X., Lin Z., Mech R. Best frame selection in a short video. In *Proceedings of the IEEE/CVF Winter Conference on applications of computer vision*. 2020. P. 3212-3221.

82. Takahashi S., Sakaguchi Y., Kouno N., Takasawa K., Ishizu K., Akagi Y., Aoyama R., Teraya N., Bolatkan A., Shinkai N., Machino H. Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review. *Journal of Medical Systems*. 2024. №48(1). P. 84.
83. Trinh L., Anwar A., Mercelis S. SeaDSC: A video-based unsupervised method for dynamic scene change detection in unmanned surface vehicles. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024. P. 840-847.
84. Lin C.J., Garg S., Chin T.J., Dayoub F. Robust Scene Change Detection Using Visual Foundation Models and Cross-Attention Mechanisms. *arXiv preprint*. arXiv:2409.16850. 2024.
85. Armeniakos G., Zervakis G., Soudris D., Henkel J. Hardware approximate techniques for deep neural network accelerators: A survey. *ACM Computing Surveys*. 2022. №55(4). P. 1-36.
86. Liao Y., Naghizadeh P. Social bias meets data bias: The impacts of labeling and measurement errors on fairness criteria. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2023. Vol. 37, No. 7. P. 8764-8772.
87. Tommasi T., Patricia N., Caputo B., Tuytelaars T. A deeper look at dataset bias. *Domain Adaptation in Computer Vision Applications*. 2017. P. 37-55.
88. Gowda S.N., Arnab A., Huang J. Optimizing ViViT training: Time and memory reduction for action recognition. *arXiv preprint*. arXiv:2306.04822. 2023.
89. Gowda S.N., Arnab A., Huang J. Optimizing factorized encoder models: Time and memory reduction for scalable and efficient action recognition. *European Conference on Computer Vision*. 2024. P. 457-474.
90. Chicco D. Siamese neural networks: An overview. *Artificial Neural Networks*. 2021. P. 73-94.
91. Koch G., Zemel R., Salakhutdinov R. Siamese neural networks for one-shot image recognition. *ICML Deep Learning Workshop*. 2015. Vol. 2, №2. P. 1-30.

92. Nandy A., Haldar S., Banerjee S., Mitra S. A survey on applications of siamese neural networks in computer vision. *2020 International Conference for Emerging Technology (INCET)*. IEEE. 2020. P. 1-5.
93. Zheng W., Yang L., Genco R.J., Wactawski-Wende J., Buck M., Sun Y. SENSE: Siamese neural network for sequence embedding and alignment-free comparison. *Bioinformatics*. 2019. №35(11). P. 1820-1828.
94. Wang F., Liu H. Understanding the behaviour of contrastive loss. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. P. 2495-2504.
95. Ghojogh B., Sikaroudi M., Shafiei S., Tizhoosh H.R., Karray F., Crowley M. Fisher discriminant triplet and contrastive losses for training siamese networks. *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020. P. 1-7.
96. Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015. P. 1-9.
97. Xiao H., Rasul K., Vollgraf R. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint*. arXiv:1708.07747. 2017.
98. Singh D., Jain N., Jain P., Kayal P., Kumawat S., Batra N. PlantDoc: A dataset for visual plant disease detection. *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. 2020. P. 249-253.
99. Hastari D., Winanda S., Pratama A.R., Nurhaliza N., Ginting E.S. Application of convolutional neural network ResNet-50 V2 on image classification of rice plant disease. *Public Research Journal of Engineering, Data Technology and Computer Science*. 2024. №1(2).
100. Deng J., Dong W., Socher R., Li L.J., Li K., Fei-Fei L. ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009. P. 248-255.

101. Huh M., Agrawal P., Efros A.A. What makes ImageNet good for transfer learning? *arXiv preprint*. arXiv:1608.08614. 2016.
102. Mas J., Fernandez G. Video shot boundary detection based on color histogram. *TRECVID*. 2003.
103. Soyoung P., Jeongwoo. S., Sun-Joong K. Study on the effect of frame size and color histogram bins on the shot boundary detection performance. 2016. P. 1-2.
104. Kahu S.Y., Raut R.B., Bhurchandi K.M. Review and evaluation of color spaces for image/video compression. *Color Research & Application*. 2019. №44(1). P. 8-33.
105. Zedan I., Elsayed K., Emary E. Abrupt Cut Detection in News Videos Using Dominant Colors Representation. 2016.
106. Mas J., Fernandez G. Video shot boundary detection based on color histogram. *TRECVID*. 2003
107. Huan Z., Xiuhuan L., Lilei Y. Shot boundary detection based on mutual information and Canny edge detector. 2008. P. 1124-1128.
108. Zhu W. AutoShot: A short video dataset and state-of-the-art shot boundary detection. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2023. P. 2238-2247.
109. Yamak P.T., Yujian L., Gadosey P.K. A comparison between ARIMA, LSTM, and GRU for time series forecasting. *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*. 2019. P. 49-55.
110. Cutting J.E., Candan A. Shot durations, shot classes, and the increased pace of popular movies. 2015.
111. Dangel F., Tatzel L., Hennig P. ViViT: Curvature access through the generalized Gauss-Newton's low-rank structure. *arXiv preprint*. arXiv:2106.02624. 2021.

112. Singh S., Dewangan S., Krishna G.S., Tyagi V., Reddy S., Medi P.R. Video vision transformers for violence detection. *arXiv preprint*. arXiv:2209.03561. 2022.
113. LeCun Y., Denker J., Solla S. Optimal brain damage. *Advances in Neural Information Processing Systems*. 1989. №2.
114. Hassibi B., Stork D.G., Wolff G.J. Optimal brain surgeon and general network pruning. *IEEE International Conference on Neural Networks*. 1993. P. 293-299.
115. Castellano G., Fanelli A.M., Pelillo M. An iterative pruning algorithm for feedforward neural networks. *IEEE Transactions on Neural Networks*. 1997. №8(3). P. 519-531.
116. Rachwan J., Zügner D., Charpentier B., Geisler S., Ayle M., Günnemann S. Winning the lottery ahead of time: Efficient early network pruning. *International Conference on Machine Learning*. 2022. P. 18293-18309.
117. Anwar S., Hwang K., Sung W. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*. 2017. №13(3). P. 1-18.
118. Yang Z., Zhang H. Comparative analysis of structured pruning and unstructured pruning. *International Conference on Frontier Computing*. 2021. P. 882-889.
119. Guo Y., Yao A., Chen Y. Dynamic network surgery for efficient DNNs. *Advances in Neural Information Processing Systems*. 2016. №29.
120. Lee N., Ajanthan T., Torr P.H. SNIP: Single-shot network pruning based on connection sensitivity. *arXiv preprint*. arXiv:1810.02340. 2018.
121. Voita E., Talbot D., Moiseev F., Sennrich R., Titov I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint*. arXiv:1905.09418. 2019.
122. Michel P., Levy O., Neubig G. Are sixteen heads really better than one? *Advances in Neural Information Processing Systems*. 2019. №32.

123. Sanh V., Wolf T., Rush A. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*. 2020. №33. P. 20378-20389.
124. Devlin J. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*.arXiv:1810.04805. 2018.
125. Floridi L., Chiriatti M. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*. 2020. №30. P. 681-694.
126. Dai Z., Yang Z., Yang Y., Cohen W.W., Carbonell J., Le Q.V., Salakhutdinov R. Transformer-XL: Language modeling with longer-term dependency. 2018.
127. Markidis S., Der Chien S.W., Laure E., Peng I.B., Vetter J.S. NVIDIA Tensor Core programmability, performance & precision. *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2018. P. 522-531.
128. Cebrian J.M., Natvig L., Jahre M. Scalability analysis of AVX-512 extensions. *The Journal of Supercomputing*. 2020. №76(3). P. 2082-2097.
129. Nazir Z., Yarovenko V., Park J.G. nterpretable ML enhanced CNN Performance Analysis of cuBLAS, cuDNN and TensorRT. *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 2023. P. 1260-1265.
130. Wang E., Zhang Q., Shen B., Zhang G., Lu X., Wu Q., Wang Y. Intel Math Kernel Library. *High-Performance Computing on the Intel® Xeon Phi™: How to Fully Exploit MIC Architectures*. 2014. P. 167-188.
131. Zhang H., Zhao Y., Zheng J., Zhuang C., Gu J., Chen G. CSR: Achieving 1-bit key-value cache via sparse representation. *arXiv preprint*. arXiv:2412.11741. 2024.
132. Dery L., Kolawole S., Kagy J.F., Smith V., Neubig G., Talwalkar A. Everybody prune now: Structured pruning of LLMs with only forward passes. *arXiv preprint*. arXiv:2402.05406. 2024.

133. Zhang Y., Zhao L., Lin M., Sun Y., Yao Y., Han X., Tanner J., Liu S., Ji R. Dynamic sparse no training: Training-free fine-tuning for sparse LLMs. *arXiv preprint*. arXiv:2310.08915. 2023.
134. Mishra A., Latorre J.A., Pool J., Stosic D., Stosic D., Venkatesh G., Yu C., Micikevicius P. Accelerating sparse deep neural networks. *arXiv preprint*. arXiv:2104.08378. 2021.
135. Huang J., Yu C.D., van de Geijn R.A. Implementing Strassen's algorithm with CUTLASS on NVIDIA Volta GPUs. *arXiv preprint*. arXiv:1808.07984. 2018.
136. Pang B., Nijkamp E., Wu Y.N. Deep learning with TensorFlow: A review. *Journal of Educational and Behavioral Statistics*. 2020. №45(2). P. 227-248.
137. Harris C.R., Millman K.J., Van Der Walt S.J., Gommers R., Virtanen P., Cournapeau D., Wieser E., Taylor J., Berg S., Smith N.J., Kern R. Array programming with NumPy. *Nature*. 2020. №585(7825). P. 357-362.
138. Pulli K., Baksheev A., Korniyakov K., Eruhimov V. Real-time computer vision with OpenCV. *Communications of the ACM*. 2012. №55(6). P. 61-69.
139. Olston C., Fiedel N., Gorovoy K., Harmsen J., Lao L., Li F., Rajashekhar V., Ramesh S., Soyke J. TensorFlow-Serving: Flexible, high-performance ML serving. *arXiv preprint*. arXiv:1712.06139. 2017.
140. Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*. 2019. №32.
141. Werbos P.J. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*. 1990. №78(10). P. 1550-1560.

142. Andrychowicz M., Denil M., Gomez S., Hoffman M.W., Pfau D., Schaul T., Shillingford B., De Freitas N. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*. 2016. №29.

143. Здор К.А., Шалденко О.В. Концепція обробки зображення на основі багатозадачних сіамських нейронних мереж. *XX Міжнародна науково-практична конференція молодих вчених і студентів*. Київ. 25–28 квітня 2023 року. С. 210-211.

144. Melnychenko A., Zdor K. Applying classification and regression supplementary output in siamese neural network using Fashion MNIST and PlantVillage datasets. *VII Міжнародна науково-практична конференція «Modern Problems of Science, Education and Society»*. Київ, Україна. 11-13 вересня 2023. С. 126-129.

145. Melnychenko A., Zdor K. Appling classification and regression supplemetary outputs in siamese neural network using plantvillage dataset, I Міжнародна науково-практична конференція «Current challenges of science and education», 18-20 вересня 2023, Берлін, Німеччина. С. 79-82.

146. Melnychenko A., Zdor K. Appling classification and regression supplemetary output in siamese neural network using fashion MNIST and plantvillage datasets, X Міжнародна науково-практична конференція «Innovations and prospects in modern science», 25-27 вересня 2023, Стокгольм, Швеція. С. 87-92.

147. Мельниченко А., Здор К. Збільшення ефективності оптимізації моделей архітектури ViT перед навчанням шляхом включення активацій механізму самоуваги, I міжнародна науково–практична конференція «Сучасні аспекти інженерії програмного забезпечення», 14 грудня 2023, Київ, Україна.

148. Мельниченко А.В., Здор К.А. Врахування механізмів самоуваги при прунінгу моделей нейронних мереж Vision Transformer. Збірник матеріалів III Міжнародної науково-технічної конференції «Системи і

технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку», 30 листопада 2023 року, Київ, Україна. С. 214 – 215.

149. Zdor K., Shaldenko O., Nedashkivskiy O., Melnychenko A., Leveraging ViViT transformers and foresight pruning for scalable scene change detection on distributed architecture, *Зв'язок*. 2025. №1. Р. 3-8.

Додаток А

Наукові праці, в яких опубліковані основні наукові результати дисертації:

1. Здор К. А., Шалденко О. В. Нейро-математичний підхід для виявлення змін планів у відеопослідовностях. Зв'язок. 2024. №6(172). С. 91-97
2. Melnychenko, A., Zdor K. Incorporating attention score to improve foresight pruning on transformer models. Computer Science and Applied Mathematics, 2023, №2, P.18-22.
3. Melnychenko, A., Zdor, K. Efficiency of supplementary outputs in siamese neural networks. Advanced Information Systems, 2023, Volume 7, №3, P. 49–53.
4. Zdor K., Shaldenko O., Nedashkivskiy O., Melnychenko A., Leveraging ViViT transformers and foresight pruning for scalable scene change detection on distributed architecture, Зв'язок. 2025. №1. Р. 3-8.

Наукові праці, які засвідчують апробацію матеріалів дисертації:

1. Здор К. А., Шалденко О.В. Концепція обробки зображення на основі багатозадачних сіамських нейронних мереж, XX Міжнародна науково-практична конференція молодих вчених і студентів, м. Київ, 25–28 квітня 2023 року, с. 210-211
2. Melnychenko, A., Zdor, K. Applying classification and regression supplementary output in siamese neural network using fashion MNIST and plantvillage datasets, VII Міжнародна науково-практична конференція «Modern problems of science, education and society», 11-13 вересня 2023 Київ, Україна, С. 126-129.

3. Melnychenko, A., & Zdor, K. Applying classification and regression supplementary outputs in siamese neural network using plantvillage dataset, I Міжнародна науково-практична конференція «Current challenges of science and education», 18-20 вересня 2023, Берлін, Німеччина. С. 79-82.

4. Melnychenko A., Zdor K. Applying classification and regression supplementary output in siamese neural network using fashion MNIST and plantvillage datasets, X Міжнародна науково-практична конференція «Innovations and prospects in modern science», 25-27 вересня 2023, Стокгольм, Швеція. С. 87-92.

5. Мельниченко А., Здор К. Збільшення ефективності оптимізації моделей архітектури ViT перед навчанням шляхом включення активацій механізму самоуваги, I міжнародна науково-практична конференція «Сучасні аспекти інженерії програмного забезпечення», 14 грудня 2023, Київ, Україна.

6. Мельниченко А.В., Здор К.А. Врахування механізмів самоуваги при прунінгу моделей нейронних мереж Vision Transformer. Збірник матеріалів III Міжнародної науково-технічної конференції «Системи і технології зв'язку, інформатизації та кібербезпеки: актуальні питання і тенденції розвитку», 30 листопада 2023 року, Київ, Україна. С. 214 – 215.

Додаток Б

ЗАТВЕРДЖУЮ
проректор з навчальної роботи
КПІ ім.Ігоря Сікорського,
кандидат технічних наук, доцент



Тетяна ЖЕЛЯСКОВА

« 24 » лютого 2025 р.

АКТ

про впровадження результатів дисертаційних досліджень аспіранта кафедри інженерії програмного забезпечення в енергетиці, навчально-наукового інституту атомної та теплової енергетики Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Здра Костянтина Андрійовича за темою «Моделі та програмні засоби підвищення швидкодії визначення відеоатрибутів за допомогою розбиття на сцени» на здобуття наукового ступеня доктора філософії за спеціальності 121 «Інженерія програмного забезпечення»

Отримані особисто Здором Костянтином Андрійовичем, результати дисертаційних досліджень за темою «Моделі та програмні засоби підвищення швидкодії визначення відеоатрибутів за допомогою розбиття на сцени», а саме: алгоритм розбиття відео на плани на основі поєднання математичного підходу з рекурентними нейронними мережами; алгоритм розбиття відео на плани використовуючи ключові кадри та модель архітектури візуального трансформера для відео- впроваджені в навчальний процес кафедри «Цифрові технології в енергетиці» у лекційні та лабораторні заняття навчальної дисципліни: «Цифрова обробка зображень та комп'ютерний зір» за спеціальністю 122 Комп'ютерні науки за освітнім рівнем підготовки магістра (2024-2025):

Лекція: Методи сегментації, засновані на кластеризації. Методи сегментації з використанням гістограм;

Лекція: Алгоритми колірної сегментації та їх реалізація;

Лекція: Системи обробки 3D зображень;


Лабораторна робота: Дослідження можливостей покращення зображень методом вирівнювання гістограм.

Зазначене актуалізує зміст навчальної дисципліни до потреб практики.

Директор навчально-наукового інституту
атомної та теплової енергетики,
доктор технічних наук, професор


Євген ПИСЬМЕННИЙ

Завідувач кафедри
цифрових технологій в енергетиці,
доктор технічних наук, професор


Наталія АУШЕВА

Вчений секретар кафедри
цифрових технологій в енергетиці,
кандидат технічних наук, доцент


Світлана ШАПОВАЛОВА

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ «ВОТЧЕД»
04119, м. Київ, вул. Хохлових Сім'ї, буд. 8, код ЄДРПОУ 43433547

№2010-1

Від 10.02.2025

АКТ

**Впровадження результатів і висновків дисертаційної роботи
Здора Костянтина Андрійовича
«Моделі та програмні засоби підвищення швидкодії визначення
відеоатрибутів за допомогою розбиття на сцени»**

Даний акт засвідчує, що результати та висновки дисертаційної роботи Здора Костянтина Андрійовича «Моделі та програмні засоби підвищення швидкодії визначення відеоатрибутів за допомогою розбиття на сцени» використані для оптимізації визначення сцен в рамках грантового договору № 957321 на реалізацію проєкту START-UP DRIVEN INNOVATION IN EUROPEAN MEDIA (STADIEM) у межах Рамкової програми Європейської Комісії Горизонт 2020 «Дослідження реакції аудиторії на аудіовізуальний контент за допомогою нейронних мереж».

Керівник відділу досліджень
ТОВ «ВОТЧЕД»



Безсмертний В.Ю.

Додаток В

Лістинг коду класів для визначення зміни сцен:

```
import os
import time
import numpy as np
from trailer_analyses.utils.image_comparer import ImageComparer
from trailer_analyses.video_io import VideoReader
from trailer_analyses.utils.base_utils import del_extra_indexes
import pandas as pd
from datetime import datetime, timedelta

from trailer_analyses.video_io.video_reader_cv import VideoReaderCV

class VideoTimecodesAnalyzer:
    def __init__(self, batch_size = 60, std_k = 1., std_k2 = 1.7,
min_distance=None):
        #std_k2 = 1.5
        self.batch_size = 30
        self.batch_size = batch_size
        self.image_comparer = ImageComparer()
        self.std_k = 1.
        self.std_k = std_k
        self.std_k2 = 6.5
        self.std_k2 = std_k2
        self.min_distance = min_distance

    def analyze(self, data_source, path):
```

```

        save_path,    detailed_save_path=    path.replace('.mp4',
'.csv').replace('data/', 'data/results/'), path.replace('.mp4', '_det.csv').replace('data/',
'data/results/')

        start_time_record = time.time()

        assert isinstance(data_source, VideoReader) or isinstance(data_source,
VideoReaderCV)

        self.data_source = data_source

        self.min_distance = self.min_distance or self.data_source.fps//2

        start_index = 0

        global_data = []

        global_data_indexes = []

        while start_index + self.batch_size*self.data_source.fps <=
self.data_source.length:

            data = self._get_distances(start_index)

            data, data_indexes = self._find_scenes(data, start_index)

            if len(data_indexes):

                if start_index != data_indexes[-1]:

                    start_index = data_indexes[-1]

                else:

                    start_index = start_index + self.batch_size*self.data_source.fps

                    global_data.append(data[:data_indexes[-1]])

                    global_data_indexes.extend(data_indexes)

            else:

                start_index = start_index + self.batch_size*self.data_source.fps

                print(start_index + self.batch_size*self.data_source.fps, '/',
self.data_source.length)

        data = self._get_distances(start_index)

        data, data_indexes = self._find_scenes(data, start_index)

```



```

# start_index = data_indexes[-1]
if len(data_indexes):
    global_data.append(data[:data_indexes[-1]])
    global_data_indexes.extend(data_indexes)
    global_data_indexes = del_extra_indexes(global_data_indexes,
self.min_distance)

start_time = datetime(2000,1,1,0,0,0,0)
times = sorted(set(['00:00:00.000']+[start_time +
timedelta(seconds=t/self.data_source.fps)).strftime('%H:%M:%S.%f') for t in
global_data_indexes]))
detailed_df = pd.DataFrame({'start_time':times})
detailed_df['name'] = range(1,1+len(detailed_df))
if detailed_save_path:
    os.makedirs(os.path.dirname(detailed_save_path), exist_ok=True)
    detailed_df.to_csv(detailed_save_path, index=False)
df = self._rename_results(detailed_df)
if save_path:
    os.makedirs(os.path.dirname(save_path), exist_ok=True)
    df.to_csv(save_path, index=False)
end_time = time.time()
print(f'Video timecodes analysis took {end_time-start_time_record}
seconds')
return df

def _rename_results(self, df):
    df['start_time'] = df['start_time'].apply(lambda x: x.split('.')[0][1:])
    df = df.drop_duplicates()
    df['name'] = range(1,1+len(df))

```

```
return df
```

```
def _get_distances(self, start):
```

```
    prev_frame= None
```

```
    results = []
```

```
    for frame in self.data_source.read_batch(start, self.batch_size):
```

```
        if prev_frame is None:
```

```
            prev_frame = frame
```

```
            result = self.image_comparer.compare(prev_frame, frame)
```

```
            results.append(result)
```

```
            prev_frame = frame
```

```
    return np.array(results).T
```

```
def _find_scenes(self, data, extra):
```

```
    results = []
```

```
    for item in data:
```

```
        self.model.layers[1].reset_states(states=final_states)
```

```
        y = self.model.predict(np.array([[item]]), verbose=0)[0][0]
```

```
        results.append(y>self.detection_threshold)
```

```
        final_states = self.model.layers[1].states
```

```
    data_indexes = np.where(data)[0]
```

```
    data_indexes = data_indexes+extra
```

```
    data_indexes = del_extra_indexes(data_indexes, self.min_distance)
```

```
    return data, data_indexes
```