Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ СТВОРЕННЯ КОНСПЕКТІВ ЛЕКЦІЙ НА ОСНОВІ АНАЛІЗУ ВІДЕО, АУДІО-КОНТЕНТУ ТА ПРЕЗЕНТАЦІЇ

(тема)

Виконав:
здобувач 4 року навчання,
групи ІТІНФ-21-2
Максімов Г. Р.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Яковлева О. В.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувачеві Максімову Глібу Романовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка застосунку для створення конспектів лекцій на основі аналізу відео, аудіо-контенту та презентації

затверджена наказом університету від 19 травня 2025 року № 381Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 24 травня 2025 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, мова програмування Python, бібліотеки Pandas, NumPy, PyTorch, pydub, librosa, ffmpeg, moviepy, iohttp, transformers, whisper, pytubefix, CLIP, DBSCAN, HDNSCAN, SigLip,yt-dlp, середовище Jupyter Notebook та VSCode, Flask.

4. Перелік питань, що потрібно опрацювати в роботі
1. Прогрес та можливості використання штучного інтелекту в різних сферах.
2. Огляд існуючих програмних рішень для створення конспектів.
3. Існуючі моделі для узагальнення текстової інформації.
4. Варіанти ефективного навчального конспекту в цифровому форматі.
5. Розробка макету конспекту на основі відео, аудіоконтенту та презентації.
6. Проєктування та розробка застосунку для створення конспектів лекцій.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми автоматичного створення конспектів, наявні методи та програмні рішення до вирішення задачі автоматичного нотування, постановка задачі, тестові відео та презентації, структура моделей отримання векторів ознак зображень та текстових описів, перевірка подібності векторів ознак зображень, розробка алгоритму групування кадрів відео за належністю до слайду відповідної презентації, експериментами з моделями, висновок щодо обрання моделі, алгоритм пошуку на основі векторів, діаграми варіантів використання та взаємодії компонентів застосунку, ілюстрація роботи застосунку, оцінювання конспектів незалежними експертами та аналіз результатів, аналіз точності роботи застосунку.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-14.04.25	
4	Аналіз технічних засобів	15.04.25-20.04.25	
5	Розробка методу	21.04.25-27.04.25	
6	Програмна реалізація	28.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Яковлева О. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 62 с., 6 табл., 22 рис., 1 дод., 33 джерело.

СЕГМЕНТАЦІЯ ВІДЕО, ТРАНСКРИБУВАННЯ ВІДЕО, ДІАРИЗАЦІЯ, УЗАГАЛЬНЕННЯ ВІДЕО, ОБРОБКА ВІДЕО, ВІДОБРАЖЕННЯ ВІДЕОКАДРІВ НА СЛАЙДИ, ВИЛУЧЕННЯ ІНФОРМАЦІЇ, ПОШУК ЗОБРАЖЕНЬ, ТЕКСТОВИЙ ЗАПИТ, КОСИНУСНА ПОДІБНІСТЬ, CLIP-МОДЕЛІ, ОБРОБКА ЗОБРАЖЕНЬ, DBSCAN, HDBSCAN.

Об’єктом роботи є питання створення конспектів лекцій на основі мультимодального контенту.

Метою роботи є розробка застосунку для створення конспектів лекцій на основі аналізу відео, аудіо-контенту та презентації.

В роботі було проведено аналіз існуючих програм для транскрибування тексту та нотування відео. Було розроблено метод зіставлення відповідних кадрів із слайдами, який базується на поєднанні текстової та щільної векторної подібності зображень разом із динамічним пороговим значенням за допомогою алгоритму кластеризації DBSCAN.

У результаті було розроблено програмне забезпечення для автоматичного створення нотаток із відео та презентації.

VIDEO SEGMENTATION, VIDEO TRANSCRIBATION, DIARIZATION, VIDEO SUMMARIZATION, VIDEO PROCESSING, MAPPING VIDFEO FRAMES TO SLIDES, INFORMATION EXTRACTION, IMAGE SEARCH, TEXT QUERY, COSINE SIMILARITY, CLIP MODELS, IMAGE PROCESSING, DBSCAN, HDBSCAN.

The object of work is the issue of creating lecture notes based on multimodal content.

The aim of the work is to develop an application for creating lecture notes based on the analysis of video, audio content and presentation.

The Methods used are transcription and summarization along with automatic slide-to-frame mapping. The method of mapping the relevant frames to slides is based on the fusion of textual and dense vector similarity of the images along with dynamic thresholding using DBSCAN clustering algorithm.

As a result, we have software for automated note taking from video and presentation.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 9
1 Сучасний прогрес в узагальненні інформації з відео, аудіо та тексту 10
1.1 Прогрес та можливості використання штучного інтелекту в різних сферах 10
1.2 Існуючі моделі для узагальнення текстової інформації 12
1.3 Варіанти ефективного навчального конспекту в цифровому форматі 14
1.4 Огляд існуючих програмних рішень для створення конспектів.... 15
1.5 Метрики оцінки якості електронного конспекту 16
1.6 Постановка задачі 18
2 Побудова алгоритму створення конспектів лекцій на основі аналізу відео, аудіоконтенту та презентації 20
2.1 Розробка макету конспекту на основі відео, аудіоконтенту та презентації 20
2.2 Обробка аудіоконтенту 21
2.2.1 Пошук фрагментів з людським мовленням 23
2.3 Обробка зображень 25
2.3.1 Виокремлення ознак з зображень 26
2.3.2 Дослідження алгоритмів кластеризації та ІЕМ 27
2.4 Алгоритм групування кадрів за належністю до слайду 30
2.5 Узагальнення та структурування інформації 32
2.6 Загальний алгоритм для створення конспекту 33
3 Розробка застосунку для генерування конспекту 36
3.1 Вибір та налаштування програмного середовища 36
3.2 Загальна архітектура системи 38
3.3 Підготовка набору даних із відео та презентацій до них 40
3.4 Ілюстрація роботи застосунку 42

3.5 Оцінка якості створеного конспекту 50
Висновки..... 54
Перелік джерел посилання 56
Додаток А Приклади конспектів..... 61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – Штучний інтелект

Гц – одиниця вимірювання, що дорівнює одному коливанню на секунду

SOTA – State of the Art (сучасний рівень розвитку технології)

GPT – Generative Pre-Trained Transformer (генеративний попередньо навчений трансформер)

VAD – Voice Activity Detection (виявлення голосової активності)

WAV – Waveform Audio File Format (формат аудіофайлів із представленням у вигляді хвильової форми)

OCR – Optical Character Recognition (оптичне розпізнавання символів)

IOU – Intersection over Union (перетин над об’єднанням – метрика схожості областей)

FPS – Frames Per Second (кадрів за секунду)

LLM – Large Language Model (велика мовна модель)

IEM – Image Embedding Model (модель векторного представлення зображень)

DBSCAN – Density-Based Spatial Clustering of Applications with Noise (кластеризація на основі щільності з урахуванням шуму)

API – Application Programming Interface (інтерфейс прикладного програмування)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

HTML – HyperText Markup Language (мова розмітки гіпертексту)

PCM_S16LE – Pulse-Code Modulation, 16-bit signed little-endian (імпульсно-кодова модуляція, 16-бітне знакове число у форматі «молодший байт перший»)

CLIP – Contrastive Language-Image Pretraining (контрастивне попереднє навчання на зображеннях і текстах)

SigLIP – Sigmoid Language-Image Pretraining (сигмоїдальне попереднє навчання на мовно-візуальних даних)

T5 – Text-to-Text Transfer Transformer (трансформер для перенесення між завданнями у форматі текст у текст)

CV – Computer Vision (комп’ютерний зір)

NLP – Natural Language Processing (обробка природної мови)

ВСТУП

Нотування є дуже важливою складовою процесу засвоєння матеріалу лекції, але, іноді, на це зовсім не має часу. Це процес, який складається із прочитання тексту-джерела, виділення основних положень у ньому, відбору прикладів і комбінування матеріалу, формування тексту конспекту.

Автоматизація цього процесу може значно прискорити процес навчання та його якість. Основна ціль нотування – структуризація та поглиблений аналіз інформації.

Актуальність роботи полягає у тому, що у наш час є багато інформації для засвоєння студентом, особливо під час дистанційного навчання. Зважаючи на відомі умови дистанційної освіти, студент повинен переглянути багато записів лекцій у найкоротші терміни і зробити це швидко зі збереженням якості.

Робота присвячена розробці застосунку для створення конспектів лекцій на основі аналізу відео, аудіо-контенту та презентації, який дозволяє узагальнити інформацію з аудіо, класифікувати характер тексту, та прив’язати до кожного слайду презентації узагальнену інформацію, що йому відповідає. Для побудови такого конспекту використовувалися методи та моделі транскрибування аудіо, нормалізації транскрибованого тексту, пошуку кадрів, синхронізації мультимодального контенту, узагальнення і структурування інформації.

1 СУЧАСНИЙ ПРОГРЕС В УЗАГАЛЬНЕННІ ІНФОРМАЦІЇ З ВІДЕО, АУДІО ТА ТЕКСТУ

1.1 Прогрес та можливості використання штучного інтелекту в різних сферах

Багато років підходи комп'ютерного зору (Computer Vision, CV) та обробки природної мови (Natural Language Processing, NLP) спиралися на методи, які використовували шаблони та функції, що побудовано на основі евристики. Такими класичними методами CV є, наприклад, оператори Кенні, Собела, Превитта, Лапласа та морфологічні операції для виділення контурів, детектори Харріса для визначення кутів, дескриптори SIFT, SURF, ORB для опису характерних точок на зображенні [1-4], матриця збігів та маски Лавса для визначення текстурних ознак [5-8]. Для класифікації зображень використовувалися методи машинного навчання, наприклад, такі як метод опорних векторів (Support Vector Machine, SVM), випадковий ліс (Random Forest). Такі підходи дозволяли автоматизувати багато задач, але вони дуже були чутливі до завад та вимагали постійних налаштувань.

Справжній прорив відбувся із появою глибинного навчання, а саме згорткових нейронних мереж (Convolutional Neural Networks, CNN), які з'явилися у 2012 році та стали основним інструментом для вирішення задач комп'ютерного зору. Незважаючи на величезний прогрес нейронних мереж, у реальних проєктах досі широко використовуються комбінації класичних алгоритмів та нейронних моделей. Класичні методи часто застосовуються для попередньої обробки даних, наприклад, бінаризація, підвищення якості зображень, виділення важливих ділянок, а вже глибинні нейронні мережі виконують складніші завдання, такі як розпізнавання та інтерпретація контенту [9-14].

Паралельно з розвитком CV, величезний прогрес відбувся в області NLP, зокрема через створення великих мовних моделей (Large Language Models,

LLM). Такі моделі як GPT, T5 (Text-to-Text Transfer Transformer) навчилися ефективно працювати з текстами завдяки попередньому тренуванню на великих корпусах даних та використанню архітектури трансформерів. Завдяки високій якості аналізу та генерації тексту, можливості узагальнювати LLM активно впроваджуються в реальні продукти: чат-боти, пошукові системи, системи для консультування клієнтів [15-16].

Сьогодні існує доволі багато задач, де штучний інтелект (ШІ) зарекомендував себе як сервіс з найвищим рівнем розвитку (SOTA – State Of The Art), наприклад:

- підсумовування тексту;
- генерація тексту;
- опис зображень;
- генерація зображень;
- транскрибування усного мовлення;
- конвертація тексту в усне мовлення.

В багатьох задачах ШІ справляється із завданнями на рівні людини, або навіть випереджає людські можливості. На рисунку 1.1 наведені результати випробувань систем штучного інтелекту на предмет різних можливостей порівняно з людськими можливостями, де в кожньому направленні початкову продуктивність ШІ встановлено на рівні -100, продуктивність людини використовується як базовий орієнтир, встановлений на нулі. Коли показники ШІ перетинають нульову позначку, це означає, що система працює краще за людину. Ще донедавна жоден метод ШІ не міг надійно розпізнавати мову чи зображення на рівні людини. Проте за цей час можливості ШІ суттєво зросли, і тепер в окремих тестах системи він демонструють результати, що перевищують людські.

Також часто виникають задачі, де поєднуються методи класичного CV, CV на основі нейронних мереж і LLM. Одним із прикладів таких задач є створення конспектів лекцій на основі аналізу відео, аудіо-контенту та презентацій.

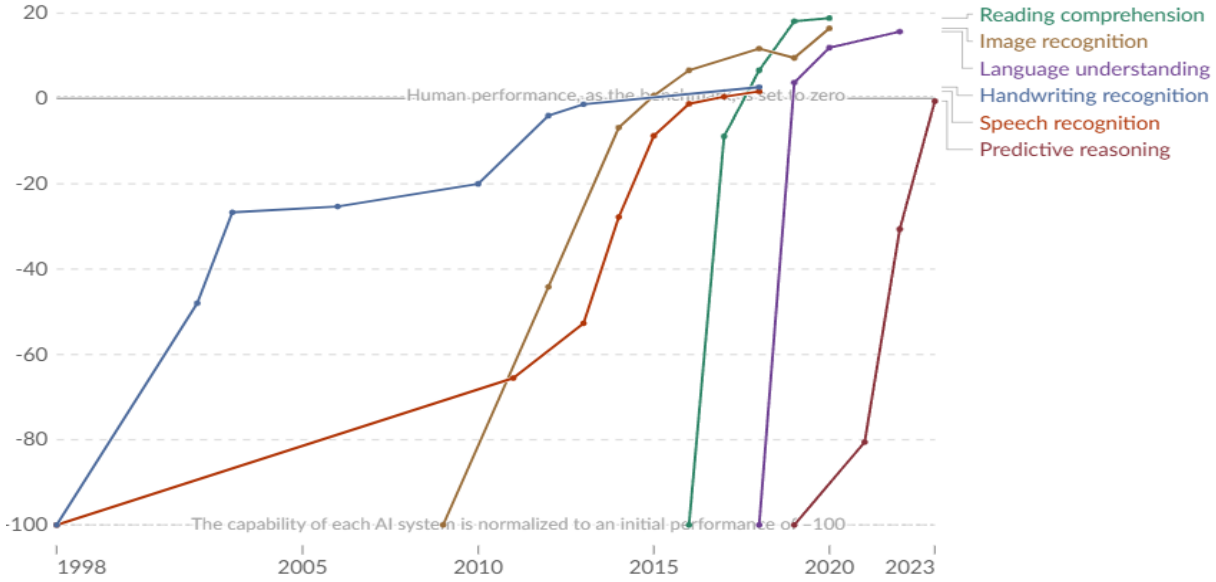

Рисунок 1.1 – Результати випробувань систем ШІ на предмет різних можливостей порівняно з людськими можливостями [17]

Для розв’язання вище вказаних задач необхідно:

– обробити відео та презентаційні матеріали (задача CV);

– розпізнати усне мовлення (задача speech-to-text);

– структурувати отриману текстову інформацію та виділити ключові моменти (задача для LLM);

– інтегрувати результати в єдиний зручний формат для користувача.

Таким чином, сучасні методи ШІ технології відкривають нові можливості для автоматизації складних когнітивних задач, які класичними методами CV та NLP було вирішити не можливо.

1.2 Існуючі моделі для узагальнення текстової інформації

На даний момент існує доволі багато моделей для узагальнення текстової інформації. Найбільш відомими представниками є моделі архітектури GPT – генеративні попередньо навчені трансформери, наприклад

Claude, Mistral, DeepSeek, Wizard, Gemini, BLOOM, Llama тощо. Схема архітектури вказана на рисунку 1.2. Схема оригінальної архітектури трансформерної мережі наведена на рисунку 1.3.

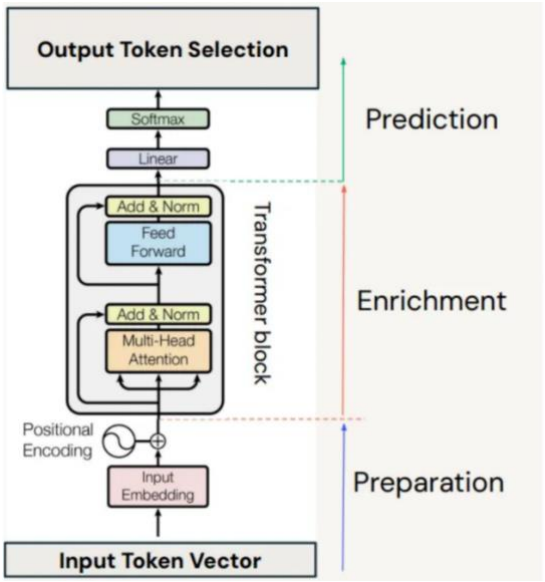

Рисунок 1.2 – Схема архітектури трансформерної нейромережі GPT [18]

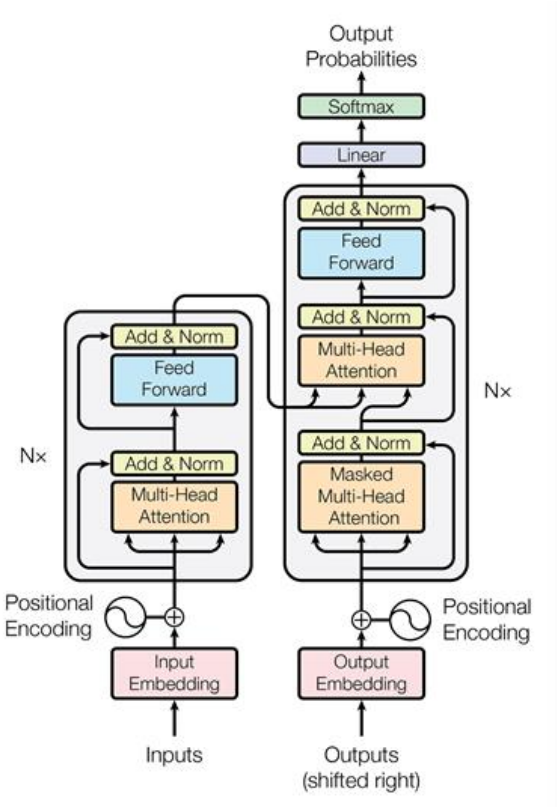

Рисунок 1.3 – Схема оригінальної архітектури трансформерної нейронної мережі [19]

GPT – це архітектура, що була опублікована у 2023 році й здобула популярність завдяки своїй потужності та масштабованості. Моделі цієї архітектури використовують так званий механізм уваги, який дозволяє:

- враховувати порядок слів;
- розташування відносно одне одного;
- реалізувати «пам’ять»;
- враховувати взаємовідношення слів.

Менш популярними є моделі оригінальної архітектури трансформерної нейронної мережі. Ця архітектура, на відміну від GPT, має енкодер, що у свою чергу аналізує вхідну послідовність, тим самим виокремлює складніші за текст патерни. Ця архітектура не здобула такої популярності, як GPT, адже не є оптимальною з точки зору відношення необхідних ресурсів до точності.

Усі вказані моделі відрізняються тільки розміром, навчальною вибіркою та провайдером. Провайдер це компанія, що розробила модель та надає її до користування. Найпопулярнішими провайдерами на даний момент є OpenAI, Ollama, Anthropic, Azure OpenAI, Groq, Google Gemini, Amazon Bedrock, Google Vertex, Mistral, Hugging Face.

1.3 Варіанти ефективного навчального конспекту в цифровому форматі

Існує 2 варіанти ефективного навчального конспекту в цифровому форматі: текстовий та мультимодальний.

Текстовий конспект представляє собою обробку та сумаризацію транскрибованого тексту відео. Це дуже потужний інструмент, який може зекономити вам час й допомогти у навчанні. Мультимодальний конспект поєднує у собі як текстову, так і візуальну інформацію з відео. Це можуть бути слайди презентації, кадри з відео або інша корисна інформація. Другий варіант набагато складніше автоматизувати, адже він більш комплексний і потребує не тільки транскрипції, а й підбору ключових кадрів та розбиття на секції за

часом. На даний момент задача автоматизованого нотування вирішена неповністю й представлена єдиним чином, тобто у вигляді сумаризації транскрибованого тексту без посилання на кадри з відео або слайди з презентації.

Усі рішення на ринку є приблизно однаковими, а якщо й відрізняються, то тільки додатковим функціоналом у вигляді словників знань або чатів для відповідей на питання моделями. Тим не менш, всі вони базуються тільки на транскрибуванні та узагальненні з додаванням евристичних методів, наприклад вищевказаних словників знань або виокремлення ключових фраз. Також зустрічаються рішення з функціоналом визначення тайм-кодів, але вони роблять це у рамках слова або фрази, що є не дуже корисним та неточним.

1.4 Огляд існуючих програмних рішень для створення конспектів

Сьогодні на ринку існує безліч рішень для нотування з різноманітним функціоналом. Усі подібні рішення – це YouTube-плагіни або сайти.

Рішення можуть бути безкоштовними, з обмеженим безкоштовним користуванням або ж повністю платні. Основний функціонал – це транскрипція відео та сумаризація тексту. Іноді зустрічаються рішення, що можуть структурувати інформацію більш цікавими способами, наприклад:

– побудовою карт знань;

– виділенням основних висновків;

– виділенням питань до самоперевірки;

– наданням пояснення до матеріалу за допомогою вбудованої QA системи.

Огляд існуючих наявних програмних рішень наведено в таблиці 1.1.

Таблиця 1.1 – Огляд існуючих програмних рішень

Назва	**Працює з українською мовою**	**Функціональність**	**Чи обов'язкова реєстрація?**	**Моделі, що використовуються**	**Тип продукту**
1	**2**	**3**	**4**	**5**	**6**
Glasp [20]	Ні	узагальнення	Так	Claude, Mistral, Gemini, GPT-4o-mini, GPT-4o, GPT-4o canvas	Плагін
YouLearn [21]	Ні	узагальнення та QA	Так	GPT	Сайт
NoteGPT [22]	Так	розбиття на глави, узагальнення, карти знань	Так	GPT	Сайт
Kagi [23]	Ні	узагальнення	Так	GPT	Плагін
Slider [24]	Так	розбиття на глави, узагальнення, QA	Ні	GPT	Сайт
Mymap [25]	Ні	узагальнення, карти знань, QA	Так	GPT	Сайт
Otio [26]	Так	узагальнення, написання основних висновків	Так	GPT-4o, Claude, DeepSeek	Сайт
Getrecal [27]	Так	узагаль-нення	Так	GPT	Сайт

1.5 Метрики оцінки якості електронного конспекту

Щоб перевірити якість розбиття відео на секції, необхідно реалізувати метрику, яка відображатиме такі аспекти, як кількість точок розриву та співпадіння проміжків.

Щоб врахувати ці фактори, було розроблено метрику, складену з двох компонентів:

– компонент A – це відношення суми різниць початкових точок відрізків часу до тривалості відео. Даний компонент не враховує перший і останній

сегмент відео. Математична модель наведена в формулі 1.1;

– компонент B – це відношення кількості зайвих точок до кількості прогнозованих або спостережуваних ділянок (обирається за принципом максимальної кількості). Математична модель цього компоненту описана в формулі 1.2.

$$A = \frac{\sum_i^{n-2} \Delta_i}{\tilde{L}}, \tag{1.1}$$

де $\tilde{L}$ – тривалість відеоряду без першого та останнього сегменту;

Δ_i – різниця початків розміченого й знайденого відрізків часу, при чому $\sum_i^{n-2} \Delta_i \in [0, \tilde{L})$ та $\sum_i^{n-2} \Delta_i \leftrightarrow \frac{\sum_i^{n-2} \Delta_i}{\tilde{L}} \in [0, 1]$.

$$B = \frac{n_{extra}}{max(\hat{n}, n)}, \tag{1.2}$$

де $\hat{n}$ – кількість спостережуваних ділянок;

n – кількість прогнозованих ділянок;

n_{extra} – кількість зайвих точок, при чому $n_{extra} \in [0, \max(\hat{n}, n))$ та $n_{extra} \in \big[0, \max(\hat{n}, n)\big) \leftrightarrow \frac{n_{extra}}{\max(\hat{n}, n)} \in [0, 1]$.

$$n_{extra} = |\hat{n} - n|, \tag{1.3}$$

де $\hat{n}$ – кількість спостережуваних ділянок;

n – кількість прогнозованих ділянок.

Оскільки різниця між кількістю точок може бути більшою за n або $\hat{n}$, то для обчислення метрики її треба нормалізувати.

Кінцева метрика – це різниця одиниці та зваженої суми обох компонентів. Математична модель кінцевої метрики описана в формулі 1.4.

$$FinalScore = 1 - (0{,}7 \times A + 0.3 \times B). \quad (1.4)$$

На ряду з оцінкою тайм-кодів, задача потребує оцінки точності групування за слайдами та точності екстракції організаційної інформації та домашніх завдань.

Екстракцію текстової інформації можна оцінити через порівняння отриманих результатів із розміченими даними. Порівняння проводитиметься за допомогою великих мовних моделей. Якість групування ж можна оцінити завдяки косинусній близькості зображень.

Для оцінки загальної якості застосунку було залучено групу незалежних експертів. Деталі аналізу буде наведено в підрозділі 3.5.

1.6 Постановка задачі

Таким чином, на сьогодні існує нестача інтегрованих рішень для автоматичного створення конспектів на основі відео, аудіо-контенту та презентацій. Існуючі сервіси мають низку обмежень, а саме, не враховують візуальну інформацію, не виконують синхронізацію з ключовими кадрами, не дозволяють завантажувати супровідні слайди.

У зв’язку з цим виникає потреба у створенні системи, яка забезпечить якісне мультимодальне узагальнення інформації з лекційного матеріалу та постає актуальна задача – створення застосунку, здатного формувати структурований конспект на основі мультимодального аналізу відео, аудіо та презентацій.

Об’єктом роботи є питання створення конспектів лекцій на основі мультимодального контенту.

Метою роботи є розробка застосунку для створення конспектів лекцій на основі аналізу відео, аудіо-контенту та презентації.

Для досягнення цієї мети необхідно вирішити такі завдання:

– провести аналіз сучасного стану питання узагальнення інформації з відео, аудіо та тексту та формування конспектів, зокрема існуючі програмні рішення для створення конспектів;

– оглянути сучасні мультимодальні моделі ШІ та їх можливостей щодо обробки відео, аудіо та текстових даних;

– розробити структуру конспекту на основі презентації, відео та аудіо контенту;

– сформувати набір даних, який міститиме відеофайли, презентації до них та розмітку відео, а саме часові мітки, що відповідають появі слайдів та ключових кадрів;

– розробити метрики щодо оцінки якості створеного конспекту;

– дослідити методи та моделі, розробити алгоритм, щодо обробки аудіо контенту, а саме, проаналізувати моделі пошуку фрагментів з людським мовленням, транскрибування, постобробки транскрибування;

– розробити алгоритм обробки відеоконтенту та прив’язки його до слайдів презентації, оцінити його точність;

– розглянути питання синхронізації тексту та слайдів;

– вирішити питання узагальнення та структурування інформації;

– сформувати загальний алгоритм для створення конспекту;

– розробити архітектуру програмного застосунку для створення конспекту та реалізувати застосунок;

– визначити якість згенерованих конспектів за допомогою експертної оцінки та метрик.

2 ПОБУДОВА АЛГОРИТМУ СТВОРЕННЯ КОНСПЕКТІВ ЛЕКЦІЙ НА ОСНОВІ АНАЛІЗУ ВІДЕО, АУДІОКОНТЕНТУ ТА ПРЕЗЕНТАЦІЇ

2.1 Розробка макету конспекту на основі відео, аудіоконтенту та презентації

Для створення якісного та корисного конспекту лекції необхідно мати продуману, виважену структуру контенту. Макет конспекту складається із чотирьох частин, серед яких можна виділити:

– назву та структуру лекції;

– коротке узагальнення та основні тези лекції;

– третю частину, що складається із секцій в залежності від кількості обробленого контенту. Кожна секція має наступний зміст:

1) назва секції;
2) слайд, що відповідає даній секції;
3) відповідні нотатки;

– частину з питаннями для самоперевірки.

Така структура дозволяє повністю передати сутність лекції, адже розділяє різні типи інформації, а не змішує все в один шматок тексту, що позитивно впливає на точність і на комфорт користувача. Така сегментація не лише покращує логічний потік інформації, але й значно покращує читабельність та користувацький досвід.

Модульний характер цього формату дозволяє учням швидко зрозуміти суть лекції, переглянути ключові висновки та перевірити своє розуміння за допомогою питань для самоперевірки. Цей метод сприяє кращому запам'ятовуванню та робить процес навчання більш захопливим.

Візуальне представлення структури конспекту наведено на рисунку 2.1, де кожен блок ілюструє певний компонент структури лекції.

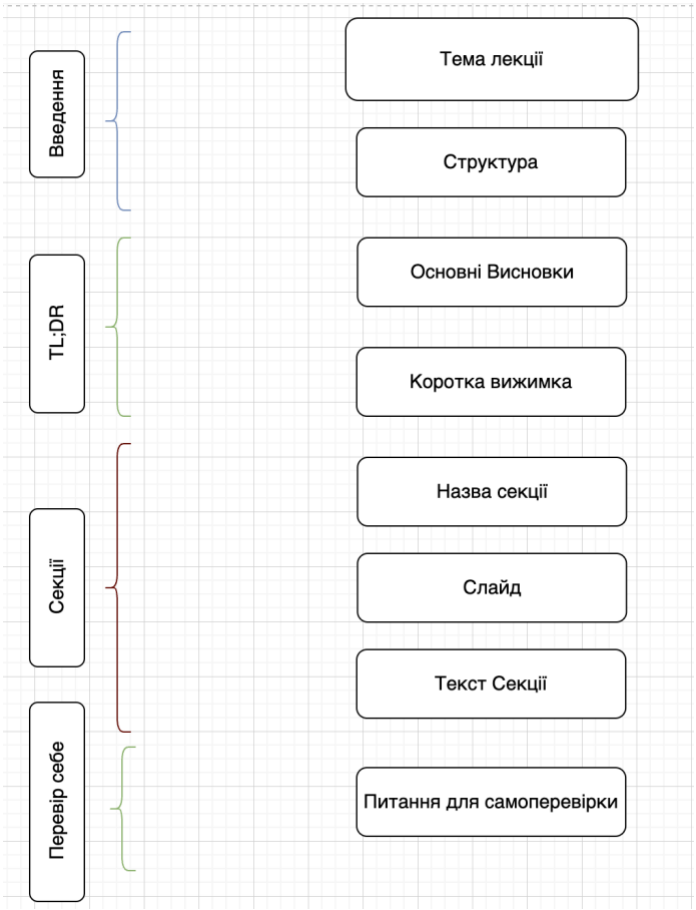

Рисунок 2.1 – Схема макету конспекту

2.2 Обробка аудіоконтенту

Алгоритм обробки аудіоконтенту включає наступні кроки:

Крок 1. Відокремлення аудіо від відео з використанням ffmpeg-python.

Крок 2. Розбиття аудіо на шматки по 30 секунд.

Крок 3. Кодування за допомогою pcm_s16le кодеку із частотою дискретизації 16000 Гц – коливань на секунду.

Крок 4. Збереження у форматі аудіофайлів із представленням у вигляді хвильової форми WAV (Waveform Audio File Format).

Кожен крок є необхідним, бо використані speech-to-text моделі потребують вищезазначених параметрів аудіо та натреновані на вибірках із саме такими характеристиками.

Також треба зазначити, що архітектура вищезгаданих моделей була розроблена, зважаючи на ці параметри.

Як показує практика, якщо надавати на вхід аудіо з більшою частотою дискретизації або більшої довжини, то модель починає галюцинувати, знижується точність, збільшується час генерації. Це призводить до гіршої якості транскрипції.

Саме тому перед тим, як використовувати speech-to-text моделі для транскрипції треба провести обробку аудіоданих та упевнитися, що вони відповідають усім необхідним характеристикам, а саме:

– частота дискретизації 16000 Гц;

– довжина не більше 30 секунд;

– відсутність довгих пауз між мовленням (більш ніж 1 секунди).

Модель Whisper від OpenAI – це великий перетворювач послідовності в послідовність, розроблений для розпізнавання та перекладу мовлення.

Він приймає аудіо на вхід у вигляді логарифмічних спектрограм та обробляє їх через стек шарів згорткової нейронної мережі та трансформаторного енкодера.

Ці блоки кодера витягують значущі ознаки з аудіо, використовуючи механізм уваги для розуміння часових закономірностей.

Вихідні дані енкодера передаються в декодер-трансформер, який крок за кроком передбачає текстові токени.

Whisper навчається одночасно на кількох завданнях, таких як:

– транскрипція мовлення;

– його переклад на англійську мову;

– виявлення відсутності мовлення.

Для цього використовується формат багатозадачного навчання зі спеціальними токенами, які вказують на завдання, мову та часові позначки. Модель підтримує понад 680 000 годин навчальних даних, включаючи різні мови та типи мовлення. Вона може розпізнавати як англійське, так і неанглійське мовлення, перекладати між мовами та ігнорувати немовленнєве

аудіо, таке як музика.

Декодер звертає увагу як на попередньо згенерований текст, так і на закодований аудіо, щоб передбачити наступне слово.

Загалом, Whisper розроблений гнучким, надійним та здатним узагальнювати широкий спектр мовленнєвих та аудіовхідних даних. На рисунку 2.2 вказана схема архітектури описаної моделі.

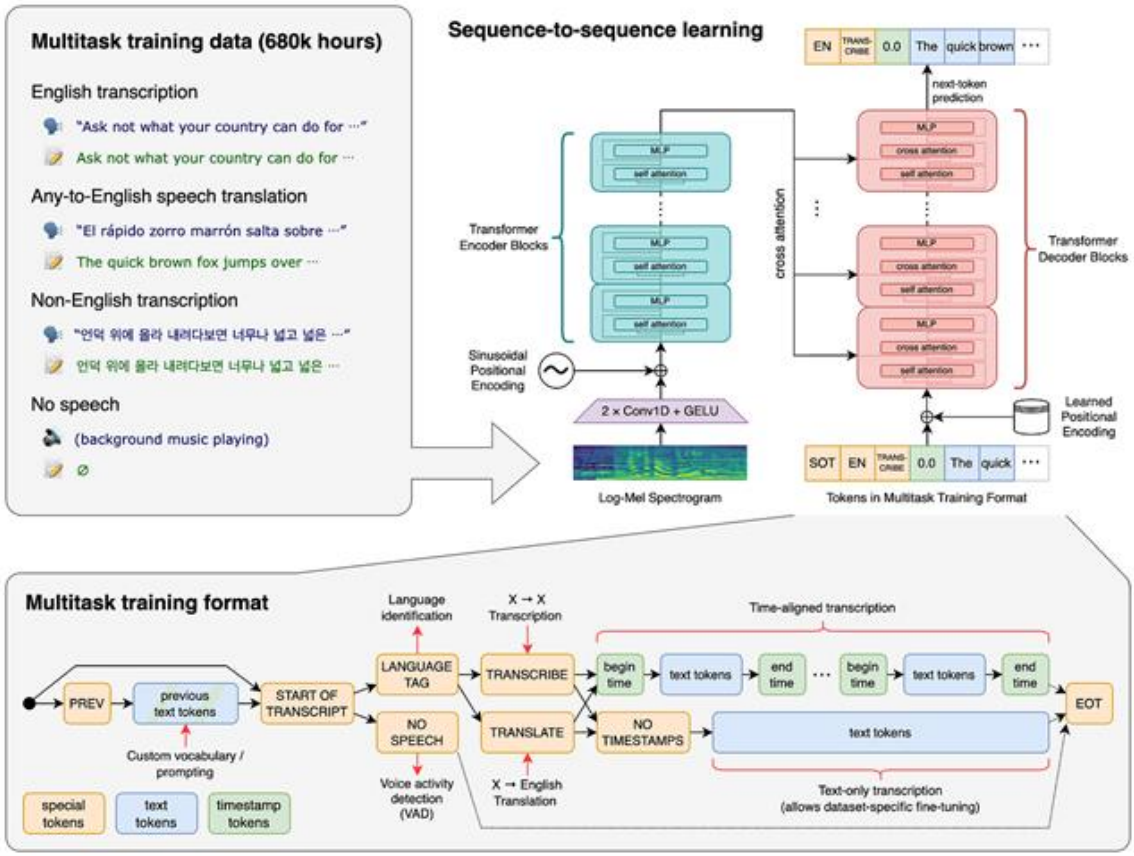

Рисунок 2.2 – Схема архітектури speech-to-text моделі Whisper [28]

2.2.1 Пошук фрагментів з людським мовленням

Також, дуже важливим чинником у задачі транскрибування є наявність мовчазних проміжків. Причиною може бути перерва на лекції, яку записували студенти, після чого не вирізали цей шматок. Також можливим є випадок, коли лектор робить занадто довгі паузи при читанні матеріалу: наприклад, читає

диктант або задає питання з очікуванням відповіді. Усі ці чинники призводять до того, що мовлення постійно обривається. У свою чергу, це шкодить подальшому транскрибуванню.

Для вирішення цієї проблеми існують моделі виявлення мовної активності VAD, що дозволяють знайти та видалити мовчазні проміжки. На рисунку 2.3 зображено архітектуру VAD, яка виявляє сегменти мовлення з необробленого аудіо за допомогою легкого та ефективного алгоритму.

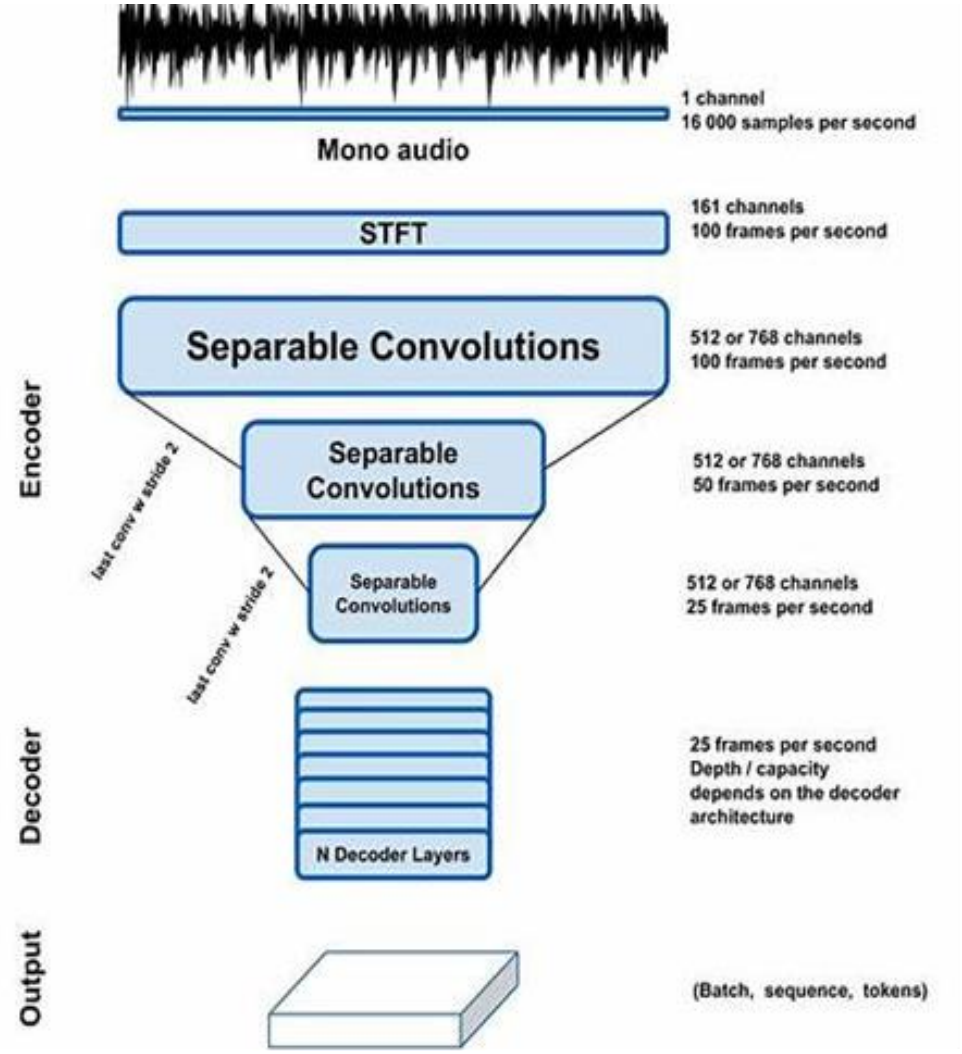

Рисунок 2.3 – Схема архітектури VAD моделі [29]

На вхід модель приймає з монофонічний аудіосигнал, дискретизований з частотою 16000 Гц. Цей сигнал представляє стандартний одноканальний аудіовхід.

Спочатку він перетворюється на часово-частотне представлення за допомогою короткочасного перетворення Фур’є (STFT), створюючи 161 канал зі швидкістю 100 кадрів за секунду. Після чого йде серія роздільних згорткових шарів, які є обчислювально ефективними та виявляють глибші ознаки, зберігаючи часову роздільну здатність.

У міру проходження даних через більше згорткових шарів частота кадрів поступово зменшується вдвічі – спочатку до 50, потім до 25 кадрів за секунду – зберігаючи при цьому велику кількість каналів ознак (512 або 768). Ці шари дозволяють моделі вивчати складні закономірності, які відрізняють мовлення від фонового шуму.

Отримані ознаки потім передаються на кілька шарів декодера, кількість та глибина яких залежать від складності моделі. Декодер працює зі швидкістю 25 кадрів за секунду та інтерпретує карту ознак, щоб класифікувати кожний кадр як мовлення чи відсутність мовлення. Зрештою, вихідний файл являє собою структурований формат, що містить послідовності токенів або міток, що вказують, де на вхідних даних виявлено мовлення.

2.3 Обробка зображень

Для ефективного групування слайдів лекції на основі їхньої візуальної схожості реалізовано двоетапний підхід до обробки зображень. Перший крок включає вилучення ознак з кожного зображення. Ці ознаки слугують основою для побудови надійної метрики подібності, яка може кількісно оцінити, наскільки візуально близькі два слайди один до одного.

Після того, як ознаки зображення визначено, наступним кроком є розробка та реалізація алгоритму, здатного групувати слайди на основі обчислених подібностей. Цей алгоритм групування є фундаментальною частиною системи, оскільки він дозволяє сегментувати лекцію на зв'язні та логічно пов'язані блоки.

2.3.1 Виокремлення ознак з зображень

Найбільш очевидними ознаками зображення є:

– векторне представлення, яке містить у собі інформацію про структуру, розподіл кольорів та інші патерни. Цей вектор ознак можна отримати за допомогою моделі векторного представлення зображень IEM (Image Embedding Model);

– текст на зображенні, який можна зчитати за допомогою моделі оптичного розпізнавання символів OCR (Optical Character Recognition).

Ці ознаки доволі легко та швидко генеруються й повністю описують всі характеристики зображення. На рисунку 2.4 зображено архітектуру моделі PaddleOCR, яка вирішує задачу OCR шляхом вилучення як інформації про макет, так і ключової інформації. На вхід іде зображення, яке може містити текст у різній орієнтації.

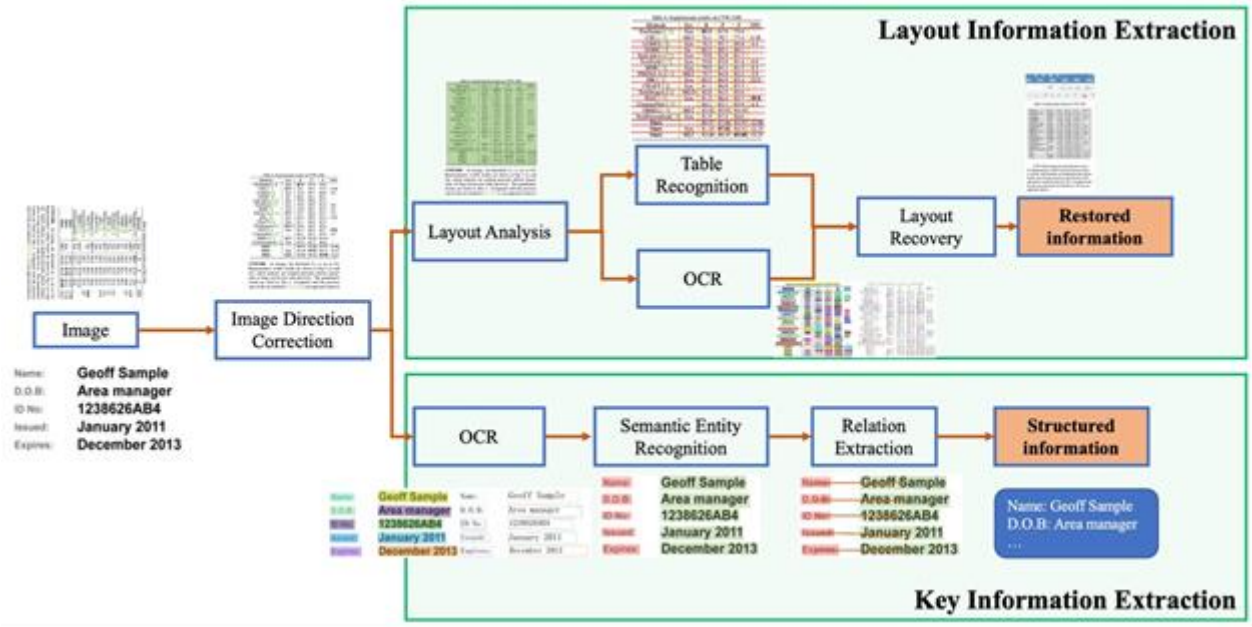

Рисунок 2.4 – Схема архітектури OCR моделі [30]

Спочатку зображення піддається корекції орієнтації. Це є необхідним для правильного вирівняння вмісту для аналізу. Потім картинка проходить дві паралельні гілки:

– вилучення інформації про макет;

– вилучення ключової інформації.

Для розуміння макета зображення виконується аналіз макета та розпізнавання таблиць, де виявляється візуальна структура.

Оптичне розпізнавання символів використовується для вилучення тексту в цих виявлених областях. Потім макет реконструюється для реконструкції оригінальної структури документа. Паралельно OCR також використовується для розпізнавання семантичних сутностей виявляє іменовані поля (наприклад, ім’я, дата народження). Далі вилучення зв’язків пов’язує ці поля з їхніми значеннями для створення семантично змістовної структурованої інформації. Зрештою, PaddleOCR не тільки зчитує текст, але й інтелектуально організовує та інтерпретує його, роблячи його придатним для подальших завдань.

Для зіставлення кадрів зі слайдами треба розробити метрику схожості на основі виявлених ознак зображень. В рамках цієї задачі було обрано зважену суму косинусної подібності векторів зображень та метрикою «перетин над об’єднанням» IOU (Intersection over Union) унікальних слів, зчитаних OCR моделлю.

2.3.2 Дослідження алгоритмів кластеризації та ІЕМ

В рамках дослідження було обрано популярні моделі векторного представлення зображень. Результати порівняння наведено в таблиці 2.1.

Результати показують, що найкраща загальна точність як для часових інтервалів, так і для пошуку кадрів (0,84 та 0,79 відповідно) досягається кількома комбінаціями, зокрема, при використанні DBSCAN з будь-якою з трьох моделей.

Таблиця 2.1 – Порівняння ІЕМ та алгоритмів кластеризації

ІЕМ	**Алгоритм кластеризації**	**Середня точність часових проміжків**	**Середня точність пошуку кадрів**	**Середній час виконання (с)**	**Ціна ($)**
1	**2**	**3**	**4**	**5**	**6**
openai/clip-vit-large-patch14-336	DBSCAN	0,84	0,79	97	0,022
openai/clip-vit-large-patch14-336	HDBSCAN	0,83	0,78	93	0,023
openai/clip-vit-large-patch14	DBSCAN	0,84	0,79	83	0,021
openai/clip-vit-large-patch14	HDBSCAN	0,81	0,78	85	0,023
google/siglip-so400m-patch14-384	DBSCAN	0,84	0,79	94	0,024
google/siglip-so400m-patch14-384	HDBSCAN	0,81	0,78	92	0,023
google/siglip2-so400m-patch14-384	DBSCAN	0,84	0,79	94	0,021
google/siglip2-so400m-patch14-384	HDBSCAN	0,82	0,78	98	0,022

HDBSCAN, як правило, дещо поступається DBSCAN за обома показниками точності у всіх моделях. Час виконання варіюється, найшвидшою моделлю є openai/clip-vit-large-patch14 з DBSCAN, яка завершується за 83 секунди. Щодо ціни, всі комбінації дуже близькі, коливаючись від 0,0011 до 0,0013 доларів США, при чому clip-vit-large-patch14 з використанням DBSCAN є найдешевшою та найефективнішою моделлю. Загалом, DBSCAN видається більш сприятливим як з точки зору якості, та ціни.

На рисунку 2.5 зображено архітектуру моделі CLIP, що складається з текстового енкодера та візуального енкодера, які переводять текст і зображення у спільний векторний простір, де спільні пари «зображення-текст» вирівнюються. Модель навчається придушувати асоціації між нерелевантними парами «зображення-текст», зберігаючи при цьому значущі.

Цей процес дозволяє CLIP удосконалювати свої знання та зменшувати зміщення без повторного навчання з нуля.

SigLIP покращує CLIP, замінюючи втрату контрасту втратою зіставлення пари «зображення-тексту» на основі сигмоїда, що забезпечує кращу ефективність вибірки. У той час як CLIP використовує косинусну подібність та перехресну ентропію для вирівнювання пар «зображення-текст», SigLIP спрощує це за допомогою незалежних прогнозів для кожної пари, зменшуючи складність навчання. В результаті, SigLIP досягає конкурентоспроможних або кращих результатів завдяки більш масштабованому навчанню та покращеній стійкості в сценаріях з нульовим показником.

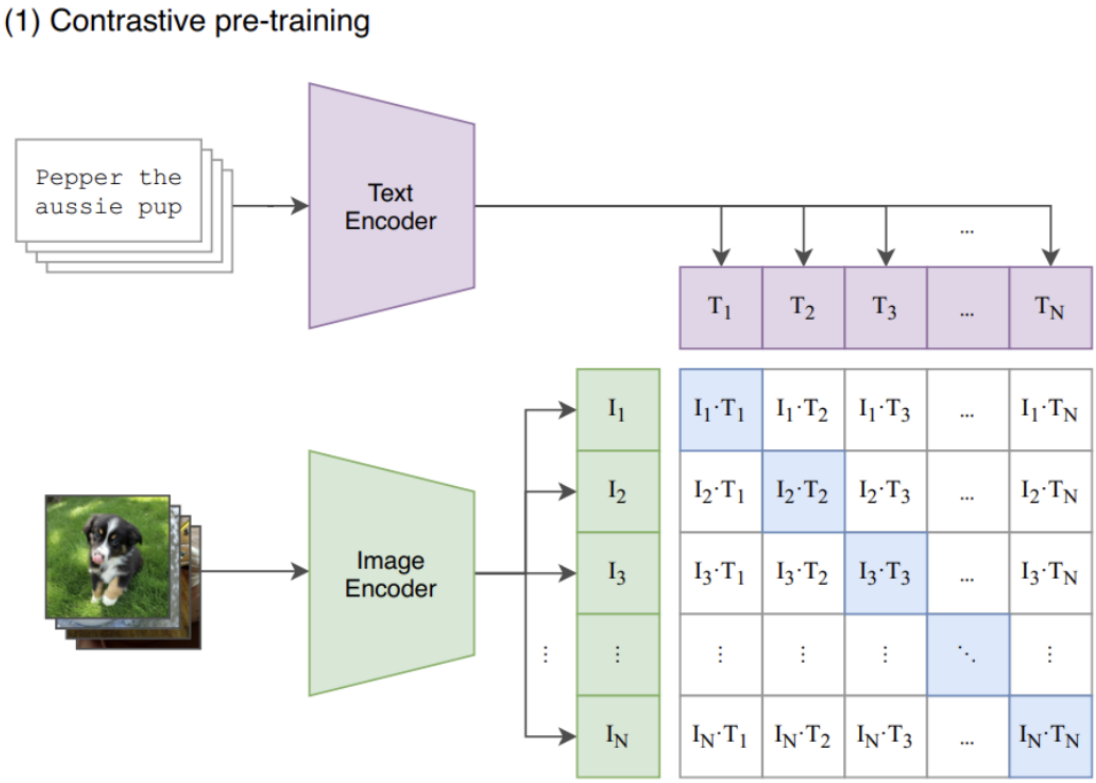

Рисунок 2.5 – Схема архітектури моделі CLIP [31]

На рисунку 2.6 зображено схему роботи DBSCAN – алгоритм неконтрольованої кластеризації, який групує щільно упаковані точки та відокремлює викиди в областях з низькою щільністю.

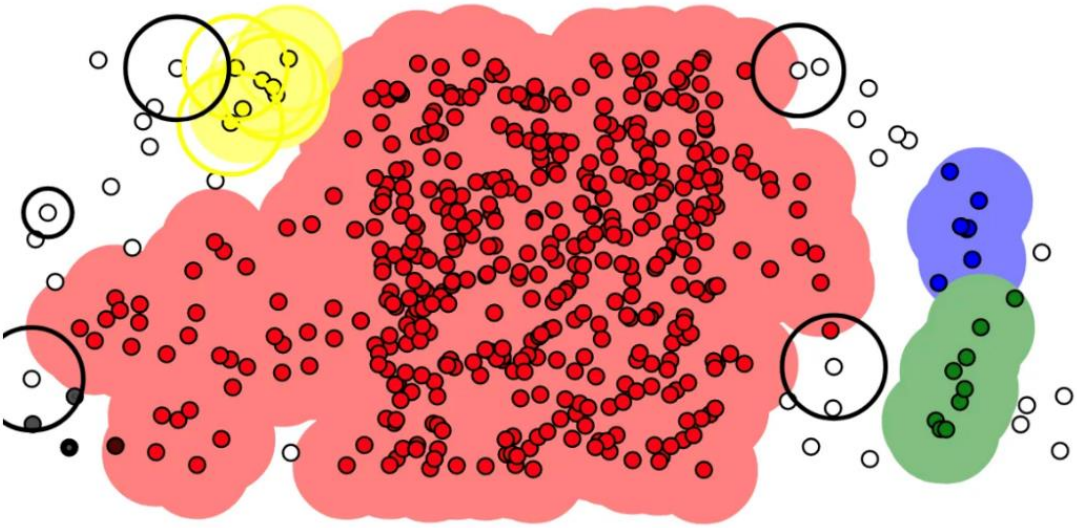

Рисунок 2.6 – Схема роботи алгоритму DBSCAN [32]

Він працює шляхом визначення основних точок – тих, що мають принаймні мінімальну кількість сусідів на певній відстані – та розширює кластери від них. Точки в межах досяжності основної точки додаються до того ж кластера, тоді як ті, що недосяжні, позначаються як шум. На відміну від k-середніх, DBSCAN не вимагає попереднього визначення кількості кластерів і може знаходити кластери довільної форми. Він особливо корисний для наборів даних з шумом або нерівномірною щільністю.

2.4 Алгоритм групування кадрів за належністю до слайду

Алгоритм групування кадрів за належністю до слайду складається з групування, склеювання отриманих груп та розрахунку тривалості груп.

Групування складається з наступних кроків:

Крок 1. Для кожного кадру розраховується подібність до кожного слайду.

Крок 2. За допомогою алгоритму DBCSAN система поділяє значення подібності на кластери.

Крок 3. На основі кластеру за найбільшим середнім значенням подібності розраховується порогове значення.

Крок 4. Слайд з найвищим значенням подібності прикріпляється до кадру якщо від задовільнив порогове значення подібності, що було підібрано автоматично, зважаючи на розподіл подібностей всіх слайдів до даного кадру.

Крок 5. Кадри групуються за належністю до слайду, або, навпаки, за відсутністю слайду. Групування відбувається в історичному порядку, тобто у кожній групі, кадри йдуть один за одним як у відео. Групи поділяються на розмічені та нерозмічені в залежності від того, чи був знайдений слайд для всіх кадрів групи. Згідно з логікою роботи алгоритму, нерозмічені групи – це проміжки між розміченими. На рисунку 2.7 вказаний приклад нерозміченої групи. Тепер, коли застосунок має колекцію, що містить вищезазначені групи в історичному порядку, можна приступати до постобробки цих груп. На даному етапі, це лише поєднання груп за належністю до слайду. Ця операція потрібна згідно з роботою алгоритму й попереджує помилки у групуванні.


```
[
 {"group": [
   {"id'":725, 'slide':40],
   {"id'":726, 'slide':40],
   {"id'":727, 'slide':40],
   {"id'":728, 'slide':40]],
   llabel:'asssigned',
   start_time: 2160,
   end_time: 2172
 },
 {"group": [
   {"id'":725, 'slide':None),
   {"id'":726, 'slide':15,
   {"id'":727, 'slide':45,
   {"id'":728, 'slide':21]],
   label:'unassigned',
   start_time: 2172,
   end_time: 2184
 },
 {"group": [
   {"id'":725, 'slide':41],
   {"id'":726, 'slide':41],
   {"id'":727, 'slide':41],
   {"id'":728, 'slide':41]],
   label:'assigned',
   start_time: 2184,
   end_time: 2196
]
```


Рисунок 2.7 – Приклад нерозміченої групи

Крок 6. Згладжування помилок у групах. Для всіх нерозмічених груп проводиться обчислення розподілу значень знайдених слайді. Якщо

нерозмічена група містить більше трьох елементів та більше 80 % належать до одного й того ж слайду, то вся група розмічається цим слайдом. Завдяки цій обробці алгоритм має деяку ступінь свободи та стає «м'якшим», більш генералізуючим. У свою чергу, це підвищує точність згладжує помилки класифікації.

Так як після згладжування помилок у групах було створено нові групи, то алгоритм повинен знову провести постобробку за наступним алгоритмом:

Крок 1. Склеюються групи, що належать до 1 слайду.

Крок 2. Якщо нерозмічена група містить лише 1 елемент, то вона поєднується з попередньою.

Крок 3. Якщо нерозмічена група містить від 2 до 10 елементів, то вона розділяється між попередньою й наступною.

На останньому етапі, зважаючи на те, ffmpeg вилучив кадри з відео при FPS=1/3, для кожної групи розраховується її тривалість.

2.5 Узагальнення та структурування інформації

Автоматизація процесу нотування потребує доволі багато роботи за текстом. Для того, щоб якісно узагальнити текст та виокремити важливі тези, треба провести низку експериментів з мовними моделями та запитами для них. Найважливішими чинниками у задачі написання запитів є:

– структурованість – мається на увазі, що запит повинен мати чітку структуру й поділятися на секції:

1) системна команда, що описує роль моделі та специфіку роботи;
2) рекомендації до процесу генерації;
3) список із попередженнями або поясненнями до задачі;
4) список дозволеної та забороненої відповіді;
5) за необхідності, секція з прикладами.
6) місця для заповнення вхідних даних та їхній опис;

– відсутність логічних та орфографічних помилок є дуже важливою, адже наявність останніх може згубно вплинути на розподіл ймовірностей при генерації;

– чітка постанова задачі, тобто один запит – одна чітка, недвозначна задача;

– обов’язковою є наявність інструкції для декомпозиції задачі на множину послідовних кроків. Це може доволі помітно підвищити якість задач, що потребують покрокового аналізу.

Таким чином, якщо виконати всі вищеперелічені умови та правильно підібрати модель, то задача узагальнення тексту вирішується доволі легко. В рамках автоматизації процесу нотування вищезазначені техніки було використано під час наступного:

– виокремлення ключових тез;

– сумаризації тексту;

– написання списку з питань до самоперевірки.

Треба зазначити, що кожна модель потребує експериментів з описаннями, структурою та формою запитів. Отже, треба проводити доволі багато експериментів для того, щоб добитися бажаної якості генерації.

2.6 Загальний алгоритм для створення конспекту

Далі наведено загальний алгоритм для створення конспекту:

Крок 1. Виокремлення кадрів з відео з частотою кадрів 1 кадр кожні 3 секунди та слайдів з презентації у форматі PNG.

Крок 2. Обчислення embedding-векторів для кожного слайду кадру.

Крок 3. Зчитування тексту з кожного зображення за допомогою OCR.

Крок 4. Зіставлення слайдів презентації й кадрів з відео за допомогою зваженої суми косинусної близькості зображень й IOU по зчитаному OCR моделлю тексту.

Крок 5. Засновуючись на зіставленні ми групуємо кадри за належністю до слайду.

Крок 6. Ураховуючи те, що кожен кадр у групі це 3 секунди відео ми можемо розбити відео по групах на секції, де кожна секція містить час початку, час кінця та слайд.

Крок 6. Тепер, маючи колекцію, що містить тайм-коди, ми можемо розбити відео на шматки.

Крок 7. На цьому етапі було отримано колекцію зі шматків відео різної довжини. Щоб speech-to-text модель їх транскрибувала, необхідно наступне:

1) відокремити аудіо від відео й зберегти у форматі WAV;

2) знайти фрагменти мовлення за допомогою VAD;

3) склеїти й обробити фрагменти мовлення й зберегти у форматі WAV.

Крок 8. Маючи колекцію WAV-файлів, можна приступити до транскрипції за наступним алгоритмом:

1) завантажуємо speech-to-text модель;

2) обробляємо дані;

3) розподіляємо аудіо файли на колекції однакового розміру для паралельної обробки моделлю;

4) транскрибуємо кожну колекцію;

5) об’єднуємо транскрибовані файли у групи по тайм-кодах.

Крок 9. Після створення набору груп за тайм-кодами, текстом та слайдом, наступним кроком буде обробка та узагальнення тексту. На цьому етапі з кожної групи треба виокремити наступне:

1) короткі ключові тези;

2) коротку сумаризацію тексту секції;

3) питання для самоперевірки.

Крок 10. Для обробки таких частин, як ключові тези та питання для самоперевірки, необхідно виконати наступне:

1) зібрати в одну колекцію зі всіх секцій;

2) виокремити унікальні;

3) вибрати з них найбільш репрезентативні;
4) відобразити окремо у конспекті.

Крок 11. На даному кроці всі дані є зібраними та обробленими. Отже, наступним кроком є написання конспекту в форматі гіпертекстової розмітки HTML з подальшою конвертацією в PDF-файл. Для виконання цього необхідно прописати шаблони, використовуючи мову розмітки HTML та мову стилізації сторінок *CSS* (каскадні таблиці стилів) із додаванням місць для автоматичного заповнення даних та склеїти всі частини зі збереженням у файл. Для конвертації в PDF використано бібліотеку weasyprint мови Python.

Таким чином, в результаті роботи алгоритму користувач отримує конспект, який містить усе потрібне для засвоєння матеріалу. Додатковою зручною можливістю є розуміння, з якого моменту відео чи презентації було отримано дані для конкретного слайду. Це є можливим, адже кожна секція містить тайм-код та номер слайду, що дає користувачеві змогу відкрити лекцію в цьому часовому проміжку, передивитися, або ж відкрити презентацію на цьому слайді.

Ці властивості та формат роблять конспект набагато більш структурованим й зрозумілішим за звичайний транскрибований текст, який немає візуальної інформації.

3 РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ГЕНЕРУВАННЯ КОНСПЕКТУ

3.1 Вибір та налаштування програмного середовища

Програмне забезпечення було розгорнуто на віддаленому сервері з використанням операційної системи Linux та його дистрибутиву Ubuntu Server версії 24.04. Це одна з найпопулярніших систем серед розробників, що працюють зі складними обчисленнями, зокрема в галузі штучного інтелекту, комп'ютерної візуалізації та обробки великих обсягів даних.

Головною перевагою Ubuntu є її стабільність, активна підтримка багатьох бібліотек, зокрема NVIDIA CUDA версії 12.6, яка є ключовою для реалізації обчислень на графічних процесорах. Це дозволяє значно пришвидшити роботу нейронних мереж та процес обробки даних. Бібліотека CUDA забезпечує доступ до можливостей обчислювальної потужності графічних процесорів серії NVIDIA та дозволяє виконувати паралельні обчислення, що забезпечується набагато швидше, ніж при використанні лише центрального процесора.

У ролі графічного процесора було використано NVIDIA RTX A6000, яка має 48 гігабайт відеопам'яті. A6000 – це графічний прискорювач рівня робочої станції, здатний вмістити найважчі моделі й може обробляти багато паралельних задач без втрати продуктивності. Використання такої карти значно скорочує час обробки даних, що особливо критично під час розробки мультимодальних систем ШІ або під час роботи з великими відеофайлами.

В ролі центрального процесора було обрано Intel Core i9 тринадцятого покоління. Це сучасний багатоядерний процесор з високою тактовою частотою, який дозволяє ефективно обробляти кілька завдань одночасно. Разом із 64 гігабайтами оперативної пам'яті, така конфігурація дозволяє виконувати важкі обчислення не тільки на графічному процесорі, але й на центральному процесорі, а також забезпечує стабільність роботи при тривалому навантаженні.

Завдяки цим характеристикам, сервер підходить як для розробки, так і для експлуатації системи в продуктивному середовищі.

Для написання коду використовувався редактор Visual Studio Code версії 1.99.3. Це сучасне середовище розробки, яке підтримує численні розширення для роботи з Python, форматування коду, автодоповнення, тестування, контроль версій та підключення до віддалених серверів. Завдяки своїй гнучкості та простоті налаштування, Visual Studio Code дозволив оптимізувати процес розробки, зменшити кількість синтаксичних помилок та зробити роботу з кодом зручнішою.

Основною мовою програмування для проєкту був Python версії 3.12. Ця технологія є однією з найпопулярніших у сфері штучного інтелекту, обробки даних та автоматизації. Вона має простий синтаксис з широкою екосистемою бібліотек. У цьому проєкті Python використовується як для написання серверного коду, так і для реалізації аналітичних та дослідницьких компонентів системи.

Серверна частина програми реалізована за допомогою фреймворку Flask версії 3.1.0. Flask – це мінімалістичний веб-фреймворк, який дозволяє швидко створювати API або вебінтерфейси з гнучкою архітектурою. Порівняно з більш складними рішеннями, такими як Django, Flask не надає зайвих доповнень, що не використовуються, причому дозволяє розробнику мати повний контроль над структурою програми.

Для обробки відео- та аудіофайлів було обрано рішення FFmpeg версії 4.2.7. Це інструмент командного рядка, який дозволяє працювати з медіапотоками, конвертувати формати, вирізати фрагменти, об’єднувати файли, кодувати потоки тощо. Важливо підкреслити, що використання найпростішого FFmpeg без бібліотек-обгорток пов’язане з тим, що останнім часом файли часто автоматично кодуються та перекодуються, і це може знижувати швидкість та вносити проблеми з якістю в систему. FFmpeg у чистому вигляді дозволяє повністю контролювати процеси обробки, тим самим слідкувати за доречністю операцій над даними.

Для візуалізації конспекту використовувалися сучасні вебтехнології HTML5 та CSS. HTML5 забезпечував семантичну структуру контенту, а CSS дозволяв задавати базовий зовнішній вигляд, зберігаючи чистоту розмітки. Такий підхід полегшив користувачеві перегляд згенерованого конспекту без використання додаткового клієнтського програмного забезпечення.

У проєкті також використовувалася значна кількість сторонніх бібліотек Python, які були критично важливими для досягнення повної функціональності системи. Вони виконували наступні ролі:

– обробка та аналітика даних (pandas, numpy);

– конвертація між високорівневими форматами (pydub, librosa, moviepy);

– взаємодія з мовними моделями та нейронними мережами (transformers, torch, whisper);

– завантаження даних з Інтернету (yt-dlp, aiohttp);

– асинхронна взаємодія з системними процесами (asyncio, subprocess).

Загальний обраний технологічний стек дозволив побудувати потужну, адаптивну та продуктивну систему, яка ефективно працює з мультимедійними даними, підтримує інтеграцію із сучасними великими мовними моделями та забезпечує високий рівень автоматизації на всіх етапах обробки інформації.

3.2 Загальна архітектура системи

На рисунку 3.1 проілюстровано діаграму роботи застосунку, починаючи від вхідних даних та закінчуючи згенерованим конспектом. Архітектура застосунку поділяється на 6 модулів: обробки зображень, групування кадрів, обробки відеоконтенту, аудіо, тексту та модуль для візуалізації конспекту.

Модуль для обробки зображень представляє собою програму, що виконує 4 основні дії:

– розбиває презентацію на картинки у форматі PDF;

– дістає кадри з відео з частотою в 1 кадр на 3 секунди;

– зчитує текст зі слайдів та кадрів;

– переводить кадри та слайди у векторне представленнями.

Модуль для групування кадрів за належністю до слайдів зображень представляє собою програмне рішення згідно алгоритму, розписаного в підрозділі 2.4.

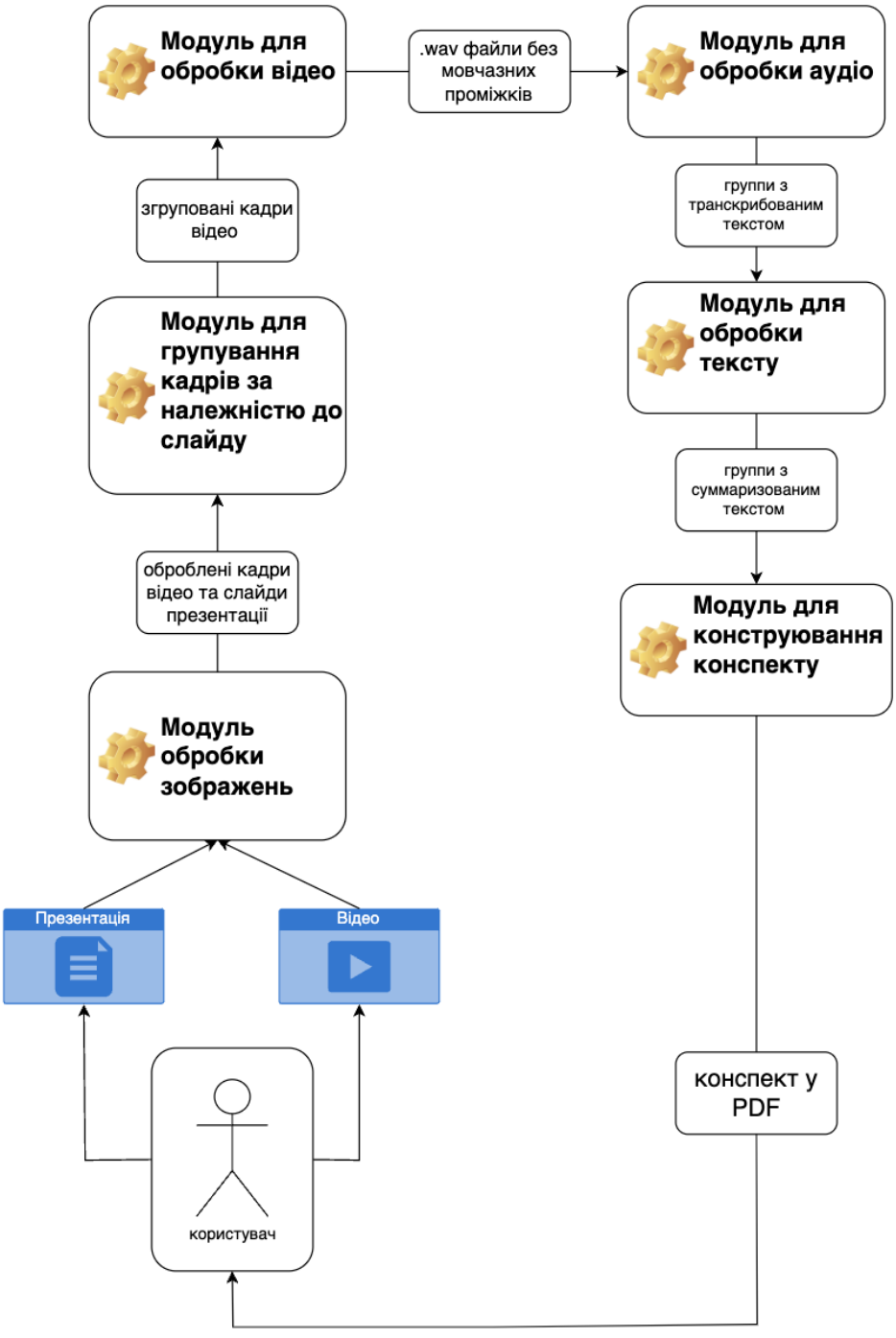

Рисунок 3.1 – Схема архітектури системи

Модуль для обробки відео виконує 4 дії:

– розбиває відео файли базуючись на вищезаначених групах;

– видаляє мовчазні проміжки за допомогою VAD;

– розбиває відео на фрагменти по 30 секунд, щоб Speech-to-text модель мала змогу обробити аудіо;

– виокремлює аудіо з цих відео й зберігає у форматі WAV.

Модуль для обробки аудіо розшифровує текст з WAV аудіофайлів та зберігає його як метадані для кожної групи.

Модуль для обробки тексту розбиває його на наступні категорії:

– основні висновки;

– коротка вижимка;

– декілька секцій з наступним складом:

1) текст секції;

2) основні висновки;

– перевір себе;

Модуль для генерації конспекту оформлює всю інформацію як HTML-файл й конвертує його в PDF, зберігаючи готовий конспект до системної папки «Завантаження».

3.3 Підготовка набору даних із відео та презентацій до них

Для формування повноцінного й різноманітного набору даних, який би відповідав завданням тестування застосунку, було підібрано набір відео, що охоплюють різну тематику.

Кожне з відео супроводжувалося відповідною презентацією, що дозволяє створити цілісний мультимодальний конспект.

З метою розширення та балансування набору даних було прийнято рішення про штучне розбиття деяких повних відео на фрагменти тривалістю від 5 до 40 хвилин. Це дозволяє забезпечити наявність у наборі як

довготривалих, так і коротких відео, що є надзвичайно корисним для оцінки ефективності роботи застосунку в різних умовах. Така стратегія підготовки даних є обґрунтованою з точки зору забезпечення як гнучкості тестування, так і рівномірного розподілу огляду для аналізу.

Усього в результаті обробки та поділу було сформовано 26 окремих відеофайлів. Це є достатнім для валідації, особливо з урахуванням різноманітності тематики та тривалості кожного з елементів.

Для реалізації розмітки – ключового з точки зору створення якісного супровідного текстового або структурованого матеріалу – було вирішено залучити незалежних експертів. Це дозволяє мінімізувати упередженість та забезпечити об’єктивність отриманої розмітки. Також їм було надано чіткий шаблон для заповнення, який ілюструється на рисунку 3.2. Такий формат дає змогу оцінити якість роботи програми з точки зору користувача і надає розуміння як покращити систему. Цей шаблон дозволяє створити максимально репрезентативний набір даних, збалансований за тривалістю, тематикою й структурою. До того ж, залучення незалежних аннотаторів із чітко визначеним форматом інструкції має значну надійність майбутнього аналізу.

```
video_1 = {
    'code_per_frame': {
        '00:00,00:14': 7,
        '00:14,00:45': 8,
        '00:45,02:27': 9,
        '02:27,06:11': 10,
        '06:11,10:00': 11
    }
}
```

```
video_2 = {
    'code_per_frame': {
        '00:00,02:28': 11,
        '02:28,04:12': 12,
        '04:12,04:59': 13,
        '04:59,09:17': 14,
        '09:17,10:00': 15
    }
}
```

Рисунок 3.2 – Формат розмітки даних

На основі раніше сформованого набору даних стало можливим перейти до етапу оцінки якості роботи системи в завданнях пошуку релевантних слайдів та виділення відповідних часових інтервалів. Дані, отримані в

результаті попередньої розмітки, слугували основою для такого аналізу. Без якісного набору даних оцінка була б складною або значно менш інформативною.

Розрахунок ефективності проводився за допомогою спеціалізованих метрик, які відрізняються різними аспектами роботи системи – точністю, повнотою, якістю синхронізації слайдів з відео тощо. Всі ці метрики детально описані в підрозділі 1.5, що забезпечує прозорість та відтворюваність підходу. Завдяки такому рівню деталізації можна не тільки відстежувати прогрес, але й локалізувати існуючі слабкі місця в алгоритмах або джерелах даних

3.4 Ілюстрація роботи застосунку

На рисунку 3.3 зображена початкова веб-сторінка, що слугує інтерфейсом користувача. Ця сторінка дозволяє завантажувати презентацію у форматі PDF та вказувати URL-адресу відповідного відео на YouTube, що формує основу для системи, яка поєднує візуальні та слухові навчальні матеріали.

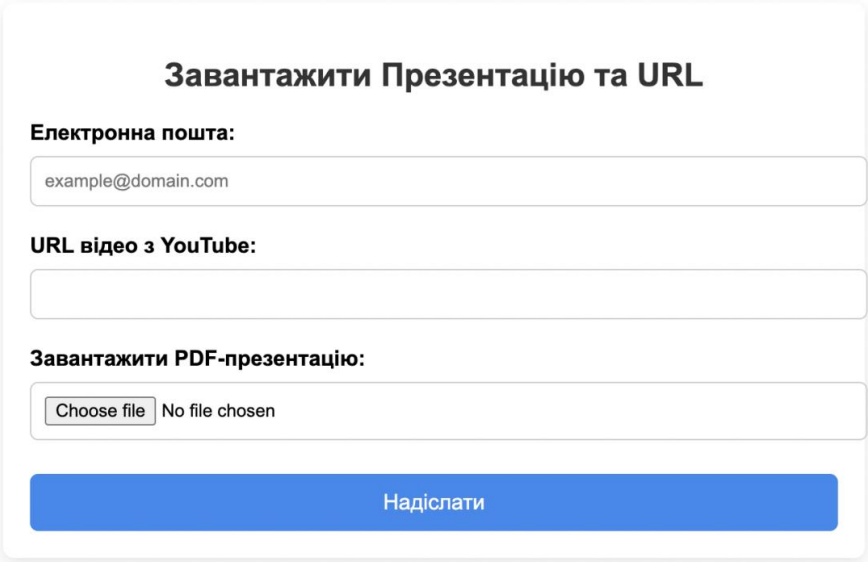

Рисунок 3.3 – Початкове вікно програми

Форма містить поле завантаження файлу та поле введення URL-адреси, що дозволяє структуроване введення, необхідне для автоматизованого аналізу та вирівнювання слайдів із відеосегментами.

Кнопка «Надіслати» запускає процес обробки даних на сервері, після чого користувачу буде надіслано конспект у форматі PDF на пошту.

Рисунок 3.4 демонструє робочий приклад використання програми. Користувач ввів посилання на лекцію з YouTube та вибрав файл презентації під назвою «kauk.pdf» для завантаження, також ввів свою електронну адресу. Після надсилання система використає ці дані для зіставлення слайдів з PDF-файлу з моментами з відео й надішле результат на пошту.

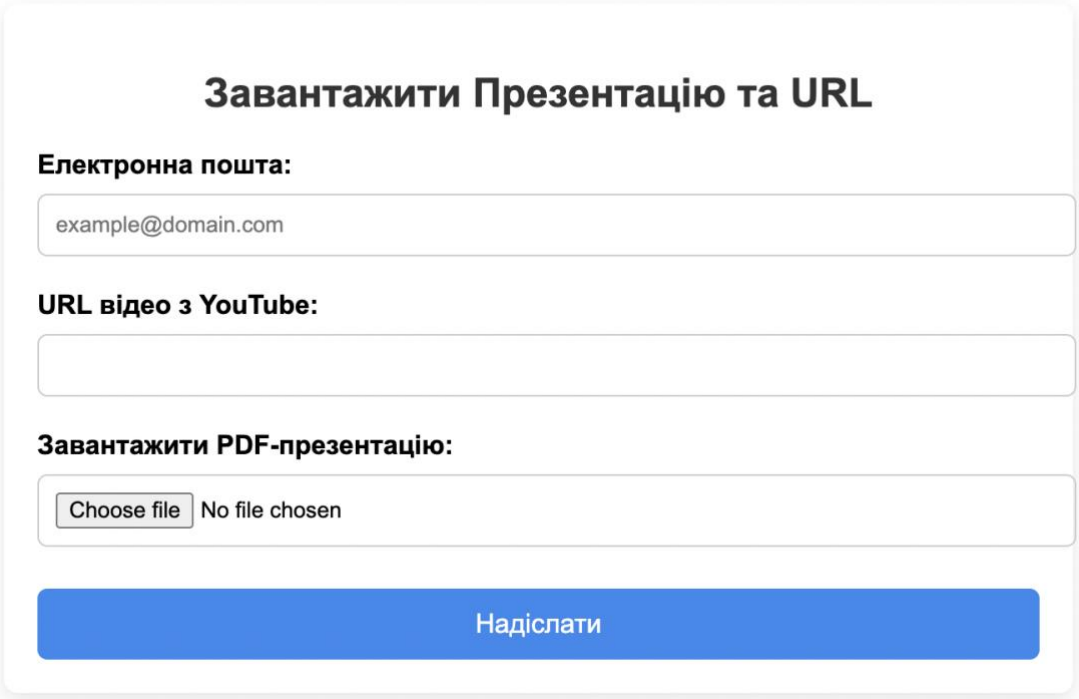

Рисунок 3.4 – Початкове вікно програми з введеними даними

На рисунку 3.5 зображено приклад секції «Введення», що розбиває контент на чітко визначені та обмежені в часі розділи, що дає можливість швидко знайти й переглянути потрібний проміжок в відео заміть того, щоб шукати його. Це також покращує чіткість та планування, оскільки як доповідачі, так і глядачі можуть передбачити хід та тривалість кожної теми.

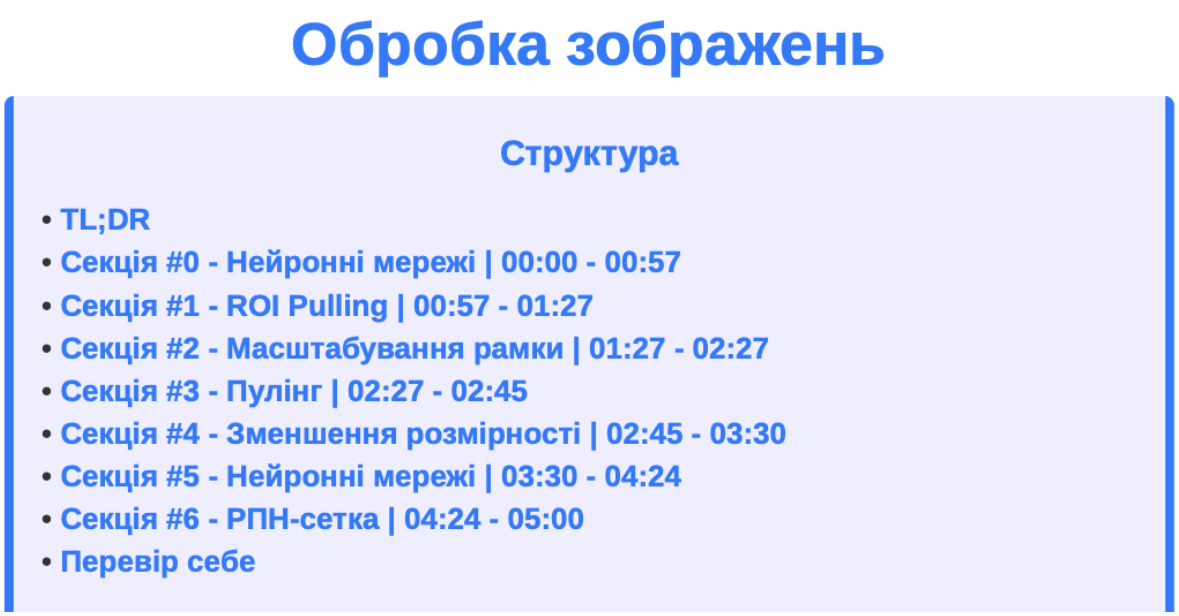

Рисунок 3.5 – Приклад секції «Введення»

Такий формат сприяє узгодженості між кількома сесіями або презентаціями, що полегшує порівняння та тематичну організацію контенту. Для систем, що використовують вирівнювання контенту на основі штучного інтелекту, ця структура забезпечує міцну основу для автоматичного відображення слайдів, аудіо- та відеокадрів. Вона підтримує фрагментацію, що корисно для запам’ятовування та перегляду контенту. Упорядковані та пронумеровані розділи додають структурованості конспекту, зберігаючи гнучкість у глибині теми.

Загалом, така структура секції «Введення» є зручною та зрозумілою для людини, що робить її ідеальною для сучасних, технологічно розширених навчальних середовищ.

На рисунку 3.6 зображено структуру резюме, яка особливо ефективна для узагальнення ключових ідей з презентації чи лекції. Використання розділу TL;DR (Занадто довго; Не читав) на початку одразу надає аудиторії загальний огляд, що є цінним для запам’ятовування та швидкого огляду. Марковані списки використовуються для виділення окремих висновків, що допомагає розбити складну інформацію на легко засвоювані елементи. Така структура

особливо корисна в освітніх та академічних умовах, де важливі ясність та зосередженість.

Крім того, відділення основних висновків «Основні висновки» від резюме «Коротка витримка» дозволяє створити два рівні розуміння: один аналітичний, а інший контекстний. Цей двошаровий формат підтримує різні когнітивні стилі – деякі користувачі отримують користь від стислих фактів, тоді як інші віддають перевагу читанню безперервного тексту. Він також добре піддається автоматизації: маркований список можна розібрати для індексації, тоді як абзац резюме можна використовувати для створення анотацій або описів. Загалом, ця структура посилює як розуміння, так і зручність використання, що робить її ідеальною для документування, архівування контенту та систем знань на основі штучного інтелекту.

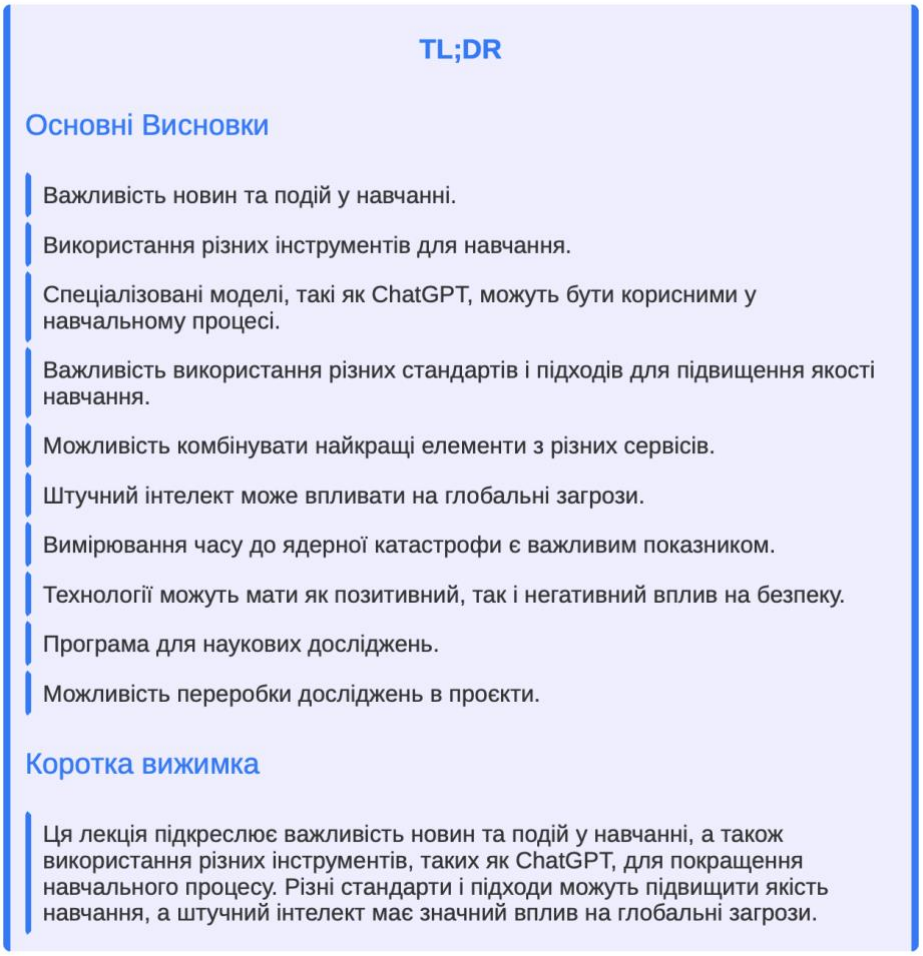

Рисунок 3.6 – Приклад секції «TL;DR»

На рисунку 3.7 зображено секцію конспекту, що дотримується ефективної та педагогічно обґрунтованої структури для представлення окремого розділу лекції. Він починається з чіткого заголовка, який поєднує назву та позначку часу, одразу повідомляючи глядачеві, що охоплює розділ і коли це відбувається на часовій шкалі відео, що є вирішальним для синхронізованого відтворення або модульного перегляду.

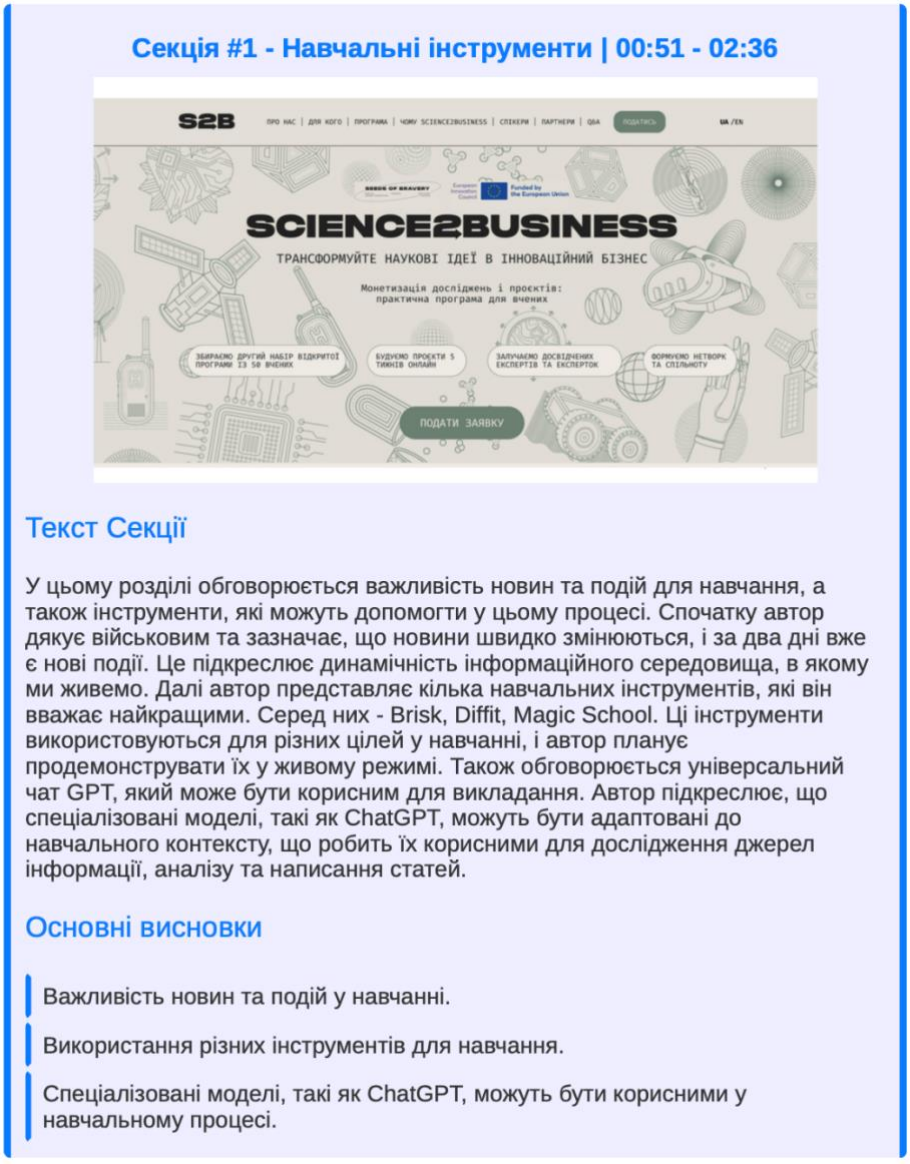

Рисунок 3.7 – Приклад секції конспекту

Розділ містить слайд презентації, що додає візуальний контекст і допомагає прив’язати абстрактні ідеї до реальних застосувань, підвищуючи залученість учнів.

Розділ «Текст розділу» надає детальне пояснення у формі абзаців, ідеально підходить для учнів, які віддають перевагу повному наративному контексту або потребують глибшого розуміння.

Після цього розділ «Основні висновки» зводить основні ідеї до короткого списку, що полегшує повторний перегляд та запам’ятовування ключових висновків. Цей двошаровий підхід – спочатку детальний виклад, потім стислий виклад – враховує різні когнітивні стилі та підтримує як поглиблене вивчення, так і швидке повторення.

З точки зору зручності використання, цей формат підтримує гнучке навчання – студенти можуть переглядати конспекти, заглиблюватися в повні пояснення або використовувати часові позначки для безпосереднього переходу до відповідних відеосегментів. Загалом, це дуже функціональна структура для мультимедійного навчання або архівування знань.

На рисунку 3.8 зображено приклад занадто короткої секції де система вирішила, що виокремлювати тези недоречно й лише перевантажить секцію. Як видно на зображенні – ця секція завдовжки лише в 27 секунд, тож немає сенсу якось додатково розписувати текст.

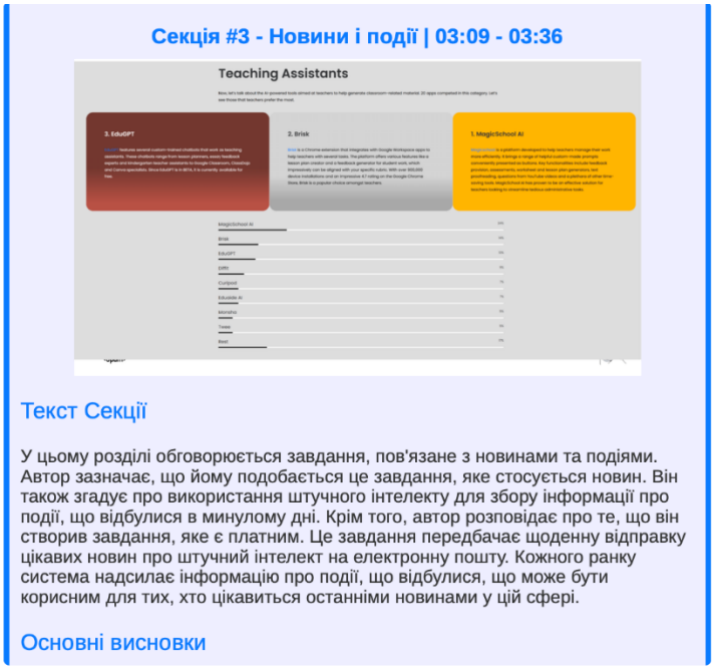

Рисунок 3.8 – Приклад занадто короткої секції конспекту без висновків

На рисунку 3.9 зображено приклад розділу «Перевір себе», який потрібен для самооцінки під назвою й відіграє вирішальну роль у закріпленні результатів навчання. Включення питань для самооцінки в кінці конспекту є ефективним педагогічним інструментом, оскільки воно заохочує учнів активно взаємодіяти з матеріалом, який вони щойно побачили чи почули. Замість пасивного споживання інформації, учнів заохочують згадувати, розмірковувати та застосовувати те, що вони знають, що значно покращує запам’ятовування.

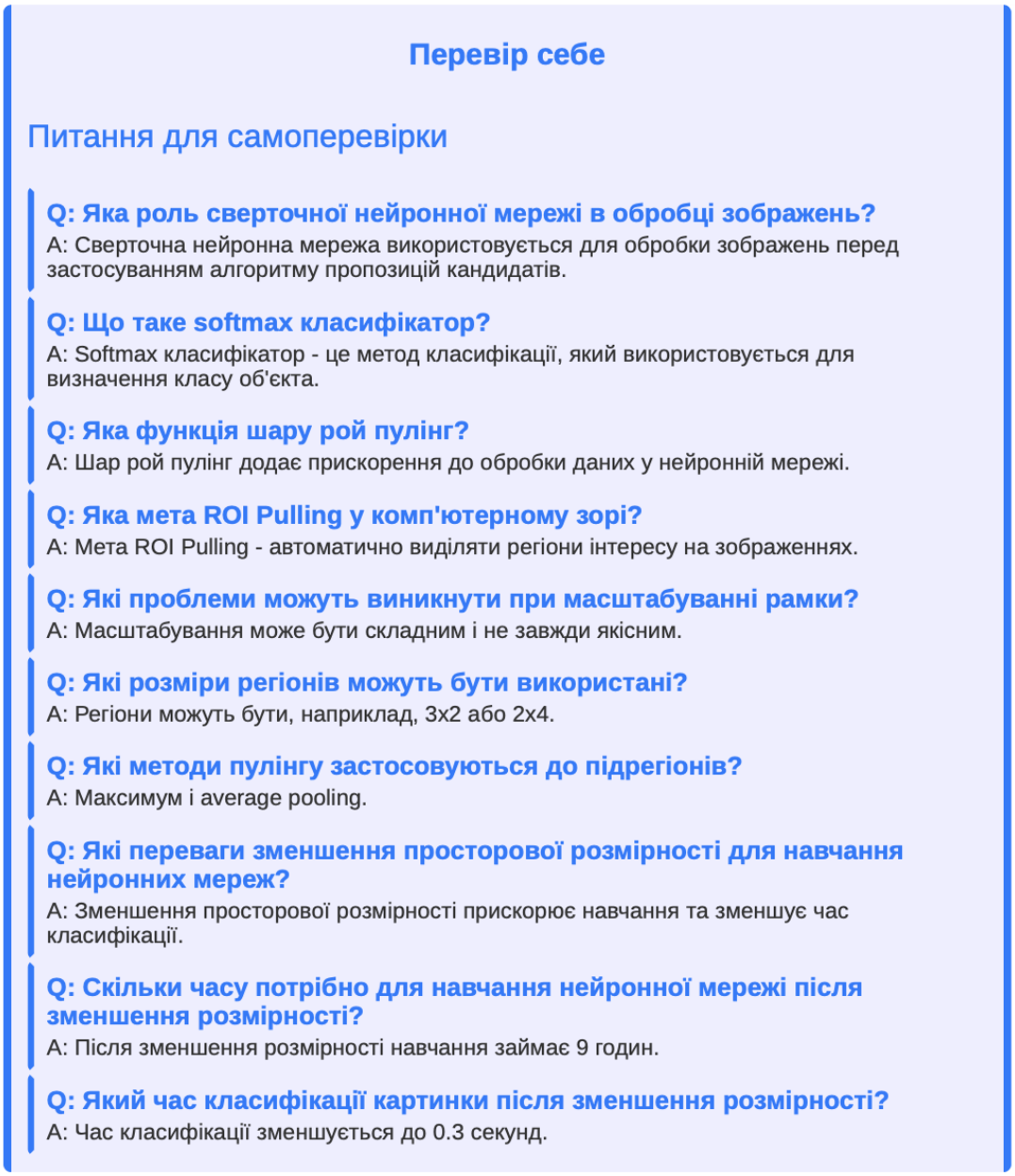

Рисунок 3.9 – Приклад секції «Перевір себе»

Кожне питання тут стосується ключової ідеї уроку, допомагаючи закріпити ключові поняття та забезпечити розуміння. Узгоджуючи питання зі змістом лекції, формат також допомагає учням визначити, що вони могли неправильно зрозуміти або пропустити.

Ця структура підтримує мета-пізнання – учні оцінюють власне розуміння, що важливо для розвитку навичок самостійного навчання. Питання формуються чітко та лаконічно, що робить їх доступними, а також заохочує до вдумливого пригадування.

Включення зразків відповідей під кожним питанням забезпечує негайний зворотний зв’язок, що є критично важливим для закріплення правильного розуміння та виправлення помилкових уявлень.

Це робить формат придатним як для самостійного навчання, так і для керованого навчання. Більше того, коли ці питання інтегровані в цифрові системи, вони можуть підтримувати інтерактивні вікторини або адаптивні функції навчання. Вони також слугують корисним містком до формувального оцінювання, допомагаючи вчителям відстежувати прогрес учнів без формального тестування.

Такі розділи самоперевірки особливо важливі в мультимедійних та асинхронних навчальних середовищах, де негайний зворотний зв’язок від вчителя може бути недоступним. Вони перетворюють статичний урок на інтерактивний навчальний цикл, де учень стає активним учасником. Вони також сприяють запам’ятовуванню, пов’язуючи абстрактні ідеї з прямими питаннями та відповідями.

Цю структуру також можна повторно використовувати для повторення перед іспитами чи оцінюванням. Загалом, це невелике доповнення до уроку з непропорційно великим впливом на якість та глибину навчання.

3.5 Оцінка якості створеного конспекту

Для оцінки якості було проведено тестування, в ході якого було оброблено 26 лекцій, після чого результати зібрано в звідну таблицю з наступними характеристиками:

- назва конспекту;
- тривалість відео лекції, в хвилинах;
- кількість слайдів у презентації;
- кількість слайдів у відео з урахуванням повторних показів;
- кількість знайдених секцій;
- кількість знайдених слайдів з презентації;
- кількість незнайдених слайдів;
- коректність заголовків (лекції та секцій) та змісту секції;
- коректність секції «Основні висновки»;
- коректність секції «Коротка вижимка»;
- коректність тексту в секціях за сенсом;
- відповідність тексту секції та слайду;
- коректність секції «Перевір себе»;
- співпадіння тайм-коду конспекту відео;
- наявність всіх слайдів;
- оцінка того, наскільки конспект дає загальне уявлення про лекцію;
- оцінка того, чи достатньо конспекту для розуміння матеріалу;
- оцінка загального враження від конспекту;
- середня оцінка по п’ятибальній шкалі;
- відсоток загальної оцінки від максимальної;

Характеристики лекцій наведено в таблиці 3.1. Приклади проведених експериментів наведено в таблицях 3.2 – 3.4.

Таблиця 3.1 – Приклади характеристик конспектів

№ п/п	Назва конспекту	Тривалість відео	Кількість слайдів у презентації	Кількість слайдів у відео
1	astrology.pdf	0:24:51	6	7
2	cv2_0.pdf	0:05:00	6	9
3	cv2_1.pdf	0:05:00	2	2
4	cv2_2.pdf	0:05:00	4	4
5	cv2_3.pdf	0:05:00	7	13
6	cv2_4.pdf	0:05:00	5	8
7	cv2_5.pdf	0:05:00	7	7
8	cv1_0.pdf	0:47:05	35	49
9	db_1.pdf	00:10:00	5	5
10	db_2.pdf	00:10:00	5	5
11	kauk_1.pdf	00:05:00	6	6
12	kauk_2.pdf	00:05:00	5	5

Таблиця 3.2 – Приклади кількісних результатів експериментів

№ п/п	Кількість знайдених секцій	Кількість знайдених слайдів з презентації	Кількість незнайдених слайдів	Наявність усіх слайдів
1	**2**	**3**	**4**	**5**
1	7	6	0	5
2	7	6	0	5
3	2	2	0	5
4	4	4	0	5
5	7	6	1	3
6	5	5	0	5
7	7	6	1	3
8	33	19	14	1,5
9	5	5	0	3.5
10	5	5	0	5
11	6	0	6	4,5
12	5	0	5	4

Таблиця 3.3 – Якісні результати експерименту

№ п/п	Коректність заголовків	Коректність секції «Основні висновки»	Коректність секції «Коротка вижимка»	Коректність тексту в секціях	Відповідність тексту секції	Коректність секції «Перевір себе»	Співпадіння тайм-коду конспекту відео
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	3,5	2,5	2	3,5	5	4,5	5
2	5	4	4	5	5	5	5
3	4	4,5	2	5	5	4	5
4	4	4	2	4,9	5	4,8	5

Продовження таблиці 3.3.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
5	2,5	3,8	3,5	5	5	5	5
6	4,8	3	2	4,7	5	3,5	5
7	4,7	3	2	3,5	2,5	4,5	3
8	1,8	3	1	4,2	2,5	2	3,5
9	3	4,5	4.5	5	4	5	5
10	4	5	4.5	5	4,5	5	5
11	3,5	3,5	5	0	4	5	0
12	3,5	4	5	0	4	5	0
…	…	…	…	…	…	…	…
avg	3,98	3,57	**3,05**	4,66	3.64	4,13	4,68

Таблиця 3.4 – Кінцеві оцінки, вирахувані в ході експерименту

№ п/п	**Чи дає конспект загальне уявлення?**	**Чи достатньо конспекту для розуміння матеріалу?**	**Загальне враження**	**Середня оцінка**
1	**2**	**3**	**4**	**5**
1	5	4	4	3,75
2	5	4,8	4,9	4,95
3	5	4,5	4,7	4,35
4	4,9	4,9	5	4,5
5	5	4,7	4,8	3,65
6	4,8	4,7	4,5	4,65
7	3	2,5	3	3,85
8	4	3	3,5	2,65
9	4	3	4	3,75
10	4.5	4	4.5	4,75
11	3,5	4	4	4,25
12	3,5	4	4	4
…	…	…	…	…
avg	4,29	4,01	4,47	4.26

У середньому, найвищим критерієм оцінювання є те, чи забезпечують конспекти загальне розуміння лекції, з високою оцінкою, яка дорівнює 4,01, що свідчить про те, що більшість конспектів успішно передають суть лекції. Аналогічно, узгодженість між конспектами та часовими позначками відео також оцінюється добре (показник дорівнює 4,68), що свідчить про хорошу часову структуру та точне зіставлення між змістом та оригінальним джерелом.

Текстова правильність у розділах також має сприятливі показники, показуючи, що більшість конспектів містять логічно послідовний зміст, який

має сенс незалежно один від одного. Збіг між текстом слайдів та змістом розділів підтверджує ідею про те, що ключова інформація зі слайдів правильно інтерпретується та інтегрується в текст. Оцінки загального враження, середнє значення яких дорівнює 4,47, означають, що хоча конспекти не ідеальні, загалом сприймаються досить позитивно.

Розділи «Перевірте себе» мають помірну оцінку , що свідчить про можливості для покращення допомоги учням у самооцінці. Важливо, що нотатки вважаються достатніми для розуміння матеріалу навіть без перегляду відео, що свідчить про їхню ефективність як окремих документів. Однак середні загальні оцінки та наявність слайдів показують, що іноді трапляються невідповідності або відсутні компоненти.

Примітно, що дві області потребують уваги: розділ «Ключові висновки» та особливо резюме «TL;DR», обидва з яких оцінені значно нижче. Це свідчить про те, що резюме та висновки часто не мають чіткості або повноти, що зменшує їхню корисність. Покращення цих двох елементів, ймовірно, матиме найбільший вплив на загальну задоволеність користувачів та результати навчання.

ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи було досліджено та реалізовано повноцінну систему автоматизованого нотування лекцій, що об’єднує відео, аудіо дані. Основною метою було не лише транскрибувати мовлення, а й розробити структуру, яка дозволяє користувачеві швидко орієнтуватися в навчальному матеріалі.

Було проведено аналіз сучасних моделей штучного інтелекту, зокрема мовних моделей, моделей розпізнавання мовлення, OCR та методів комп’ютерного зору.

Для покращення якості обробки аудіо було використано Voice Activity Detection (VAD), що дозволило видалити мовчазні проміжки й покращити транскрибацію.

Було розроблено спеціальний алгоритм зіставлення кадрів відео зі слайдами на основі векторної подібності та текстового аналізу, що забезпечує точну синхронізацію візуального та мовного контенту.

Кадри групувалися за допомогою алгоритму DBSCAN, що дозволило гнучко адаптуватися до змін структури лекційного відео.

Для структурування інформації були реалізовані техніки написання запитів та сумаризації, що дозволило виокремити ключові тези та питання для самоперевірки.

Було сформовано макет HTML-конспекту, який потім конвертувався у PDF-файл, що робить його доступним і зручним для поширення та архівації. В якості середовища розробки обрано стек технологій з акцентом на Python, Flask, FFmpeg та сучасні моделі з Hugging Face, що забезпечило гнучкість та продуктивність.

Було створено власний набір даних з 26 відео прикладів, які дали змогу якісно протестувати систему. Метрики оцінки якості були розроблені з урахуванням складності задачі мультимодального вирівнювання та узагальнення.

В результаті вдалося створити систему, яка не лише автоматизує процес нотування, а й робить його глибшим, точнішим та більш адаптивним до різних навчальних стилів.

В результаті було доведено точність пошуку кадрів до 84 відсотків точності, у той час як загальна середня оцінка експертів становить 80 відсотків точності, а показник «Загальне враження» досягає 89 відсотків.

Отриманий конспект значно перевершує звичайну транскрипцію, оскільки містить часові позначки, слайди, структуровані тези та питання. Система є масштабованою і може бути адаптована під інші мови або типи контенту.

У подальшому є сенс допрацювати секції «TL;DR» та «Ключові висновки» та «Перевір себе», зробивши їх більш чіткими, стислими та змістовними, оскільки саме вони мають найнижчі оцінки та істотно впливають на користь і сприйняття матеріалу.

Результати роботи було апробовано у вигляді тез доповідей під час 29-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті» [33].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Gorokhovatskyi , O., & Yakovleva , O. (2024). Medoids as a packing of ORB image descriptors. Advanced Information Systems, 8(2), 5–11. . DOI: 10.20998/2522-9052.2024.2.01.

2. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, Indonesian Journal of Electrical Engineering and Computer Science, vol. 33, no. 1, pp. 113-125. DOI: 10.11591/ijeecs.v33.i1.pp113-125

3. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, IEEE Access, vol. 13, pp. 43631-43641, doi: 10.1109/ACCESS.2025.3548910.

4. Yakovleva, O., & Nikolaieva, K. (2020). Research Of Descriptor Based Image Normalization And Comparative Analysis Of SURF, SIFT, BRISK, ORB, KAZE, AKAZE Descriptors. *Advanced Information Systems, 4*(4), 89-101. https://doi.org/10.20998/2522-9052.2020.4.13.

5. Yakovleva O., Nebeský L, Liakhov P. (2023). Research methods of texture image analysis to solve the texture search problem. Proceedings of the IV International Scientific and Practical Conference «The world of modern technologies and inventions». Vienna, Austria. 2023. pp. 252-261 URL: https://isg-konf.com/the-world-of-modern-technologies-and-inventions/.

6. Яковлева, О. В., & Кускова, І. В. (2006). Дослідження результатів сегментації зображень методом матриць збігів. Вісник Національного технічного університету "ХПІ". №39 - С.164 -171.

7. Яковлева, О. В., & Панченко, І. А. (2007). Застосування енергетичних характеристик Лавса для сегментації зображень. Біоніка інтелекту : науково-технічний журнал. №2(67). - С.94-98.

8. Яковлева О.В., Нестерова О.П. (2009) (2009) Порівняльний аналіз методів характеристик Лавса і матриць збігів у задачах сегментації текстурних зображень. Прикладна радіо-електроніка: науч.-техн. журнал, Том 8, №2. - С.181 - 187.

9. Ковтуненко, А. Р., Яковлева, О. В., Любченко, В. А., & Янголенко, О. В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників. Вісник Національного технічного університету ХПІ (3). 24-31 https://doi.org/10.20998/2079-0023.2020.01.05.

10. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. CEUR Workshop Proceedings Vol. 3403. pp. 69-86. ISSN 1613-0073.

11. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, IEEE Access, vol. 12, pp. 73376-73385. DOI 10.1109/ACCESS.2024.3404371.

12. Yakovleva, O., Kovač, M., Ardasov, V. & Yeremenko, I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities. Public Administration and Regional Development, 19(1), pp. 158–183.

13. Yakovleva O., Matúšová S., Tvoroshenko I., Isaiev Y. (2024). Visitor counting based on video stream analysis from surveillance cameras. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 20, no. 1, pp. 67–87. ISSN 1337-2955 URL: https://isg-konf.com/computerintegrated-technologies-of-automation-of-technological-processes/.

14. Yakovleva, O., Matúšová, S., Koshel, V. Implementation of AI approaches in current tools for managing image collections to improve the search capabilities // Proceedings of the IV Correspondence International Scientific and Practical Conference «Science in motion: classic and modern tools and methods in scientific investigations» in Periodical International scientific journal «Grail of science». (February 21, 2025). Vinnytsia, Ukraine - Vienna, Austria. 2025. Vol. 49. P. 752–755. https://doi.org/10.36074/grail-of-science.21.02.2025.096. ISSN 2710–3056.

15. Yakovleva O., Nebeský L., Kirichenko A. (2023). Using the GPT models for responses based on custom content to develop neural consultant for university applicants. Abstracts of V International Scientific and Practical Conference «The world of modern technologies and inventions» Madrid, Spain. Pp. 172-178. URL: https://eu-conf.com/ua/events/trends-in-science-regarding-the-creation-of-new-teaching-methods/

16. Yakovleva Olena, Matúšová Silvia, Táncošová Judita (2024, December 16-18). Investigation of LLMs for generating answers based on user-provided content to support educational and organizational processes. Abstracts of XVI International Scientific and Practical Conference «Modern and new technical trends that help humanity». Thessaloniki, Greece, Pp. 289-295. ISBN – 9-789-40377-568-5 URL: https://eu-conf.com/events/modern-and-new-technical-trends-that-help-humanity/

17. Artificial Intelligence - Our World in Data. URL: https://ourworldindata.org/artificial-intelligence (дата звернення 21.04.2025).

18. Building Custom GPT with PyTorch. A Workshop on Transformers Architecture | by Akriti Upadhyay | Medium. URL: https://medium.com/@akriti.upadhyay/building-custom-gpt-with-pytorch-59e5ba8102d4 (дата звернення 21.04.2025).

19. The Transformer Model - MachineLearningMastery.com. URL: https://machinelearningmastery.com/the-transformer-model/ (дата звернення 21.04.2025).

20. YouTube Summary with ChatGPT & Claude | Glasp. URL: https://glasp.co/youtube-summary (дата звернення 21.04.2025).

21. Let's build GPT: from scratch, in code, spelled out. - Learn, Share, Collaborate. URL: https://app.youlearn.ai/ru/learn/content/kCc8FmEb1nY (дата звернення 21.04.2025).

22. NoteGPT - Your All-in-One AI Learning Assistant. Summarize, Chat & Write – Fast & Free. URL: https://notegpt.io/ (дата звернення 21.04.2025).

23. Universal Summarizer by Kagi. URL: https://kagi.com/summarizer (дата звернення 21.04.2025).

24. Sider: бічна панель ChatGPT + GPT-4o, Claude 3.5 & DeepSeek AI - Веб-магазин Chrome. URL: https://chromewebstore.google.com/detail/sider-бічна-панель-chatgp/difoiogjjojoaoomphldepapgpbgkhkb?hl=uk (дата звернення 21.04.2025).

25. The Easiest Way to Create Diagrams | MyMap AI ю URL: https://www.mymap.ai/ (дата звернення 21.04.2025).

26. Otio - Your personal librarian for the internet. URL: https://app.otio.ai/ (дата звернення 21.04.2025).

27. Recall - Summarize Anything, Forget Nothing. URL: https://getrecall.ai/ (дата звернення 21.04.2025).

28. Automatic Speech Recognition Using OpenAI Whisper without a GPU | by Benjamin Consolvo | Intel Analytics Software | Medium. URL: https://medium.com/intel-analytics-software/automatic-speech-recognition-using-openai-whisper-without-a-gpu-9d316a93860a (дата звернення 21.04.2025).

29. Towards an ImageNet Moment for Speech-to-Text | by Julia Rekamie | Medium. https://juliarekamie.medium.com/towards-an-imagenet-moment-for-speech-to-text-c284d0a269b8 (дата звернення 21.04.2025).

30. Overview - PaddleOCR Documentation. URL: https://paddlepaddle.github.io/PaddleOCR/main/en/ppstructure/overview.html (дата звернення 21.04.2025).

31. The architecture of CLIPErase. Given a CLIP model and a forget set,... | Download Scientific Diagram. URL: https://www.researchgate.net/figure/The-architecture-of-CLIPErase-Given-a-CLIP-model-and-a-forget-set-CLIPErase-has-three_fig1_385442952 (дата звернення 01.05.2025).

32. Clustering Like a Pro: A Beginner’s Guide to DBSCAN | by Sachinsoni | Medium. URL: https://medium.com/@sachinsoni600517/clustering-like-a-pro-a-beginners-guide-to-dbscan-6c8274c362c4 (дата звернення 03.05.2025).

33. Максімов, Г. Р. (2025). Розгляд питання створення конспекту лекцій на основі відео та презентації. *Радіоелектроніка і молодь у XXI столітті: Тези доповідей 29-го Міжнародного молодіжного форуму* (Харків, 16–19 квітня 2025 р.) (Т. 7, с. 92–94). Харків: ХНУРЕ.