

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут прикладного системного аналізу

Кафедра математичних методів системного аналізу

До захисту допущено:

Завідувач кафедри

_____ Оксана ТИМОЩУК

« ____ » _____ 2026 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Системний аналіз і управління»

спеціальності 124 «Системний аналіз»

**на тему: «Система автоматизованої мультимодальної класифікації
реklamних креативів на основі аналізу зображень та відео»**

Виконав:

студент IV курсу, групи КА-22

Парненко Ілля Олексійович _____

Керівник:

Професор кафедри ММСА, д.т.н., проф.

Недашківська Надія Іванівна _____

Консультант з економічного розділу:

Доцент кафедри ЕК, к.е.н., доц.

Рощина Надія Василівна _____

Консультант з нормконтролю:

Доцент кафедри ММСА, к.ф.-м.н.

Статкевич Віталій Михайлович _____

Рецензент:

Доцент кафедри СП, к.т.н., доц.

Булах Богдан Вікторович _____

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Рівень вищої освіти – перший (бакалаврський)
 Спеціальність – 124 «Системний аналіз»
 Освітня програма «Системний аналіз і управління»

ЗАТВЕРДЖУЮ
 Завідувач кафедри
 _____ Оксана ТИМОЩУК
 «__» _____ 2026 р.

ЗАВДАННЯ
на дипломну роботу студенту
Парненку Іллі Олексійовичу

1. Тема роботи «Система автоматизованої мультимодальної класифікації рекламних креативів на основі аналізу зображень та відео», керівник роботи Недашківська Надія Іванівна, професор кафедри ММСА, д.т.н., професор, затверджені наказом по університету від «27» травня 2026 р. №1944-с.
2. Термін подання студентом роботи «10» червня 2026 р.
3. Вихідні дані до роботи: вхідні посилання на рекламні креативи, завантажені банерні зображення та відеокреативи, результати попередніх класифікацій.
4. Зміст роботи: дослідження предметної області, побудова математичної моделі, програмна реалізація системи, аналіз результатів роботи, функціонально-вартісний аналіз.
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): функціональна схема системи, UML-діаграма послідовності, приклади інтерфейсу, аналіз результатів роботи системи.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Функціонально-вартісний аналіз	Рощина Н.В., доцент, к.е.н.		

7. Дата видачі завдання 18.02.2026

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми дипломної роботи.	18.02.2026-10.04.2026	виконано
2	Аналіз предметної області та огляд літератури, ознайомлення з ДСТУ 3008:2015.	11.04.2026-20.04.2026	виконано
3	Формалізація вхідних і вихідних даних системи, побудова математичної моделі задачі, робота над другим розділом.	21.04.2026-28.04.2026	виконано
4	Проектування архітектури програмної системи та алгоритмів обробки зображень і відео.	29.04.2026-09.05.2026	виконано
5	Реалізація модулів вивантаження, обробки та класифікації рекламних креативів.	10.05.2026-22.05.2026	виконано
6	Експериментальне оцінювання роботи системи, аналіз метрик та підготовка третього розділу.	22.05.2026-27.05.2026	виконано
7	Оформлення пояснювальної записки, підготовка ілюстративних матеріалів та проходження нормоконтролю.	28.05.2026-03.06.2026	виконано
8	Підготовка презентації та доповіді до захисту.	04.06.2026-09.06.2026	виконано

Студент _____

Ілля ПАРНЕНКО

Керівник _____

Надія НЕДАШКІВСЬКА

РЕФЕРАТ

Дипломна робота складається зі 115 сторінок, містить 27 рисунків, 15 таблиць, 3 додатки та 27 джерел.

РЕКЛАМНІ КРЕАТИВИ, АВТОМАТИЗОВАНА КЛАСИФІКАЦІЯ, МУЛЬТИМОДАЛЬНИЙ АНАЛІЗ, КОМП'ЮТЕРНИЙ ЗІР, ВІЗУАЛЬНО-МОВНІ МОДЕЛІ, АНАЛІЗ ЗОБРАЖЕНЬ І ВІДЕО, ПОШУК ДУБЛІКАТІВ ЗОБРАЖЕНЬ І ВІДЕО, ПЕРЦЕПТИВНЕ ХЕШУВАННЯ, ВЕКТОРНІ ПРЕДСТАВЛЕННЯ, ГРАФ ПОДІБНОСТІ.

Метою роботи є розробка системи автоматизованої мультимодальної класифікації рекламних креативів для перетворення неструктурованих зображень та відео у структурований звіт із визначеними брендом, продуктом та категорією.

Об'єктом дослідження є автоматизована обробка рекламних креативів, а саме зображень та відео.

Предметом дослідження є математичні моделі, алгоритми та програмні засоби мультимодальної класифікації рекламних креативів.

В роботі проведено аналіз предметної області, побудова математичної моделі обробки та класифікації рекламних креативів, проаналізовано результати роботи системи, а саме системні метрики та метрики якості.

Результатом цієї роботи є програмне забезпечення з графічним інтерфейсом, яке написане мовою програмування Python з використанням бібліотек CustomTkinter, OpenAI, PIL, imagehash та cv2.

Шляхом подальшого розвитку предмета дослідження може бути покращення алгоритмів пошуку дублікатів та використання інших моделей для безпосередньої класифікації рекламних матеріалів.

ABSTRACT

Diploma thesis consists of 115 pages, contains 27 figures, 15 tables, 3 appendices, and 27 references.

ADVERTISING CREATIVES, AUTOMATED CLASSIFICATION, MULTIMODAL ANALYSIS, COMPUTER VISION, VISION-LANGUAGE MODELS, IMAGE AND VIDEO ANALYSIS, IMAGE AND VIDEO DUPLICATE DETECTION, PERCEPTUAL HASHING, VECTOR REPRESENTATIONS, SIMILARITY GRAPH.

The aim of this work is to develop an automated multimodal classification system for advertising creatives to transform unstructured images and videos into a structured report specifying the brand, product, and category.

The object of study in this work is the automated processing of advertising creatives, specifically images and videos.

The subject of the study is of the set of mathematical models, algorithms, and software tools for the multimodal classification of advertising creatives.

The work includes an analysis of the subject area, the construction of a mathematical model for processing and classifying advertising creatives, and an evaluation of system performance using system and quality metrics.

The result of this work is a software application with a graphical user interface, developed in Python using CustomTkinter, OpenAI, PIL, imagehash, and cv2 libraries.

Future development can focus on improving duplicate detection algorithms and exploring alternative models for direct classification of advertising materials.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Актуальність автоматизованого аналізу рекламних креативів	10
1.2 Рекламний креатив як мультимедійний об’єкт аналізу	11
1.3 Банерні зображення та відеокреативи як неструктуровані дані	12
1.4 Огляд підходів до автоматичного аналізу рекламних матеріалів	13
1.5 Мультимодальні моделі для аналізу зображень і відео	15
1.6 Методи пошуку схожих і дубльованих рекламних матеріалів	16
1.7 Висновки до розділу 1	18
РОЗДІЛ 2 МАТЕМАТИЧНА МОДЕЛЬ ТА МЕТОДИ АВТОМАТИЗОВАНОЇ МУЛЬТИМОДАЛЬНОЇ КЛАСИФІКАЦІЇ РЕКЛАМНИХ КРЕАТИВІВ.....	19
2.1 Формалізація вхідних і вихідних даних системи.....	19
2.2 Математична модель мультимодальної класифікації	22
2.3 Формалізація пошуку схожих креативів	24
2.3.1 Перцептивне хешування зображень.....	25
2.3.2 Векторні представлення зображень	26
2.3.3 Подання відеокреативів через множину вибраних кадрів.....	27
2.4 Графова модель пошуку дублікатів	28
2.5 Історія класифікацій як механізм зменшення повторних запитів	31
2.6 Операційні критерії якості, часу та вартості роботи системи	33
2.7 Методика експериментального оцінювання результатів класифікації .	34
2.7.1 Оцінювання правильності визначення Brand, Product та Category .	35
2.7.2 Оцінювання індикатора придатності результату класифікації	36
2.8 Висновки до розділу 2	38
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ	39
3.1 Загальна архітектура програмної системи.....	39

3.1.1 Модуль вивантаження рекламних креативів.....	40
3.1.2 Модуль класифікації рекламних креативів	43
3.2 Проєктування логіки роботи системи.....	47
3.2.1 Функціональна схема системи.....	47
3.2.2 UML-діаграма послідовності роботи компонентів.....	49
3.3 Реалізація алгоритмів обробки рекламних креативів.....	51
3.3.1 Алгоритм обробки банерних зображень.....	51
3.3.2 Алгоритм обробки відеокреативів	55
3.3.3 Формування промпта та класифікація через API	57
3.4 Формування звіту, історії класифікації та операційних метрик	59
3.5 Контрольна вибірка та ручне оцінювання результатів класифікації.....	61
3.6 Аналіз результатів експериментального оцінювання роботи системи .	62
3.7 Обмеження реалізованої системи та напрями подальшого розвитку ...	67
3.8 Висновки до розділу 3	68
РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО	
ПРОДУКТУ	70
4.1 Постановка задачі проєктування	70
4.2 Обґрунтування функцій програмного продукту	71
4.3 Обґрунтування системи параметрів програмного продукту	74
4.4 Аналіз експертного оцінювання параметрів	77
4.5 Аналіз рівня якості варіантів реалізації функції.....	81
4.6 Економічний аналіз варіантів розробки програмного продукту	83
4.7 Вибір кращого варіанту ПП техніко-економічного рівня	89
4.8 Висновки до розділу 4	90
ВИСНОВКИ.....	91
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	94
ДОДАТОК А РЕЗУЛЬТАТИ КЛАСИФІКАЦІЇ.....	98
ДОДАТОК Б СЛАЙДИ ПРЕЗЕНТАЦІЇ ДО ЗАХИСТУ	105
ДОДАТОК В ДОВІДКА ПРО ВПРОВАДЖЕННЯ	115

ВСТУП

У сучасному цифровому середовищі рекламний контент є одним з найважливіших інструментів взаємодії компаній із потенційними споживачами, але стає логічним питання: «Як про їх існування взагалі дізнаються користувачі мережі інтернет?». Тут звісно більшість людей одразу відповість словом «реклама», і вони будуть абсолютно праві. На сьогоднішній день реклама є одним з найважливіших аспектів, за допомогою якої компанії можуть знайти потенційних клієнтів. Так, у дослідженні Redel et al. (2024) [1], присвяченому аналізу впливу онлайн-реклами на трафік в інтернеті, було з'ясовано цікаву річ, в проаналізованих наборах даних було близько 18,2% рекламного контенту для приватного датасету, та 31% рекламного контенту для публічного датасету. Це свідчить про те, що реклама займає суттєву частину усього інтернет трафіку.

Однак сам факт великої кількості рекламного контенту в інтернеті ще не дає можливості їх ефективного використання в аналізі. Рекламні креативи зазвичай представлені у вигляді банерів (зображень) та відео, тобто дані є неструктурованими. Часто спеціалізовані сервіси для публікації необхідної для компанії реклами можуть містити лише базову інформацію (наприклад назву рекламного продукту або посилання на сайт), але не надають готової класифікації для аналітиків. У результаті виникає потреба автоматизованого перетворення рекламних матеріалів у структурований вигляд, придатний для подальшого аналізу.

При виборі методів та алгоритмів обробки рекламних матеріалів важливо враховувати той факт, що креатив може не мати деяких ознак необхідних для аналітика. Наприклад, він може не мати в собі логотипу, або назви продукту. Також креативи не можна розділити на деяку конкретну кількість класів, так як ринок постійно збільшується і через це побудована вибірка класів може бути занадто великою та її треба буде постійно

оновлювати, що в разі ускладнює роботу. Тому треба обирати методи класифікації, які не будуть залежати від конкретної кількості класів, а також будуть достатньо швидкими та нересурсомісткими.

Ціллю даної роботи є розробка системи автоматизованої класифікації креативів на основі відео та банерів (зображень), яка надасть можливість перетворювати неструктуровані рекламні матеріали у структурований вигляд звіту, придатний для подальшого аналізу. Результатом роботи системи є визначення для кожного креативу: бренду, продукту та категорії.

Предметом дослідження є математичні моделі, алгоритми та програмні засоби мультимодальної класифікації рекламних креативів за ознаками Brand, Product та Category, а також методи пошуку дублікатів і повторного використання історії класифікацій.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

В цьому розділі буде розглянуто актуальність предмету дослідження, основні терміни та підходи для розуміння задачі автоматизованої класифікації рекламних креативів.

1.1 Актуальність автоматизованого аналізу рекламних креативів

Сучасна цифрова реклама є одним із найбільш поширених видів мультимедійного контенту в мережі інтернет. Рекламні матеріали створюються у великому обсязі, дуже швидко оновлюються і можуть відрізнитися за форматом, семантикою, стилем подання або ж навіть мають різну цільову аудиторію.

Головна проблема з'являється тоді, коли весь цей неймовірно великий обсяг даних треба проаналізувати, щоб зрозуміти як, наприклад, покращити ефект того як реклама впливає на користувача.

Звісно кожна компанія хоче, щоб її продажі були високими, і часто подання реклами є найважливішим фактором для цього. Тут і з'являється необхідність актуалізувати ринок, дивитися на конкурентів та збирати свою велику базу даних рекламних матеріалів. Але ручна обробка рекламних матеріалів занадто трудомістка та може займати багато часу. Це стає ще більш актуальним для компаній, оскільки треба аналізувати не просто окремі приклади, а великі набори рекламних креативів.

Коли компанія бачить, що група аналітиків більшість свого часу витрачають не на реальну аналітику, а на просту класифікацію та структурування великого об'єму неструктурованих даних, вона розуміє, що потрібна система, яка зможе робити цю роботу з мінімальним людським

втручанням та набагато швидше, а також зможе заощадити багато грошей на цьому. Звідси і виникає потреба у створенні системи, яка зможе це все робити.

1.2 Рекламний креатив як мультимедійний об'єкт аналізу

Рекламний креатив є основним елементом, що підлягає обробці, аналізу, та класифікації. Він має візуальну або текстову частину, та в основному націлений на надання деяких послуг, або продаж деяких товарів [2]. У цій роботі досліджуються візуальні креативи, а саме графічні банери (зображення) та відео. Відео – це послідовність зображень (кадрів) які змінюються з часом та мають звукову доріжку.

Основна інформація, яку можна побачити в рекламних матеріалах, та завдяки якій людина розуміє семантичний зміст креативу [3], може бути така:

- 1) текст;
- 2) логотип;
- 3) зображення продукту;
- 4) фон або сцена;
- 5) персонажі або люди;
- 6) стиль.

Звісно так дані не можна назвати структурованим, оскільки вони не мають чіткої структури та попередньо визначеного формату. Кожен креатив може бути унікальним відносно інших навіть в межах одного бренду чи продукту, саме тому ці дані вважаються неструктурованими.

Ще важливо зазначити, що далеко не кожна сутність з вище перерахованих пунктів може знаходитися на рекламному креативі, через це іноді просто неможливо отримати деякі дані для майбутніх досліджень.

Для аналітиків важливо структурувати цю інформацію [4], щоб робити подальший аналіз, тому їх треба категоризувати. Основні дані, які можна отримати майже з кожного креативу, поділяються на 3 сутності:

- 1) продукт – це те, що «продається» (товар або послуга);
- 2) бренд – це те хто «продає» цей продукт (компанія або торгова марка);
- 3) категорія – це сегмент продуктів одного типу, до якого належить продукт, який «продається».

1.3 Банерні зображення та відеокреативи як неструктуровані дані

Рекламні матеріали в основному поділяються на статичні та динамічні креативи. Тому для них треба підбирати абсолютно різні методи та алгоритми обробки та підготовки до класифікації.

Статичні банери є більш легким типом рекламного креативу з погляду технічної обробки, так як вся потрібна інформація знаходиться в одному зображенні. Але навіть у цьому випадку задача класифікації не зводиться лише до розпізнавання тексту або об'єкту, оскільки бренд може бути представлений логотипом, а продукт – візуальним образом без явної назви.

Відеокреатив є більш складним типом, так як він складається з послідовних кадрів, які змінюються з часом. Всього один відеоролик може мати в собі: вступну сцену, демонстрацію продукту, текстовий блок, логотип, а іноді також заклик до дії. Але для обробки відео варто використовувати не весь відео потік, а набір вибраних кадрів [5]. Це пов'язано з тим, що більшість кадрів будуть мати мінімальні зміни відносно найближчих до нього, тому немає сенсу обробляти кожен з них і є доцільним обрати тільки необхідні.

Також важливо пам'ятати, що аудіо доріжка теж може мати в собі багато важливої інформації, але зазвичай візуальний ряд є основою, на якому вже є необхідна інформація [6].

І банери, і відео є неструктурованими даними, оскільки вони не мають наперед визначених сутностей Brand, Product і Category. Ці сутності отримуються в результаті безпосереднього аналізу візуальної частини та тексту, пошуку об'єктів та розумінням семантики між ними та контексту сцени.

1.4 Огляд підходів до автоматичного аналізу рекламних матеріалів

Найпростішим підходом до класифікації рекламних креативів є звісно ручна експертна розмітка. У такому випадку людина переглядає кожен креатив і визначає бренд, продукт та категорію самостійно. Перевагою цього підходу є можливість враховувати увесь контекст, або деякі неявні ознаки та неоднозначні випадки. Однак при великій кількості рекламних матеріалів ручна розмітка стає повільною, дорогою, а втома та повторюваність дій зменшує ефективність роботи. При масштабуванні кількість людських годин та витрат збільшується неймовірно швидко.

Іншим підходом може бути використання OCR (Optical Character Recognition) для виділення тексту з рекламних креативів [7]. Цей метод дозволяє доволі швидко отримати увесь текст з креативу, але не підходить для такої задачі. Як було сказано раніше, статичний банер або динамічне відео має в собі дуже велику кількість візуальних даних, які не є текстом, наприклад, логотип або об'єкт, що являє собою продукт. Також OCR не розпізнає зміст зображення як цілісного рекламного повідомлення, тобто не розуміє семантики. Тому цей підхід часто просто не зможе визначити необхідні сутності, якщо вони чітко не підписані.

Ще одним підходом може бути Object detection [8], він може знаходити конкретні об'єкти на зображенні, що дозволяє покрити випадки коли на креативі є продукт, але він не підписаний. Проте виявлення об'єктів не

дозволяє читати текст, через це втрачається дуже велика кількість інформації. Також важливо знати що, як і OCR, цей підхід не враховує семантики креативу, що не дає можливості виділити конкретні потрібні ознаки.

Наступний дуже популярний підхід є класична CNN (Convolutional Neural Networks) для класифікації зображень [9]. Треба розуміти, що рекламні матеріали не можна поділити на деяку конкретно задану кількість класів, що унеможливорює створення заздалегідь визначеної множини класів. Так як класичний класифікатор зображень, що стосується також і OCR, потребує такої множини для подальшого навчання та роботи, звідси зрозуміло, що цей спосіб не підходить для поточної задачі.

Найбільш перспективний та актуальний підхід для вирішення такого типу задач може бути використання мультимодальних моделей, зокрема, наприклад, VLM (Vision-Language Models) [10]. Вони можуть одночасно враховувати текстові елементи, конкретні об'єкти на зображенні та візуальний зміст креативу. Також VLM не потребує конкретної множини класів, та може приймати текстові інструкції, що полегшує роботу з нею. Тому цей підхід найкраще підійде для класифікації реклами, яка поєднує в собі: зображення, текст, логотип, композицію та контекст [11], але може бути нестабільним у випадках потреби написання назв одних і тих же сутностей однаково [12].

Нижче, у таблиці 1.1, наведено можливості та важливі обмеження вищезгаданих підходів.

Таблиця 1.1 – Можливості та обмеження вищезгаданих підходів.

Підхід	Можливості	Обмеження
Ручна розмітка	висока якість, розуміння контексту	повільна, дорога, не масштабується
OCR	витягує текст	не розуміє логотипи, продукт і сцену
Object Detection	знаходить об'єкти	не бачить семантики та тексту

Продовження таблиці 1.1

Image Classification	класифікує за фіксованими класами	потребує навчальної вибірки і сталого набору класів
Multimodal або VLM	аналізує зображення з текстовою інструкцією	залежить від інструкції, класифікація буває не стабільною

1.5 Мультимодальні моделі для аналізу зображень і відео

Мультимодальні моделі поєднують обробку текстових та візуальних даних. В цьому дослідженні це значить, що модель може отримати зображення або набір кадрів з відео разом з текстовою інструкцією [13]. Після чого модель буде видавати текстовий результат у вигляді таблиці в стилі Markdown, а спеціальна функція за допомогою регулярних виразів шукає цю таблицю в тексті та переводить в структуровані дані. Ці дані збираються у відповідь у вигляді звіту із сутностями Brand, Product та Category.

Текстова інструкція або ж «Промпт» - це конкретне текстове завдання, яке користувач подає моделі. Завдяки цьому MLLM (Multimodal Large Language Model) або ж будь-яка інша модель, яка на вхід може приймати текст, розуміє, чого потребує користувач.

Переваги такого підходу в тому, що рекламний креатив аналізується не тільки як звичайний набір пікселів, тексту або об’єктів, а як цілісна візуально-семантична структура. Це дозволяє розуміти контекст тих чи інших фрагментів зображення, завдяки чому модель може зрозуміти, що належить до відповідних шуканих сутностей. У свою чергу, написавши промпт, можна чітко задати інструкцію до того, як правильно побудувати відповідь, та що написати.

Робота із динамічними рекламними матеріалами трохи відрізняється від статичних, оскільки у відео є ціла послідовність кадрів, які змінюються у часі. В такому випадку треба обрати деяку кількість ключових кадрів, і подати ці кадри на модель та задати промпт з поясненням, що ці кадри належать до одного відео, тому модель може бачити семантику, а пронумерувавши ці кадри числом можна також задати часовий контекст.

Навчання моделі такого масштабу самостійно є не доречним, це займало б дуже багато часу, а виділення серверів та ресурсів під таку модель для швидкої роботи коштувало б компанії неймовірно багато грошей.

Так як ціль даного дослідження побудувати швидку та ефективну систему, а не навчити модель, то для цього найкраще підійде використання спеціалізованих хмарних сервісів. Для реалізації завдання мультимодальної класифікації можна використовувати прикладний програмний сервіс (API) платформи OpenAI, а саме моделі сімейства GPT (Generative Pre-trained Transformer) [14]. Оскільки не всі моделі цього сімейства можуть приймати на вхід зображення, перед початком роботи важливо обрати саме мультимодальну версію, а точніше з підтримкою VLM.

1.6 Методи пошуку схожих і дубльованих рекламних матеріалів

У великих наборах даних, де багато рекламних матеріалів часто трапляються схожі або абсолютно однакові креативи. Зазвичай вони відрізняються розміром, форматом, незначною зміною тону та кольору, якістю, але все одно несуть однаковий зміст. Це пов'язано із тим, що, наприклад, на сайтах можуть використовувати виділені ділянки різних розмірів та форм, і реклама повинна бути добре видна користувачу, тому банери можуть бути різноманітними, але мати одну і ту ж суть [15, 16].

Адаптивність (Resizing) – це дуже розповсюджена методика, наприклад, такі гіганти як Google Ads чи Meta вимагаються від користувача десятки різних пропорцій під рекламний матеріал. При зміні кольору, тону або ж заклику до дій (наприклад «Купуйте!» або «Узяти зараз!») маркетологи часто експериментують для визначення того, який креатив краще вплине на споживача. Зміна формату або якості пояснюється тим, що сайти мають свої ліміти на розміри файлів, тому для одного креативу може створюватися декілька варіантів форматів (наприклад PNG, WebP або GIF) та з різним ступенем стиснення, що і утворює такі «копії».

Обробка великої кількості дубльованих креативів не є доцільним, оскільки йде сильне навантаження на систему, швидкість зменшується, а ціна збільшується. Тому перед класифікацією треба знайти дублікати та мінімізувати витрати на обробку решти унікальних креативів. Після чого результат можна передати назад до відповідних копій, що збереже багато часу.

Для виявлення майже однакових зображень варто використовувати перцептивне хешування або ж рHash [17, 18] – це група алгоритмів, які створюють «цифровий відбиток» зображення. Він базується на тому, як людина сприймає картинку, а не просто на її точних бінарних даних. Є велика різниця між рHash та іншими алгоритмами хешування (наприклад MD5 чи SHA-256), де змінивши всього один піксель, хеш може повністю змінитися. рHash не настільки жорстко змінюється, якщо картинку трохи змінити, наприклад збільшити яскравість, або перевести в чорно-білі кольори, то він зміниться несуттєво. Він допомагає відсіяти очевидні копії на початку роботи системи.

Водночас треба розуміти, що рHash не розуміє складних перетворень, тому зміна розміру може вплинути на нього і креативи будуть вважатися різними. Тоді використовують векторне представлення або Embedding [11] – це вже числовий вектор, який описує зміст зображення. Він вже вміє розуміти семантику креативу, тому може закрити недоліки рHash. Основна проблема минулого методу пошуку схожих зображень була в тому, що якщо там будуть

два однакових об'єкти, наприклад банани, які повернуті різними кутами, але з однаковою назвою зверху, то rHash скаже, що це скоріш за все різні зображення, якщо не враховувати семантику креативу. За допомогою векторного представлення можна «придати сенсу» до зображення за допомогою чисел, що дозволяє шукати неочевидні, на перший погляд, копії.

Використання векторного представлення більш дороге за ресурсами та часом ніж rHash, тому перший підхід відсікає найочевидніші дублікати, а другий вже може обробити ті, що залишилися. Обробка динамічної реклами другим підходом не є доцільним, оскільки відео має багато кадрів, і якщо у вибірці багато таких відео, то час витрачений на аналіз такої кількості кадрів може бути занадто великим. Тож в такому випадку слід обмежитися пошуком ключових кадрів та порівнянням їх між собою за допомогою rHash.

1.7 Висновки до розділу 1

У цьому розділі було розібрано актуальність рекламного креативу як мультимедійного об'єкта для автоматизованого аналізу. Встановлено, що зображення та відео є неструктурованими даними, для яких необхідно знайти сутності Brand, Product та Category, тим самим структурувавши ці дані.

Розглянуто різні підходи для автоматичного аналізу та класифікації креативів, зокрема ручну розмітку, OCR, класичну класифікацію зображень та мультимодальні моделі. Встановлено, що останні найкраще підходять під поточну задачу, оскільки реклама потребує аналізу, логотипів, об'єктів, тексту, сцені та навіть контексту. Також пояснена причина використання мультимодальних моделей з хмарних сервісів, що дозволяє мати доступ до швидкої обробки та не навчати модель самостійно.

Наприкінці розібрано методи пошуку дублікатів рекламних креативів, що дозволяє пришвидшити роботу системи та зменшити витрати.

РОЗДІЛ 2 МАТЕМАТИЧНА МОДЕЛЬ ТА МЕТОДИ АВТОМАТИЗОВАНОЇ МУЛЬТИМОДАЛЬНОЇ КЛАСИФІКАЦІЇ РЕКЛАМНИХ КРЕАТИВІВ

У цьому розділі розглядається основна формалізація задачі через множини, граф подібності, критерії якості та обмеження. Ці теоретичні положення будуть формувати основу для практичної реалізації та її опису, наведеного в третьому розділі.

2.1 Формалізація вхідних і вихідних даних системи

На самому початку роботи вхідними даними є лише посилань на креативи, тобто ще немає безпосередньо завантажених зображень та відео. В роботі використовується зовнішній сервіс моніторингу рекламних матеріалів Gemius AdReal [19], яке і є джерелом цих вхідних посилань на рекламні креативи. Це сервіс, в якому рекламні матеріали можуть бути доступні за URL-адресами встановленого формату «<https://adreal.gemius.com/ua/show/<id>>», де $\langle id \rangle$ позначає ідентифікатор конкретного рекламного матеріалу.

Спочатку треба позначити один рекламний креатив як x_i , тоді уся множина креативів буде позначатися:

$$x_i \in X$$

де x_i – деякий креатив; X – множина креативів.

Нехай одне посилання матиме позначення l_i , тоді множина усіх посилань позначається таким чином:

$$L = \{l_1, l_2, \dots, l_n\} \quad (2.1)$$

Оскільки кожне посилання веде на конкретний креатив, з цього випливає

$$x_i = d(l_i), \quad (2.2)$$

де $d(l_i)$ – деяка функція завантаження креативу за посиланням l_i .

Так як x_i отримується після обробки посилання l_i з множини (2.1) функцією d за формулою (2.2), то логічно зрозуміти, що множина X , до якої належать усі x_i , буде представляти тільки ті креативи, які реально вдалося завантажити функцією вивантаження d

$$X = \{x_i | x_i = d(l_i) | l_i \in L, d(l_i) \neq \emptyset\} \quad (2.3)$$

Ця умова потрібна для випадків коли посилання застаріло та\або не відкривається, так як такі випадки в Gemius AdReal можливі, і тоді вивантаження креативу не представляється можливим.

Крім того множину X варто розділити на дві підмножини:

$$\begin{aligned} &X^{img} \text{ та } X^{vid}, \\ \text{при } X &= X^{img} \cup X^{vid} \text{ та } X^{img} \cap X^{vid} = \emptyset \end{aligned} \quad (2.4)$$

Тоді слід уточнити:

$$\begin{aligned} x_i^{img} &\in X^{img}, \text{ де } x_i^{img} - \text{зображення;} \\ x_i^{vid} &\in X^{vid}, \text{ де } x_i^{vid} - \text{відео.} \end{aligned}$$

Для цієї роботи звукова доріжка відео для класифікації не використовується, але на майбутнє треба враховувати, що вона теж може мати корисну інформацію.

Так як відео це збірка кадрів, які змінюються з часом, то їх можна просто розбити на зображення, тому для кожного відео вводиться позначення:

$$x_i^{vid} = \{f_{i1}, f_{i2}, \dots, f_{iT_i}\}, \quad (2.5)$$

де f_{ij} – конкретний кадр; T_i – кількість кадрів в відео.

Тоді зображення та кадри відео позначаються як x_i^{img} , $f_{ij} \in \mathbb{R}^{H_i \times W_i \times C}$, де розмірність тензорів визначається висотою H_i , шириною W_i та кількістю каналів C . В той же час слід врахувати, що брати всі кадри відео буде недоречним та затратним по ресурсам та часу, тож вводиться позначення, при якому беруться тільки конкретні кадри з відео:

$$V_i = \{f_{i1}, f_{i2}, \dots, f_{ik}\}, \quad k \ll T_i, \quad (2.6)$$

де f_{ij} – конкретний кадр; а k – кількість узятих кадрів.

А для зображення x_i^{img} можна зробити коротке позначення I_i .

Тепер треба описати вихідні дані. Почати слід з того, що отримується після класифікації, для цього треба позначити результат класифікації одного креативу з відповідними сутностями:

$$y_i = (b_i, p_i, c_i), \quad (2.7)$$

де b_i – бренд; p_i – продукт; c_i – категорія.

Тоді множина усіх результатів класифікації рекламних матеріалів буде $y_i \in Y$. З огляду на все залишається сформулювати вже остаточний звіт, який буде мати в собі посилання l_i , деякий статус $s_i \in \{0, 1, 2\}$, який буде позначати як

обробився креатив (0 – не вивантажився; 1 – класифікований; 2 – вивантажився, але не обробився), а також результат класифікації y_i :

$$r_i = (l_i, s_i, y_i) \quad (2.8)$$

Відтак отримується множина результатів обробки $r_i \in R$.

2.2 Математична модель мультимодальної класифікації

Головна проблема в тому коли треба автоматизувати таку систему. Звісно з усім цим добре впорається людина, але на класифікацію десятків тисяч рекламних матеріалів може піти дуже багато часу. Оскільки креативи створюються постійно, звичайна ручна робота може просто не встигати, тому і потрібне автоматизоване рішення.

Але тут з'являється наступна проблема, треба створити таку систему, яка буде розуміти сенс рекламного матеріалу. Текст часто може бути неповним, бренд може визначатися логотипом, а не текстом. Можливо продукт буде показаний без назви, а у відео буде декілька сцен. Насправді дуже багато факторів які треба враховувати, тому, як було сказано в підрозділі 1.4, звичайний OCR (Optical Character Recognition) не підійде, оскільки він може витягти текст, але не завжди зрозуміє, до чого саме він відноситься. В свою чергу Object Detection не підійде теж, оскільки він може знайти об'єкт, але не завжди визначає бренд, продукт та категорію, та не розуміє семантику креативу.

В підрозділі 1.1 пояснено, що на рекламному матеріалі є дуже багато інформації яку треба включати, а також треба врахувати, що не вся інформація там буде корисна. Саме тому одним з найкращих рішень є підхід мультимодальної класифікації, за якою рекламний матеріал аналізується не

просто як набір окремих об'єктів або текстових фрагментів, а як повноцінний цілісний візуально-семантичний об'єкт [9].

Спочатку вводиться позначення деякої функції обробки та класифікації $g(x_i)$, яка бере завантажений креатив, а після обробляє та класифікує якого даючи результат y_i , звідси впливає такий вираз:

$$y_i = g(x_i) \quad (2.9)$$

Також треба врахувати, що відео та зображення мають різні формати, а значить і функції класифікації будуть відрізнятися, тому вводиться уточнення до цієї функції, яке дозволяє відокремити два різні алгоритми обробки в залежності від множини, до якої належить креатив:

$$g(x_i) = \begin{cases} g_{img}(I_i), & \text{якщо } x_i \in X^{img} \\ g_{vid}(V_i), & \text{якщо } x_i \in X^{vid} \end{cases} \quad (2.10)$$

Тобто в результаті є деяка функція обробки та класифікації $g(x_i)$, яка переводить множину X у множину Y .

Також для розуміння, що результат класифікації є успішним треба ввести такий індикатор придатності результату класифікації, який вказує на те чи було отримано достатньо даних від функції g :

$$\delta(y_i) = \begin{cases} 1, & b_i \notin M \text{ або } p_i \notin M \\ 0, & b_i \in M \text{ і } p_i \in M \end{cases}, \quad (2.11)$$

де M – деяка множина значень, яка не задовольняє результат (наприклад $\emptyset$).

Саме такий індикатор обумовлений тим, що найважливішими для аналізу є Brand та Product, в той час як Category є опціональною, але корисною інформацією для подальшого уточнення в аналізі даних.

2.3 Формалізація пошуку схожих креативів

Класифікація креативів може зайняти дуже багато часу та ресурсів на наборі даних з десятків тисяч посилань, тому хорошим рішенням буде пошук схожих рекламних матеріалів, щоб не класифікувати одне і теж саме по декілька разів, оскільки інакше система буде робити зайві запити. А так як на рекламних наборах трапляються однакові креативи, то це може суттєво зберегти ресурси та час на обробку.

Звісно для пошуку схожих креативів треба представити їх у такому вигляді, щоб їх можна було математично порівняти один з одним, наприклад подати у вигляді числових ознак [10]. Саме ці ознаки вже і дозволять обчислити функцію подібності. Нехай ця функція позначається як $sim(x_i, x_j)$, а також задається поріг схожості τ , тоді отримується такий вираз:

$$x_i \sim x_j \Leftrightarrow sim(x_i, x_j) \geq \tau \quad (2.12)$$

Тобто два креативи вважаються схожими, якщо значення функції подібності між ними не менше за заданий поріг. Тепер більш детально можна розібрати 3 способи представлення схожості креативів, а саме:

- 1) pHash – компактний хеш для приблизного порівняння зображень;
- 2) Embeddings – векторні представлення для семантичної подібності;
- 3) Key frames – подання відео через вибрані кадри.

2.3.1 Перцептивне хешування зображень

pHash має в собі декілька основних етапів. Спочатку він перетворює зображення в grayscale та змінює розмір до 32×32 . Далі застосовується DCT (Discrete Cosine Transform) або Дискретне косинусне перетворення – це математичний метод, який перетворює зображення, яке є набором пікселів, у набір частот. Після виділяє низькочастотні компоненти, що дозволяє проігнорувати шуми (дрібні зміни кольору). Вибираються компоненти, а саме часто верхній лівий кут 8×8 , так як там зазвичай основна інформація про зображення.

Далі йде квантування, береться матриця, наприклад 8×8 , рахується середнє значення, а після проходимося по кожному значенню і дивимося чи це значення більше за середнє (тоді змінюється на 1), чи менше (тоді змінюється на 0). В результаті обрахунків отримується послідовність з 0 та 1, це хеш.

В кінці при порівнянні підраховується відстань Геммінга за допомогою операції XOR: якщо вона дорівнює 0; то зображення ідентичні, чим більше число, тим менш вони ідентичні. Знаючи довжину хешу та відстань Геммінга, можна отримати відсоток схожості, якщо він менший за вище заданий поріг τ , то зображення схожі, інакше – не схожі. Головна проблема pHash в тому, що він не знає семантики креативу, він просто шукає схожі патерни.

Тож доцільно ввести деяку функцію, яка отримує хеш:

$$p_i = \phi_p(x_i), \quad (2.13)$$

де ϕ_p – деяка функція перцептивного хешування; p_i – pHash зображення.

Далі рахується відстань Геммінга таким чином:

$$d_H(p_i, p_j) = \sum_{k=1}^m 1(p_{ik} \neq p_{jk}), \quad (2.14)$$

де d_H – відстань Геммінга;

m – довжина хешу (кількість бітів);

$1(p_{ik} \neq p_{jk})$ дорівнює 1, якщо біти різні, і 0, якщо однакові.

Після чого оновлюється вираз подібності для pHash:

$$x_i \sim_p x_j \Leftrightarrow d_H(p_i, p_j) \leq \tau_H, \quad (2.15)$$

де τ_H – допустимий поріг відмінності між хешами.

В результаті можна задати функцію подібності для pHash порівняння:

$$\text{sim}_p(x_i, x_j) = 1 - \frac{d_H(p_i, p_j)}{m} \quad (2.16)$$

2.3.2 Векторні представлення зображень

Embedding вже допомагає знайти схожі за змістом зображення. З точки зору математики модель переводить зображення в набір чисел, ці числа і є вектором, коли зображення схоже по змісту – вектори теж будуть схожими.

Тож вводиться таке позначення:

$$e_i = \phi_e(x_i), e_i \in \mathbb{R}^m, \quad (2.17)$$

де x_i – зображення;

ϕ_e – деяка функція отримання embedding;

e_i – векторне представлення;

m – розмірність вектора.

Для обробки векторних даних для задання функції подібності доцільно використати косинусну подібність, яка показує наскільки два вектори спрямовані в один бік, якщо напрям майже однаковий, то тоді зображення вважаються схожими за змістом. Тобто буде отримано таку функцію подібності:

$$sim_e(x_i, x_j) = \frac{e_i \cdot e_j}{\|e_i\| \|e_j\|}, \quad (2.18)$$

де $e_i \cdot e_j$ – скалярний добуток векторів;

$\|e_i\|$ та $\|e_j\|$ – довжина векторів;

sim_e – косинусна подібність.

2.3.3 Подання відеокреативів через множину вибраних кадрів

Треба розуміти, що відео, якщо не враховувати звукову доріжку, це просто збірка кадрів (зображень), що змінюються з часом. Тому обробка відео в цілому схожа на обробку зображень.

Звісно опрацювання усіх кадрів в відео може зайняти дуже багато часу та ресурсів, тому раніше було зроблено позначку V_i , як збірка кадрів, з якими далі можна працювати. Найчастіше немає сенсу аналізувати кожен кадр, так як відео може просто бути статичним, або повільним, через що абсолютна більшість кадрів майже не будуть змінюватися.

Тому логічним рішенням буде змусити систему обробляти не все відео, а брати лише важливі або рівномірно обрані кадри [20], і працювати з ними як із зображеннями, з урахуванням того, що конкретна збірка кадрів належить до одного-єдиного креативу. В підрозділі 2.1 вже було задано функцію (2.5) а

також створено деяку множину кадрів, яка піде на обробку (2.6). З них отримується така функція подібності:

$$sim_{vid}(x_i, x_j) = \max_{f \in V_i, q \in V_j} sim_p(f, q) \quad (2.19)$$

де f – деякий кадр з першого відео;

q – деякий кадр з другого відео.

Тобто два відео вважаються схожими, якщо серед їхніх вибраних кадрів є хоча б одна достатньо схожа пара.

2.4 Графова модель пошуку дублікатів

Після того як було описано основні методи пошуку подібності, треба перейти до графової моделі пошуку дублікатів. Це система, яка серед тисячі креативів буде шукати подібні й будувати відповідний граф подібності. Звісно у великих наборів креативи часто пов'язані не просто попарно, там може бути 2, 5, 10 дублікатів, чи навіть більше. Суть цієї моделі в тому, щоб групувати креативи не попарно, а множинами дублікатів. Граф дозволить представити зв'язки між багатьма креативами, а також вибирати одного репрезентант з групи дублікатів для подальшої обробки цих рекламних матеріалів.

Почати слід з введення такого позначення:

$$G = (X, E), \quad (2.20)$$

де G – граф подібності, X – множина вершин графа, E – множина ребер.

Оскільки $x_i \in X$, то це означає, що кожен завантажений креатив є окремою вершиною графа. Ребро (подібність) між двома вершинами (креативами) з'являється тоді, і тільки тоді, коли креативи достатньо схожі:

$$E = \{(x_i, x_j) \mid \text{sim}(x_i, x_j) \geq \tau, i \neq j\}, \quad (2.21)$$

де E – множина ребер;
 (x_i, x_j) – пара креативів;
 $\text{sim}(x_i, x_j)$ – значення подібності;
 τ – поріг схожості.

Так як зображення та відео це різний тип даних, пошук дублікатів для них буде теж різним. Одразу слід уточнити що:

$$E = E^{img} \cup E^{vid}$$

Тоді пошук для зображень та відео буде мати такий вигляд за формулою (2.21):

$$\begin{aligned} E^{img} &= \{(x_i, x_j) \mid x_i, x_j \in X^{img}, \text{sim}_{img}(x_i, x_j) \geq \tau_{img}, i \neq j\}, \\ E^{vid} &= \{(x_i, x_j) \mid x_i, x_j \in X^{vid}, \text{sim}_{vid}(x_i, x_j) \geq \tau_{vid}, i \neq j\}, \end{aligned} \quad (2.22)$$

Порівняння зображення з відео не потребується, так як найчастіше це різні рекламні матеріали.

Після того як усі зв'язки проставлені, треба впровадити зв'язні компоненти графа (детальніше про зв'язні компоненти див. [21]):

$$C = \{C_1, C_2, \dots, C_m\}, \quad (2.23)$$

де C – множина всіх компонентів;
 C_k – одна зв'язна компонента;
 m – кількість компонент.

В свою чергу два креативи належать до однієї групи дублікатів, у випадку коли між ними існує шлях у графі подібності, еквівалентність якого можна задати такою формулою:

$$x_i, x_j \in C_k \Leftrightarrow \exists v_0, v_1, \dots, v_r \in X: v_0 = x_i, v_r = x_j, \forall q = 1, \dots, r: \{v_{q-1}, v_q\} \in E,$$

де v_q – вершина графа (деякий креатив з множини X); r – кількість вершин.

Також слід уточнити, що множини різних зв'язних компонент не перетинаються, а об'єднання всіх множин дає множину креативів X :

$$C_a \cap C_b = \emptyset, \quad a \neq b$$

$$\bigcup_{k=1}^m C_k = X \quad (2.24)$$

Після того як були сформовані зв'язні компоненти, далі вибираються від кожної компоненти по одному репрезентанту. Репрезентант – це один обраний креатив із групи дублікатів, який буде йти на подальшу обробку, а після повертати результат на всі його дублікати в групі. Позначається воно так:

$$u_k = rep(C_k), \quad (2.25)$$

де C_k – група схожих креативів;

u_k – репрезентант цієї групи;

rep – правило вибору репрезентанта.

Маючи репрезентанти від кожної групи, яка може мати в собі один або декілька креативів дублікатів, формується множина унікальних креативів:

$$U = \{u_1, u_2, \dots, u_m\}, \quad (2.26)$$

де m – кількість зв'язних компонентів.

Тоді з цього випливає, що множина дублікатів визначається так:

$$D = X \setminus U \quad (2.27)$$

А перенесення результату з репрезентантів на їх дублікати можна позначити таким чином:

$$\forall x_j \in C_k: y_j = y_{u_k}$$

2.5 Історія класифікацій як механізм зменшення повторних запитів

В цьому підрозділі вводиться поняття «історії». Історія попередніх класифікації це доволі простий, але ефективний спосіб зменшити час на обробку креативів шляхом збереження хороших результатів за індикатором $\delta(y_i) = 1$ для майбутнього використання. Почати слід з введення позначення історії як множини пар посилення та результату класифікації:

$$H = \{(l_i, y_j, s_i)\},$$

де l_i – посилення на креатив; y_j – результат класифікації; s_i – статус.

Далі треба ввести функцію пошуку в історії, яка перевірятиме чи є для конкретного посилення на креатив вже готовий результат:

$$h(l_i) = \begin{cases} y_j, & \text{якщо результат для } l_i \text{ знайдено в історії } H \\ \emptyset, & \text{якщо результат для } l_i \text{ відсутній} \end{cases} \quad (2.28)$$

Також слід позначити множину результатів історії:

$$L_{hist} = \{l_i \in L \mid h(l_i) \neq \emptyset\}$$

Важливо зазначити, що кожен креатив при завантаженні має зв'язок:

$$l_i \mapsto x_i = d(l_i)$$

Так як посилання є унікальними і зберігаються в зручній текстовій формі, їх легше тримати в історії, тому кожен креатив має зв'язок із своїм посиланням. Тож коли система звертається до історії з відповідним креативом, вона здійснює пошук саме по посиланню.

Якщо результат в історії знайдено, то він використовується повторно, але коли результату немає, тоді викликається функція класифікації:

$$r_i = \begin{cases} h(l_i), & \text{якщо } h(l_i) \neq \emptyset \\ g(x_i), & \text{якщо } h(l_i) = \emptyset \end{cases} \quad (2.29)$$

Також після кожного циклу класифікації історія оновлюється:

$$H_{new} = H_{old} \cup \{(l_i, y_j) \mid y_j = g(x_i), \delta(y_i) = 1\} \quad (2.30)$$

Основна ціль історії полягає в тому, щоб не повторювати обробку та класифікацію для тих креативів, що вже її успішно проходили, тобто індикатор $\delta(y_i) = 1$. Таким чином, результати класифікації рекламних матеріалів можна завжди узяти з історії окремо при формуванні результату, що може суттєво пришвидшити роботу системи у випадку повторюваних даних в наборах.

2.6 Операційні критерії якості, часу та вартості роботи системи

Для оцінки придатності класифікації вводиться множина результативно класифікованих креативів, що описується як множина креативів, які були взагалі класифіковані у разі збігу їх результату умовам індикатора:

$$X_{valid} = \{x_i \in X_{class} \mid \delta(y_i) = 1\}, \quad (2.31)$$

де X_{class} – множина креативів, для яких система повернула результат класифікації ($X_{class} \subseteq X$);

y_i – результат класифікації;

$\delta(y_i) = 1$ – результат вважається придатним, тобто містить хоча б Brand або Product.

Далі додається метрика частки результативно класифікованих креативів:

$$Q_{valid} = \frac{|X_{valid}|}{|X_{class}|}$$

Ця метрика показує, яку частку класифікованих креативів має змістовний результат, тобто не має придатні Brand або Product.

Частка класифікованих відносно усіх завантажених креативів:

$$Q_{class} = \frac{|X_{class}|}{|X|}$$

Частку унікальних креативів після дедуплікації:

$$Q_{unique} = \frac{|U|}{|X|}$$

Частка виявлених дублікатів:

$$Q_{dup} = \frac{|D|}{|X|}$$

Частка взятих з історії:

$$Q_{hist} = \frac{|L_{hist}|}{|L|}$$

Частка неробочих посилань:

$$Q_{broken} = \frac{|L| - |X|}{|L|}$$

Частка банерів (зображень) та відео серед завантажених:

$$Q_{img} = \frac{|X^{img}|}{|X|}, \quad Q_{vid} = \frac{|X^{vid}|}{|X|}$$

За потреби множини можна поділити на зображення та відео, і порахувати метрики окремо для кожної групи.

2.7 Методика експериментального оцінювання результатів класифікації

Так як система працює із рекламними матеріалами, які не мають звичайної множини класів, то фактична оцінка такої системи за допомогою класичних метрик ускладнюється. Тому експертиза розділена на оцінку якості класифікації для кожної із сутностей Brand, Product та Category, а також оцінку

роботи $\delta(y_i)$ (індикатора придатності класифікації), завдяки якому отримується множина X_{valid} як зазначено в (2.31).

2.7.1 Оцінювання правильності визначення Brand, Product та Category

Нехай система дає результат класифікації рекламних креативів:

$$\hat{y}_i = (\hat{b}_i, \hat{p}_i, \hat{c}_i),$$

де $\hat{b}_i$ – бренд, визначений системою;
 $\hat{p}_i$ – продукт, визначений системою;
 $\hat{c}_i$ – категорія, визначена системою.

А деякий експерт ручною розміткою визначає правильну відповідь:

$$y_i^* = (b_i^*, p_i^*, c_i^*),$$

де b_i^* – правильний бренд за ручною розміткою;
 p_i^* – правильний продукт;
 c_i^* – правильна категорія.

Тоді можна визначити умовні індикатори $q_*(i)$ для кожної із сутностей таким чином:

$$q_b(i) = \begin{cases} 1 & \hat{b}_i = b_i^*, \\ 0 & \hat{b}_i \neq b_i^*. \end{cases}$$

$$q_p(i) = \begin{cases} 1 & \hat{p}_i = p_i^*, \\ 0 & \hat{p}_i \neq p_i^*. \end{cases}$$

$$q_c(i) = \begin{cases} 1 & \hat{c}_i = c_i^*, \\ 0 & \hat{c}_i \neq c_i^*. \end{cases}$$

Після цього вже можна порахувати кількість правильно та неправильно класифікованих сутностей:

$$Q_{brand} = \frac{1}{n} \sum_{i=1}^n q_b(i)$$

$$Q_{product} = \frac{1}{n} \sum_{i=1}^n q_p(i)$$

$$Q_{category} = \frac{1}{n} \sum_{i=1}^n q_c(i)$$

2.7.2 Оцінювання індикатора придатності результату класифікації

Треба зазначити, що сам по собі індикатор не є гарантом якісної класифікації реклами, він визначає лише факт присутності визначених даних для необхідних сутностей. Для того щоб оцінити роботу системи, треба мати вже розмічені дані. Нехай існує деяка множина X_{test} із розміченими даними, тобто дані мають зв'язку посилання та реального вмісту креативу:

$$X_{test} \subseteq U, \quad |X_{test}| = n,$$

$$\hat{z}_i = \delta(y_i),$$

де y_i – результат класифікації креативу;

$\delta(y_i)$ – автоматичний індикатора придатності;

$\hat{z}_i$ – прогноз системи щодо придатності результату.

Тоді після отримання розміченої експертної вибірки можна позначити ручну оцінку як ще один індикатор, який буде дорівнювати 1 у випадку коли

експерт визнає результат класифікації придатним, та 0 у разі не придатності отриманого системою результату:

$$z_i = \begin{cases} 1, & \text{якщо експерт визнав результат придатним,} \\ 0, & \text{якщо експерт визнав результат не придатним.} \end{cases}$$

Таким чином порівнюючи $\hat{z}_i$ та z_i , можна порахувати важливі метрики [22].

1. Accuracy – загальна частка правильних рішень індикатора

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} .$$

2. Precision – наскільки можна довіряти результатам, які індикатор прийняв придатними

$$Precision = \frac{TP}{TP + FP} .$$

3. Recall – яку частку придатних результатів індикатор прийняв

$$Recall = \frac{TP}{TP + FN} .$$

4. F1-score – баланс між Precision та Recall

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} .$$

Ці метрики є ключовими для розуміння реальної придатності класифікації [23, 24]. Також можна отримавши Confusion Matrix, яка буде мати вигляд, як показано на таблиці 2.1. По ній одразу видно кількість правильно та неправильно класифікованих обраних автоматичним індикатором $\delta(y_i)$.

Таблиця 2.1 – Confusion Matrix

	Експерт: придатний	Експерт: непридатний
Індикатор: придатний	True Positive (TP)	False Positive (FP)
Індикатор: непридатний	False Negative (FN)	True Negative (TN)

2.8 Висновки до розділу 2

У другому розділі було формалізовані вхідні (множина посилань L , а після вивантаження множина креативів X) та вихідні дані системи, а саме множина y_i з трьома сутностями (Brand, Product, Category), та результат $r_i \in R$ до якого належать результати класифікації y_i .

Також було описано підхід до мультимодальної класифікації та їх проблеми. Аналогічно розібрано різні способи представлення креативів для пошуку схожих між собою, такі як pHash, embeddings та key frames. Після чого було побудовано графову модель пошуку дублікатів, що допомогло зрозуміти, як саме можна заощадити час та ресурси. Потім було описано систему історії, яка дозволяла не класифікувати креативи декілька разів, а зберігати придатні результати класифікації та перевикористовувати їх в наступних ітераціях.

Нарешті подано основні критерії якості, завдяки яким можна оцінити наскільки добре працює система на всіх етапах. Також описано методику оцінення індикатора придатності класифікації за метриками типу *Accuracy*, *Precision*, *Recall* та *F1*, які використовуватимуться після ручної експертної розмітки для розуміння якості роботи індикатора $\delta(y_i)$. Було пояснено, що індикатор не є гарантом правильної класифікації, але є достатнім критерієм для системи, щоб розуміти, що результат роботи є придатним.

РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ СИСТЕМИ

В цьому розділі деталізується реалізація системи, яка за множиною посилань на рекламні матеріали L формує структурований результат R , а також, як оцінити якість її роботи. Тобто її робота показуватиме те, як дані обробляються, а отримані результати дозволяють отримати висновки щодо ефективності моделей та алгоритмів.

3.1 Загальна архітектура програмної системи

Система складається з двох модулів. Перший модуль відповідає за збір файлів з посилань, без нього класифікатору не буде над чим працювати. Другий модуль відповідає з класифікацію та структурування цих самих вивантажених файлів.

Спочатку було реалізовано модуль вивантаження креативів, який відповідає за обробку посилань l_i , які входять до множини L за формулою (2.1). Ці посилання обробляються за допомогою деякої функції $d(l_i)$ для отримання файлів креативів (зображень та відео) x_i згідно з формулою (2.2). Отримані таким чином файли належать до множини вивантажених рекламних матеріалів X за формулою (2.3). Цей модуль знаходиться у теці «AdrealDownloadCreative».

Після цього було впроваджено модуль класифікації. Він відповідає за обробку та класифікацію файлів креативів x_i множини X . Основна його задача перетворити файл зображення або відео x_i у набір сутностей y_i з (2.7) за допомогою деякої функції $g(x_i)$ за формулою (2.9). Після цього формується

результат r_i множини R , як було зазначено у формулі (2.8). У цього модуля теж є окрема тека «Ads_AI-Classification».

3.1.1 Модуль вивантаження рекламних креативів

Для запуску цього програмного модуля треба зробити підготовку даних. Процес налаштування починається з переходу у відповідну теку модуля «AdrealDownloadCreative» та заповнення файлу «links.txt» посиланнями l_i , де кожне нове посилання починається з наступного рядка. Приклад структури та вміст файлу зазначено на рисунку 3.1:

```
https://adreal.gemius.com/ua/show/<id1>  
https://adreal.gemius.com/ua/show/<id2>  
https://adreal.gemius.com/ua/show/<id3>  
https://adreal.gemius.com/ua/show/<id4>  
https://adreal.gemius.com/ua/show/<id5>  
https://adreal.gemius.com/ua/show/<id6>  
https://adreal.gemius.com/ua/show/<id7>  
https://adreal.gemius.com/ua/show/<id8>  
https://adreal.gemius.com/ua/show/<id9>
```

Рисунок 3.1 – Вміст файлу «links.txt»

Після того як список посилань був вставлений у файл, можна починати їх вивантаження. Для роботи модуля треба запустити файл «RUN.bat» після чого він почне для кожного введенного посилання l_i застосовувати функцію $d(l_i)$, та вивантажувати файли креативів x_i у папку з такою назвою «Creatives_YYYY-MM-DD», де YYYY-MM-DD – це формат часу. Після чого у консоль виводиться результат роботи, як зазначено на рисунку 3.2.

```

00:21:39 INFO | [293] done in 0.15s (saved=True, type=banner)
00:21:39 INFO | [292] done in 0.19s (saved=True, type=banner)
00:21:39 INFO | [294] done in 0.17s (saved=True, type=banner)
00:21:40 INFO | [297] done in 0.15s (saved=True, type=banner)
00:21:40 INFO | [295] done in 0.25s (saved=True, type=banner)
00:21:40 INFO | [299] done in 0.13s (saved=True, type=banner)
00:21:40 INFO | [300] done in 0.26s (saved=True, type=banner)
00:21:40 INFO | Прогрес: 300/400 (75.0%), 30.1s
00:21:40 INFO | [298] done in 0.65s (saved=True, type=video)
00:21:41 INFO | [296] done in 1.31s (saved=True, type=video)
00:21:43 INFO | Прогрес: 350/400 (87.5%), 32.6s
00:21:45 INFO | Прогрес: 400/400 (100.0%), 34.4s
00:21:56 INFO | Готово. Оброблено 400 посилань за 45.8s
00:21:56 INFO | Вихідні файли: Creatives_2026-05-15 | Лог TXT: output.txt
Press any key to continue . . .

```

Рисунок 3.2 – Результат роботи модуля у консолі

Також генеруються три службові файли.

1. «h.txt» – цей файл зберігає посилання l_i , які вже були в історії H , це зроблено для того, щоб скрипт не вивантажував зайві файли, а просто повідомляв наступний модуль про те, що їх треба додати в результат класифікації, взявши з історії вже готовий y_i за формулою (2.28).
2. «output.txt» – цей файл зберігає в собі структуру з 4 стовпця, такі як посилання, статус (позначає чи успішно завантажився файл, 200 – успіх, 404 – посилання не відкрилося), бренд (який одразу дає сервіс Gemius AdReal, але це значення часто не є точним), номер (айді конкретного креативу), та тип вивантаженого файлу. Вміст та структуру цього файлу показано на рисунку 3.3.
3. «runtime_upload.json» – в цьому файлі зберігаються базова інформація по кількості вивантажених файлів, їх типів, та часу вивантаження. Вміст зазначено на рисунку 3.4.

```

link,status,brand,number,type
https://adreal.gemius.com/ua/show/<id1>,200,Smallbusiness,3,banner
https://adreal.gemius.com/ua/show/<id2>,200,Foreignads,2,banner
https://adreal.gemius.com/ua/show/<id3>,200,Smallbusiness,4,video
https://adreal.gemius.com/ua/show/<id4>,404,,,
https://adreal.gemius.com/ua/show/<id5>,404,,,
https://adreal.gemius.com/ua/show/<id6>,200,Foreignads,11,banner
https://adreal.gemius.com/ua/show/<id7>,200,Smallbusiness,12,banner
https://adreal.gemius.com/ua/show/<id8>,404,,,
https://adreal.gemius.com/ua/show/<id9>,200,Smallbusiness,15,banner

```

Рисунок 3.3 – Вміст файлу «output.txt»

```
{
  "upload_time_total_creatives_sec": 56.71817779999765,
  "started_at": "2026-05-15T00:20:59",
  "finished_at": "2026-05-15T00:21:56",
  "links_total_before_filter": 400,
  "links_removed_by_history": 0,
  "links_kept_for_download": 400,
  "downloaded_rows_total": 400,
  "downloaded_rows_200": 300,
  "downloaded_rows_non200": 100,
  "download_time_total_wall_sec": 45.845467400002235,
  "download_time_video_direct_sec": 4.625310200000968,
  "download_time_video_youtube_sec": 0.0,
  "download_time_images_sec": 235.33845459998702,
  "download_time_html_sec": 0.0,
  "downloaded_video_direct_items": 11,
  "downloaded_video_youtube_items": 0,
  "downloaded_image_items": 289,
  "downloaded_html_items": 0
}
```

Рисунок 3.4 – Вміст файлу «runtime_upload.json»

Важливо зазначити, що далеко не всі посилання мають можливість вивантажитися. Так як рекламні матеріали постійно змінюються, а старі видаляються, є нормальним, що у великому масиві посилань можуть знайтися такі, що не будуть відкриватися. В такому разі слід казати, що множина X може дорівнювати, або бути меншою за множину L по кількості елементів. Тож важливо зазначити, що система обов'язково буде відслідковувати кількість таких невалідних посилань.

В результаті, коли креативи були успішно вивантажені, вони лежатимуть, як зазначалося раніше, у папці «Creatives_YYYY-MM-DD», вміст якої показано на рисунку 3.5. Слід зазначити, що номер у файлі «output.txt» це айді (не те ж саме, що id посилання), який додається до назви файлу на початку, саме таким чином система розуміє до якого файлу рекламного матеріалу належить конкретне посилання.

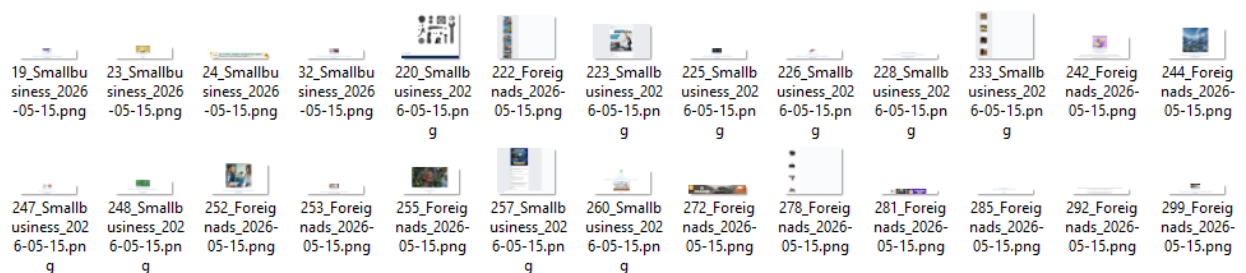


Рисунок 3.5 – Вміст теки з файлами креативів «Creatives_YYYY-MM-DD»

3.1.2 Модуль класифікації рекламних креативів

Цей модуль є окремим від модуля вивантаження файлів, але є однією зі складових системи. Він відповідає за обробку, класифікацію та зведення даних про креативи у зручний формат, а також запускається після вивантаження потрібних файлів першим модулем.

Для того щоб запустити цей модуль, треба перейти у теку «Ads_AI-Classification», та запустити файл «RUN.bat». Після цього відкриється головне вікно інтерфейсу, як показано на рисунку 3.6.

Рисунок 3.6 – Головна сторінка інтерфейсу модуля класифікації

У цьому вікні автоматично обирається тека з модулем вивантаження креативів, після цього в ній також автоматично знаходиться та обирається файл «output.txt» та тека з файлами креативів «Creatives_YYYY-MM-DD». Якщо теку з модулем вивантаження, або необхідні файли чи теки в ній не були знайдені, їх можна буде вибрати вручну.

Файл «output.txt» необхідний для відновлення зв'язків між файлами рекламних матеріалів x_i та посиланнями l_i , звідки вони були вивантажені. Також у вікні можна обрати тип репорту та дату, це лише технічні назви, щоб при збереженні метрик було зрозуміло, до якого саме звіту вони належать.

Після натискання на кнопку «Старт» починається обробка, класифікація, а після і формування звітів. Під час обробки користувач бачить вікно, зазначене на рисунку 3.7.

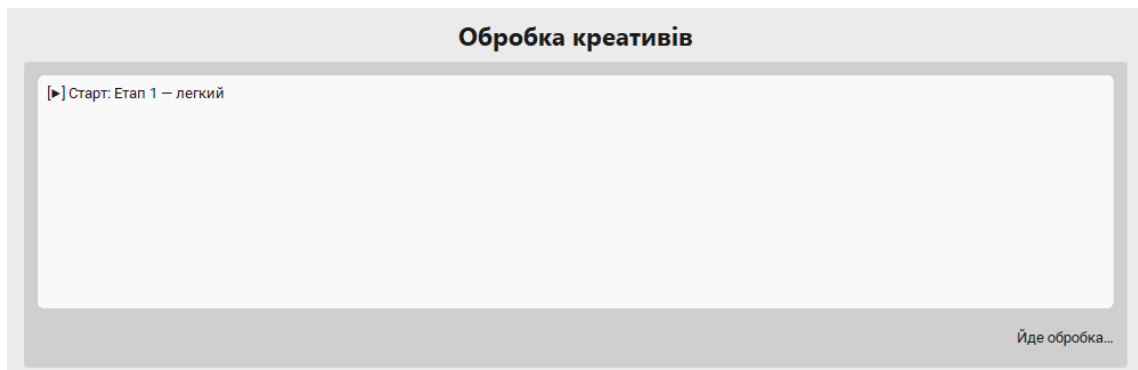


Рисунок 3.7 – Сторінка інтерфейсу під час обробки та класифікації

Спочатку модуль обробляє технічні файли щоб зрозуміти які з них вже були в історії, таким чином він одразу формує вибірку креативів, які будуть оброблятися за допомогою деякої функції $g(x_i)$, яка повертатиме y_i , як зазначено у (2.9). Також одразу визначаються типи файлів на дві підмножини зображень та відео за формулою (2.4), тому система зможе зрозуміти яку саме функцію обробки застосувати, це було вказано у (2.10).

Слід також уточнити для чого саме був збережений «бренд», який був автоматично визначений сервісом Gemius AdReal, але не є точним. Наприклад, для більшості рекламних оголошень визначається як «Foreign Ads» або «Small business». Вона потрібна для розділення файлів на менші групи всередині цих «брендів», між якими вже буде відбуватися пошук дублікатів за формулами (2.22), так як імовірність знайти їх в одному «бренді» набагато більша ніж в різних. Це суттєво економить час та ресурси, так як саме завдяки цьому алгоритму, складність пошуку дублікатів для обох типів файлів зазвичай не досягає $O(n^2)$ при умові, що «брендів» декілька.

У кінці результати обробки та класифікації об'єднуються та переходять у фінальний звіт, який показується в інтерфейсі, як вказано на рисунку 3.8.

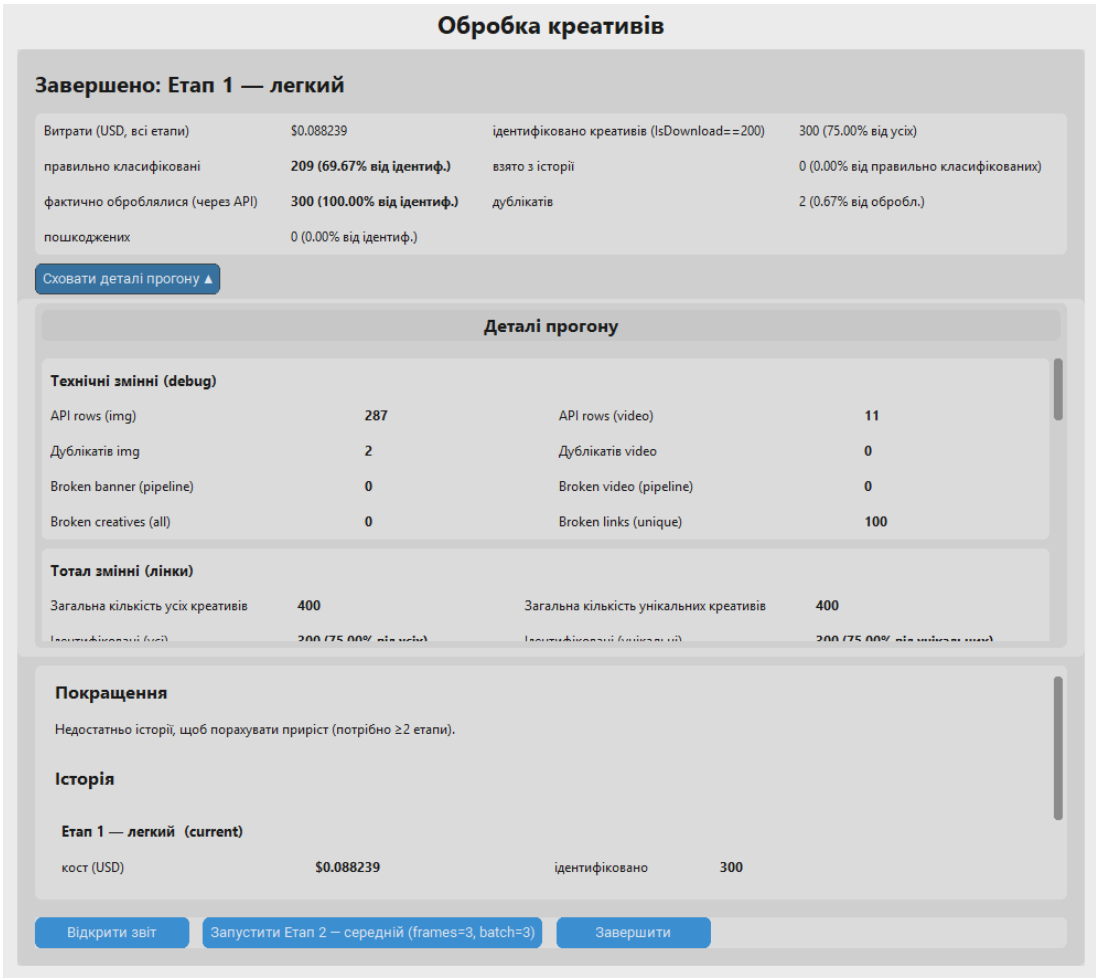


Рисунок 3.8 – Сторінка інтерфейсу після класифікації з усіма метриками

Після того як пройшла перша класифікація, можна запустити другу, вона швидко перекласифіковує ті креативи, які були не результативно класифіковані, тобто індикатор $\delta(y_i)$ дорівнює 0, його було зазначено у (2.11). Також при повторній класифікації береться більша кількість кадрів відео, що було описано у (2.6), для більш якісної класифікації. При завершенні роботи модуля користувач бачить вікно вказане на рисунку 3.9, де також буде зазначено покращення відносно минулої ітерації.

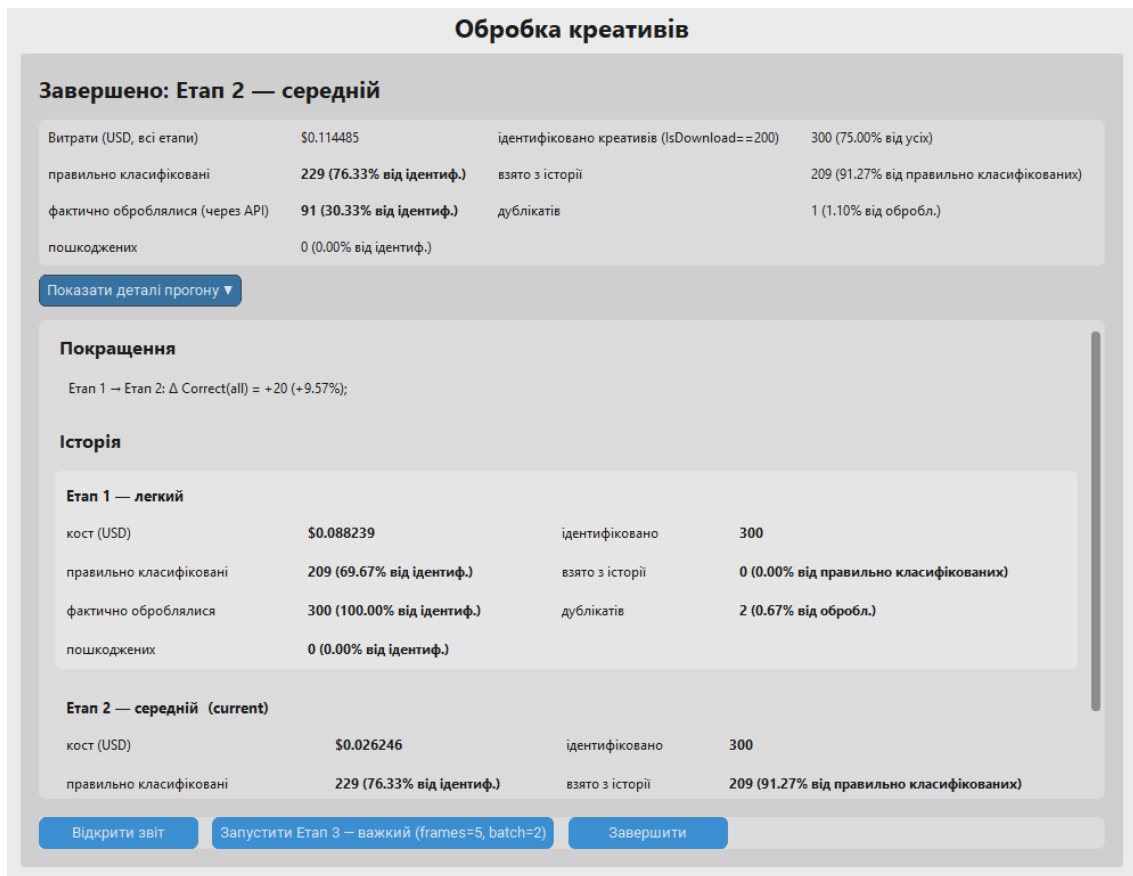


Рисунок 3.9 – Сторінка інтерфейсу після класифікації з усіма метриками після другої ітерації

Також після кожної ітерації доповнюються файли «history.csv» в якому лежить історія результативно класифікованих рекламних матеріалів за індикатором $\delta(y_i)$ та файл «history_metrics.csv» в якому лежать продубльовані метрики, які були вказані в інтерфейсі. Сам звіт генерується в окремій теці «Reports» з відповідною назвою заданою на головному вікні інтерфейсу та часом генерації. Повна ж інформація в реальному часі під час обробки, класифікації та після закінчення роботи системи виводиться у консолі, як показано на рисунку 3.10.

```

INFO: HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO: in=713 ($0.000891), cached=0 ($0.000000), out=124 ($0.001240), total=837 ($0.002131)
INFO: | % | Brand | Product | Category |
|---|---|---|---|
| 1 | Not found | Not found | Bell pepper |
| 2 | Not found | Not found | Bee pollen granules |
| 3 | Not found | Not found | News headlines |
| 4 | Not found | Not found | News headlines |
| 5 | Not found | Not found | Advertisement banner |
| 6 | Not found | Not found | Commemorative coin |
| 7 | Not found | Not found | Car headlight assembly |
INFO: Images summary: campaigns=26, representatives=87, duplicates=1
INFO: Додано 2 елементів із shared_list
INFO: Link auto-fill: було порожніх=91, стало порожніх=0
INFO: Видалено файл: C:\Users\user\PycharmProjects\__Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Class
ification\_temp\partial_image_results.csv
INFO: Видалено файл: C:\Users\user\PycharmProjects\__Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Class
ification\_temp\partial_video_results.csv
INFO: Видалено директорію: C:\Users\user\PycharmProjects\__Study\AI_Creative_Classification\AICreativeClassification\Ads_AI
-Classification\_temp
INFO: Звіт класифікації збережено: C:\Users\user\PycharmProjects\__Study\AI_Creative_Classification\AICreativeClassification\
Ads_AI-Classification\Reports\weekly report 15 May 2026 (created at 2026-05-15 02-10).csv
Загальний прогрес | in=9093 cached=0 out=1488 | $total=0.026246 | img 88/88 | vid 3/3: 100%| 91/91 [00:32<00:00, 2.8

```

Рисунок 3.10 – Частина виведеної у консоль інформації з обробки та класифікації креативів в кінці роботи

3.2 Проектування логіки роботи системи

Важливо чітко сформуванати схему системи, яка перетворює множину посилань L у множину результатів класифікації R . Якщо враховувати, що ці два модулі хоч і запускаються окремо та послідовно, але являють собою єдину систему. Тому було побудовано функціональну схему єдиної системи вивантаження, обробки та класифікації, з подальшою генерацією необхідних файлів та UML-діаграму послідовності роботи.

3.2.1 Функціональна схема системи

Для початку треба позначити функціональну схему, яку вказано на рисунку 3.11. На ній видно, що починається усе з множини посилань L , які обробляються функцією $d(l_i)$, після чого генеруються необхідні службова файли та вивантажуються рекламні матеріали множини X .

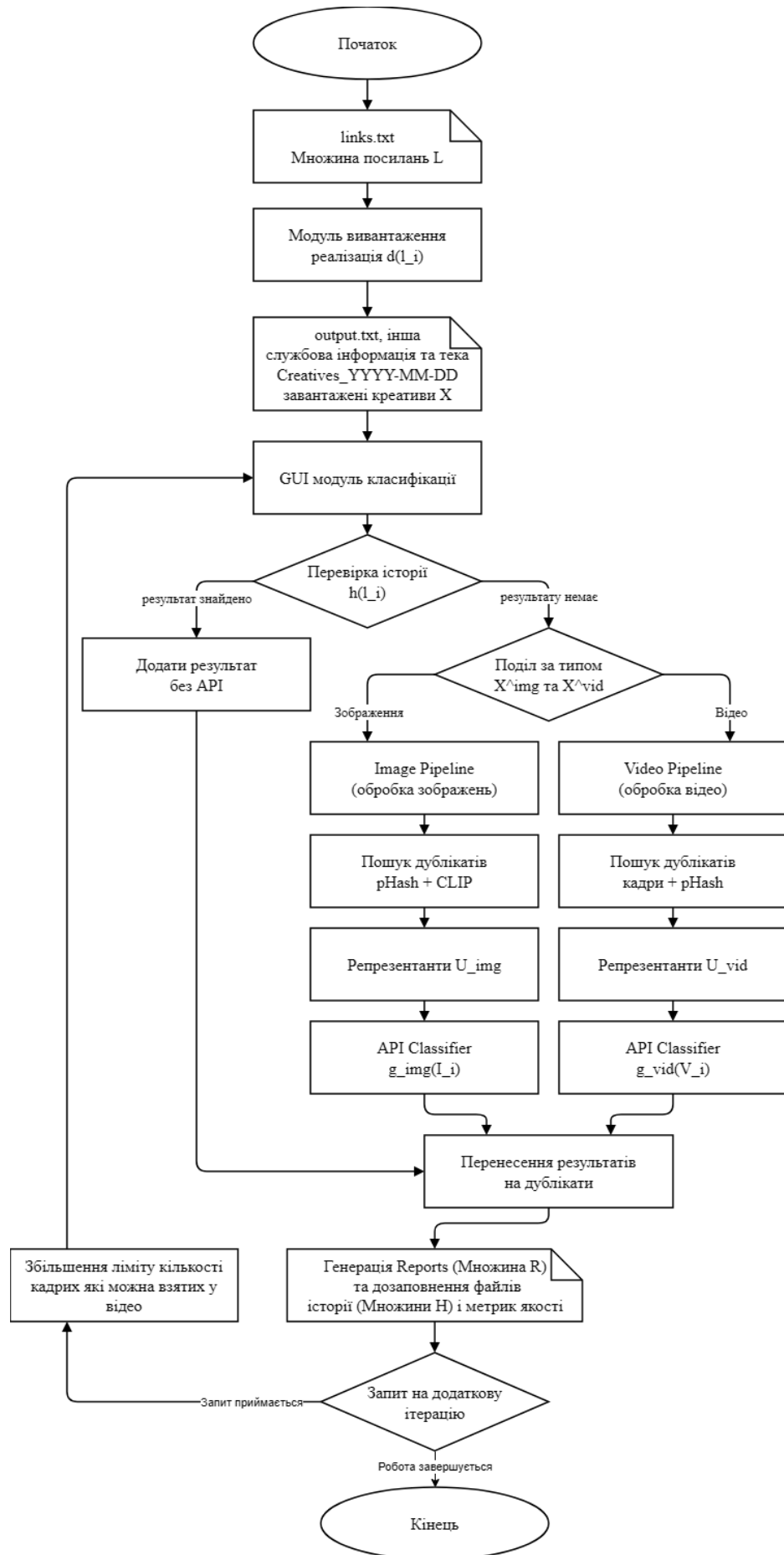


Рисунок 3.11 – Функціональна діаграма системи

Далі йде перевірка історії, як зазначено у (2.28), і якщо там нічого не знайдено, то система починає визначати якого типу кожен креатив для розуміння яку функцію обробки та класифікації $g(x_i)$ використати, як зазначено у (2.10). Наступним кроком йде порівняння за функціями подібності залежно від типу файлу та пошук дублікатів, разом із цим будується граф та вибирається один репрезентант з кожної множини групи дублікатів C_k . Потім йде класифікації та перенесення її результатів на дублікати. В кінці генеруються всі необхідні файли та репорт множини R з класифікованими даними.

3.2.2 UML-діаграма послідовності роботи компонентів

Sequence diagram показує послідовність дій системи у хронологічному порядку. По ній добре видно коли працюють конкретні функції чи блоки, і в якій послідовності, як зазначено на рисунку 3.12.

На діаграмі визначено 9 сутностей:

- 1) користувач;
- 2) модуль вивантаження;
- 3) файлова система (згенеровані модулем вивантаження технічні файли та завантажені креативи);
- 4) gui класифікатора (інтерфейс);
- 5) історія класифікації;
- 6) обробка креативів;
- 7) пошук дублікатів;
- 8) api мультимодальної моделі;
- 9) reports (звіти).

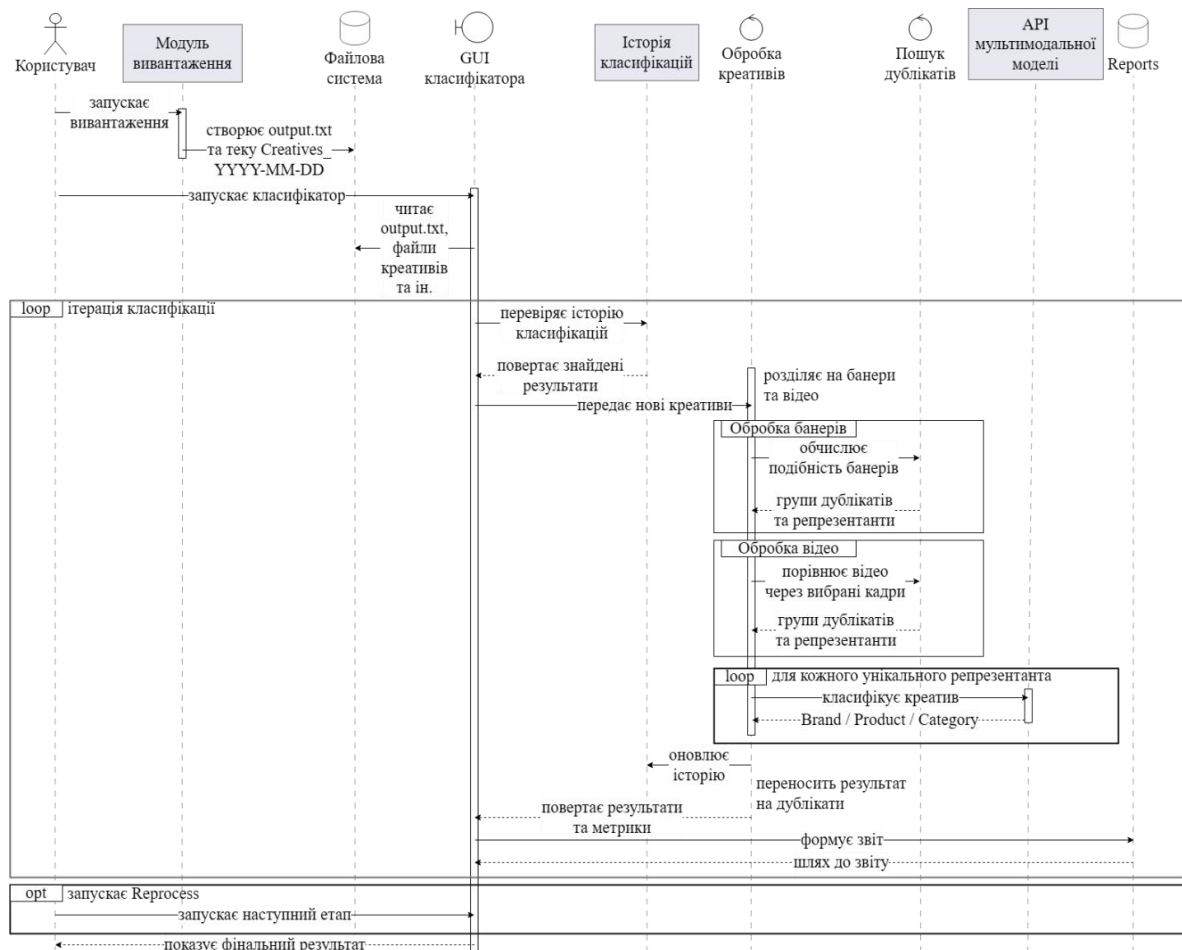


Рисунок 3.12 – Функціональна діаграма системи

Як і було згадано раніше, робота системи починається з модуля вивантаження, який бере посилення l_i та вивантажує їх у креативи x_i . Після цього починає працювати модуль обробки та класифікації, він перетворює креативи x_i у класифіковану відповідь y_i , після чого генерується звіт множини R з зібраними результатами роботи системи r_i .

На діаграмі також видно проміжні блоки окремої обробки зображень та відео на етапі обробки креативів. Далі йде цикл пошуку репрезентант для кожної групи зв'язаних дублікатів. Після оновлення історії, вивід результатів і генерацію звіту. Також є окрема опція для нової ітерації на випадок потреби другої спроби обробки зображень, та додавання більшої кількості кадрів для якіснішої класифікації рекламних відео.

3.3 Реалізація алгоритмів обробки рекламних креативів

Як і було визначено в формулі (2.10), функція $g(x_i)$ змінюється залежно від того, який тип креативів вона обробляє. Через це потрібно було розробити два окремих алгоритми залежно від формату вивантаженого файлу.

3.3.1 Алгоритм обробки банерних зображень

Для обробки множини креативів X^{img} виділяється окремий конвеєр, який відповідає за файли типу зображень. Виділяється декілька форматів файлів зображень з якими цей блок працює, а саме: .jpg, .jpeg, .png та .gif. Якщо з першими трьома питання ніяких немає, то формат «gif» ближчий до відео ніж до статичного зображення. Насправді в його випадку береться один кадр і він вже обробляється як звичайне зображення. Це рішення було прийняте через те, що більшість «gif» є чи статичними чи дуже короткими, і витратити на них стільки ж ресурсу та часу, як на відео, просто не є ефективним рішенням.

Спочатку файли групуються за «брендом», який визначив сам сервіс Gemius AdReal, для зменшення навантаження на систему. Це рішення було впроваджено коли стало зрозуміло, що шанс знайти дублікати в різних «брендах» суттєво менший за шанс знайти їх в одному. Після для кожного зображення формується рHash, як і було сказано раніше, він потрібен для швидкого пошуку схожих зображень, щоб одразу відсіяти очевидні дублікати. Загальна умова схожості креативів відповідає співвідношення (2.12), але для рHash-порівняння послідовно використовуються отримання хешу за формулою (2.13), обчислення відстані Геммінга за сумою (2.14), перевірка порогу за відношенням (2.15) та функції подібності $sim_p(x_i, x_j)$, яка була зазначена в (2.16).

Далі використовується більш потужний метод, а саме формуються векторні представлення через модель CLIP [11]. Про embedding було написано ще в пункт 2.3.2. Отримання векторного представлення зображення відповідає (2.17), а семантична подібність між такими представленнями обчислюється за (2.18). Модель CLIP потрібна не для фінальної класифікації, а для пошуку семантично схожих банерів (зображень). Так як файли були розбиті на менші групи, ця модель не викликається занадто часто, тому обробка великого об'єму файлів може зайняти декілька хвилин, а не годин.

Тоді встає питання навіщо було використовувати таку модель, якщо вона не обробляє миттєво. Насправді на великих об'ємах даних в результаті все одно заощаджується час, оскільки не треба класифікувати дуже багато дубльованих файлів які може знайти ця модель. Але час це не єдина проблема, модель CLIP є безкоштовною та запускається локально, в той час, як класифікатор через API, який вже буде займатися класифікацією креативів, є платним. Суттєве зменшення кількості запитів до API однакового розміру гарантує зменшення сумарної ціни за класифікацію.

Під час пошуку схожих зображень формується граф подібності за (2.20), а ребра між креативами визначаються за умовами, зазначеними в (2.21) та (2.22). Групи дублікатів розглядаються як зв'язні компоненти графа або ж множина (2.23). При цьому важливо зазначити, що вони не перетинаються, а їх об'єднання повністю покриває множину рекламних креативів, як вказано в (2.24). У кожній такій групі обирається репрезентант за формулою (2.25), після чого формується множина репрезентантів за (2.26), а решта креативів належать до множини дублікатів за (2.27).

Класифікація відбувається за допомогою мультимодальної моделі через OpenAI API. Тобто в цій роботі використовується вже навчена модель, як працює не локально, а на серверах OpenAI, тому класифікація відбувається доволі швидко. Під час класифікації моделі дається зображення та інструкція з проханням створити Markdown формат таблиці із необхідними колонками, які потім отримуються зі згенерованого тексту, та переводять у зручний

формат. У випадку коли класифікація не була успішно, система робить повторний запит до API.

Після класифікації, ці відповіді переносяться з репрезентантів на їх дублікати та записуються у відповідні файли.

Схема алгоритму обробки зображень показана на рисунку 3.13.

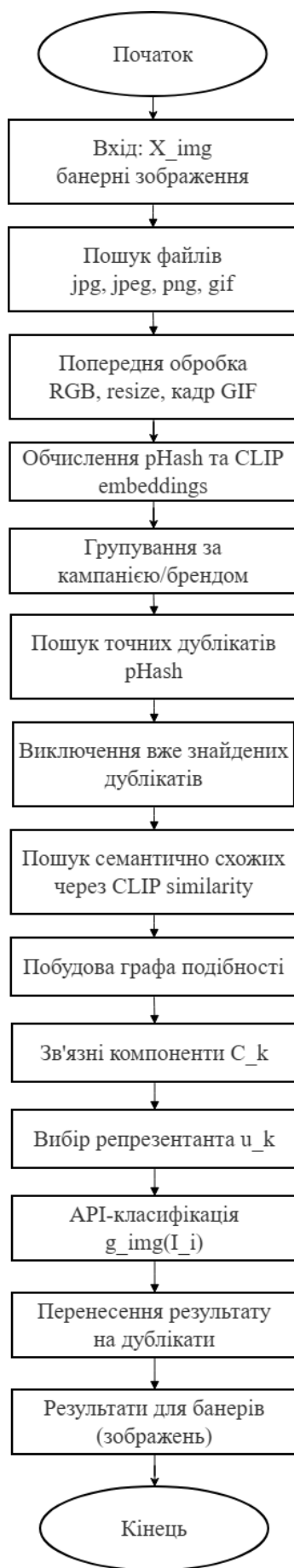


Рисунок 3.13 – Схема роботи алгоритму обробки та класифікації зображень

3.3.2 Алгоритм обробки відеокреативів

Для обробки множини креативів X^{vid} також є окремий конвеєр, він відповідає за файли типу відео. В ньому можна виділити також декілька форматів файлів з якими він може працювати, їх тут більше: .mp4, .avi, .mov, .mkv, .webm, .m4v, .mpeg та .mpg.

Під час обробки відео треба враховувати важливу річ, відео це послідовність кадрів, які змінюються з часом, а модель може класифікувати зображення, як і пошук дублікатів можна здійснювати тільки по зображеннях. Тому треба розбивати відео на кадри, як це було вказано у формулі (2.6), та працювати з ними. Але треба розуміти, що якщо відео має умовні 30 кадрів в секунду, а саме відео триває 1 хвилину, то це буде понад $30 \times 60 = 1800$ кадрів. Звісно такий великий об'єм зображень всього для одного відео обробляти не треба, тому є доцільним рішення брати лише конкретні кадри. Було додано алгоритм, який брав N кількість кадрів рівномірно по всьому проміжку відео, а не просто всі кадри.

Далі для кожного кадру рахується rHash, тобто у кожного такого відео буде N таких хешів. Кадри порівнюються всі до всіх у кожній парі відео, як і було зазначено у формулі (2.19). Тобто якщо хоч один кадр першого відео буде схожий на деякий кадр другого відео, то ці відео вважаються схожими. Цей алгоритм не є найнадійнішим, але він доволі інтуїтивний, простий і швидко працює. Створення embedding та використання моделі CLIP в обробці відео немає, так як цей метод доволі ресурсомісткий, і використовувати його на креативи відео формату не є доцільним. Під час цього, як і з зображеннями, будується граф подібності, а після беруться репрезентанти від кожної групи дублікатів.

Класифікація тут трохи відрізняється в тому, що на вхід моделі для кожного відео через API подається список підписаних кадрів відео, після чого від моделі вимагається відповісти таблицею формату Markdown, де один запис

це одне відео, а не кадр. Оскільки запити подаються одразу по декілька зображень, то при класифікації умовних 5 відео подається, наприклад, 10 кадрів, якщо від кожного береться по два кадри. Після формується таблиця з 5 записами. Завдяки цьому мультимодальна модель має семантику не тільки одного кадру, а і контексту відносно усіх поданих кадрів для кожного відео.

Після класифікації так само всі відповіді переносяться з репрезентантів на їх дублікати та записуються у відповідні файли.

Схема алгоритму обробки відео показана на рисунку 3.14.

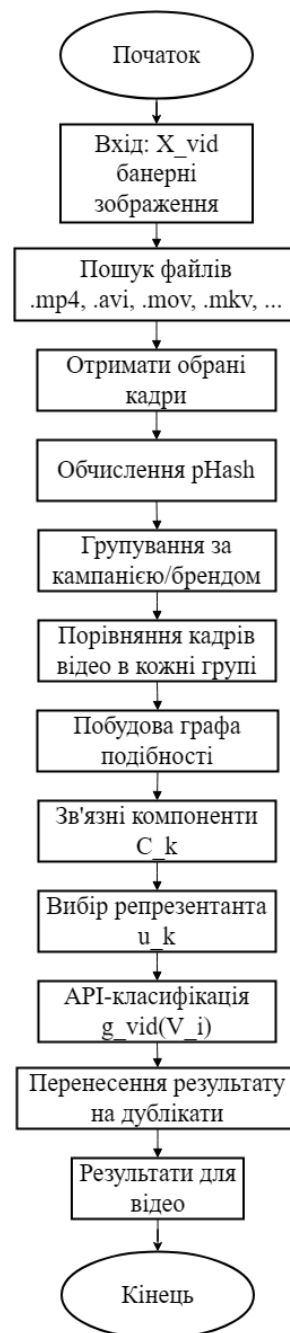


Рисунок 3.14 – Схема роботи алгоритму обробки та класифікації відео

3.3.3 Формування промпта та класифікація через API

На початку важливо нагадати, що в цій системі використовуються вже навчені моделі. Для безпосередньої класифікації використовується зовнішня мультимодальна модель «gpt-5» від OpenAI через API. У межах поточного дослідження основна увага приділяється не порівнянню моделей, а побудові повної системи обробки рекламних матеріалів та експериментальному оцінюванню результатів її роботи.

Система робить запит до сервера із промптом та файлами, після чого отримує відповідь. Задача моделі отримати зображення чи кадри до відео та інструкцію, та написати відповідь чітко у форматі Markdown таблиці. Далі спеціальний алгоритм шукає цю таблицю у згенерованому тексті та переводить її у правильний формат. Якщо таблиці не знайдено, чи в ній не вистачає записів, то система робить новий запит до API, щоб регенерувати відповідь. Такі випадки можуть відбуватися не часто, але і повністю їх позбутися немає можливості.

Окрім самих файлів на модель подається спеціально згенерований промпт. Він створений з декількох важливих частин. У першій частині визначається завдання та правила відповідно до формату креативу, як показано на рисунку 3.15. У другій частині уточняється, щоб вивід був у форматі Markdown. А у третій частині вже генерується скриптом і дозволяє динамічно додавати стільки колонок скільки треба, як зазначено на рисунку 3.16.

```

ADMIN_PROMPT_SCHEMA = {
  "image": {
    "task": (
      "You'll receive an image (with a built-in caption) and must classify it strictly into the table below. "
      "The entire output MUST be written in English only, regardless of the input language."
    ),
    "rules": [
      "Use caption and visual content to extract all fields.",
      "If something is not visible or unknown, write 'Not found'.",
      "Output ONLY the table described below."
    ]
  },
  "video": {
    "task": (
      "You'll receive frames from several videos (each frame labeled by its source video). "
      "The entire output MUST be written in English only, regardless of the input language."
      "For each video, use all its frames to determine a single classification."
    ),
    "rules": [
      "If ANY frame has information, use it to classify the WHOLE video.",
      "Do NOT classify frames individually - one row per video only.",
      "If something is not visible or unknown across frames, write 'Not found'.",
      "Output ONLY the table described below."
    ]
  }
}

```

Рисунок 3.15 – Частина схеми промпту з правилами та завданнями

```

pb = PromptBuilder(admin_prompt_schema, create_if_missing=True)

if len(pb.columns) == 1:
    pb.add_column(
        "Brand",
        image_rule="company or trademark (e.g. Samsung, L'Oreal), or 'Not found'", image_example="Samsung",
        video_rule="company or trademark (e.g. Samsung, L'Oreal), or 'Not found'", video_example="Samsung",
    )
    pb.add_column(
        "Product",
        image_rule="model name/number (e.g. S24, A15), or 'Not found'", image_example="S24",
        video_rule="model name/number (e.g. S24, A15), or 'Not found'", video_example="S24",
    )
    pb.add_column(
        "Category",
        image_rule="general type (e.g. kitchen grill, gaming laptop, throat lozenges), or 'Not found'", image_example="Smartphone",
        video_rule="general type (e.g. kitchen grill, gaming laptop, throat lozenges), or 'Not found'", video_example="Smartphone",
    )

```

Рисунок 3.16 – Приклад швидкого додавання колонок для промпту

Інструкції або промпти часто пишуть саме англійською, так як модель навчалась на текстах з інтернету, а більша частина усієї літератури написана англійською, тому и модель розуміє цю мову та слідує правилам краще. Наразі система повинна отримати відповідь y_i , яка має в собі сутності b_i – бренд, p_i – продукт, c_i – категорія, які згадані в (2.7). Тому і формуються відповідні колонки, а також номер поданого на класифікацію файлу, тому перша колонка є службовою. Відповіді моделі пишуть в консолі у реальному часі під час роботи, ці відповіді показані на рисунках 3.17–3.18.

```
INFO: HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO: in=935 ($0.001169), cached=0 ($0.000000), out=154 ($0.001540), total=1089 ($0.002709)
INFO: | % | Brand | Product | Category |
|---|---|---|---|
| 1 | Not found | Not found | Not found |
| 2 | Not found | Not found | Not found |
| 3 | Citroën | C4 | Car |
| 4 | Maserati | Grecale | Car |
| 5 | Maserati | Grecale | Car |
| 6 | Maserati | Grecale | Car |
| 7 | Opel | Frontera | Car |
| 8 | Not found | Not found | Not found |
| 9 | Not found | Not found | Not found |
| 10 | JustClass | Not found | Online learning platform |
```

Рисунок 3.17 – Відповідь ІІІ на запит на класифікацію 10 зображень

```
INFO: HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO: in=982 ($0.001228), cached=0 ($0.000000), out=93 ($0.000930), total=1075 ($0.002158)
INFO: | % | Brand | Product | Category |
|---|---|---|---|
| 1 | Nissan | Qashqai | Car |
| 2 | Kia | Not found | Car |
| 3 | ZYN | Not found | Nicotine pouches |
| 4 | Nike | Not found | Sneakers |
| 5 | alivemoment.com | Not found | Photo restoration service |
```

Рисунок 3.18 – Відповідь ІІІ на запит на класифікацію 5 відео по 2 кадри

3.4 Формування звіту, історії класифікації та операційних метрик

При формуванні звіту результати зображень та відео об'єднуються. Для кожного файлу система може відновити посилання через «output.txt», після чого записати в історію результативно класифіковані креативи відповідно до індикатора (2.11). У звіт в свою чергу записуються взагалі усі посилання та до кожного проставляється статус S_i , де:

- 1) 0 – не вивантажився;
- 2) 1 – класифікований;
- 3) 2 – вивантажився, але не обробився.

Значення 0 буває тільки тоді коли посилання чи не відкривається, чи за ним нічого немає; значення 2 буває у випадках коли файл вивантажився, але некоректно, наприклад найчастіший випадок коли відео було пошкоджено, тому воно змогло вивантажитися, але не змогло обробитися. Такі ситуації

іноді бувають, тому щоб не губити посилання, для них було вирішено виділити окремий статус 2.

Після роботи системи доповнюються файли «history.csv» та «history_metrics.csv». Повторне використання результату з історії або запуск нової класифікації для креативу відповідає формулі (2.29), а оновлення історії після отримання придатного результату виконується за (2.30). Також генерується файлу звіту «[type] report [day] [month] [year] (created at YYYY-MM-DD hh-mm).csv», назва якого формується відносно початкових налаштувань на головній сторінці інтерфейсу, а також часу генерації. Частковий зміст файлів вказано на рисунках 3.19–3.21. Також повний список класифікації з файлу звіту зазначено в таблиці А.1 додатку А.

Link	DateAdded	Status	Ext	Product					
https://adreal.gemius.com/ua/show/P0B	15.05.2026		1 .mp4	SolaX - Home energy storage system					
https://adreal.gemius.com/ua/show/P0A	15.05.2026		1 .mp4	Nissan Qashqai Car					
https://adreal.gemius.com/ua/show/P0a	15.05.2026		1 .mp4	Kia - Car					
https://adreal.gemius.com/ua/show/P08	15.05.2026		1 .mp4	ZYN - Nicotine pouches					
https://adreal.gemius.com/ua/show/P0A	15.05.2026		1 .mp4	Nike - Sneakers					
https://adreal.gemius.com/ua/show/P0B	15.05.2026		1 .mp4	alivemoment.com - Photo restoration service					
https://adreal.gemius.com/ua/show/P01	15.05.2026		1 .mp4	Shop Pay - Payment/checkout service					
https://adreal.gemius.com/ua/show/P04	15.05.2026		1 .mp4	IQOS ILUMA Heated tobacco device					
https://adreal.gemius.com/ua/show/P06	15.05.2026		1 .png	Citroën C4 Car					
https://adreal.gemius.com/ua/show/P04	15.05.2026		1 .png	Maserati Grecale Car					
https://adreal.gemius.com/ua/show/P07	15.05.2026		1 .png	Maserati Grecale Car					
https://adreal.gemius.com/ua/show/P08	15.05.2026		1 .png	Maserati Grecale Car					
https://adreal.gemius.com/ua/show/P00	15.05.2026		1 .png	Opel Frontera Car					

Рисунок 3.19 – Частковий вміст файлу «history.csv»

report_date	generated_at	report_type	stage	total_links	total_links	identified	identified	identified	identified	identified	identified	identified
15.05.2026	15.05.2026 1:16	weekly	1	400	400	300	300	75.0	75.0	11	11	2
15.05.2026	15.05.2026 2:11	weekly	2	400	400	300	300	75.0	75.0	11	11	2

Рисунок 3.20 – Частковий вміст файлу «history_metrics.csv»

(не всі метрики видно)

Link	Status	Product							
https://adreal.gemius.com/ua/show/P0C	1	InfoCar.site - Headlight polishing service							
https://adreal.gemius.com/ua/show/P0C	1	PixVerse V5.5 Model AI video generation tool							
https://adreal.gemius.com/ua/show/P0C	1	Упаковочка - Packaging supplies							
https://adreal.gemius.com/ua/show/P0C	1	KARMA - Drone frame components							
https://adreal.gemius.com/ua/show/P0C	1	- DS Engines Game creation tool advertisement							
https://adreal.gemius.com/ua/show/P0C	1	Opel Frontera Car							
https://adreal.gemius.com/ua/show/P0C	1	VELO - Nicotine pouches							
https://adreal.gemius.com/ua/show/P0C	1	myIQ - Online quiz ad							
https://adreal.gemius.com/ua/show/P0C	1	- - Car headlight assembly							
https://adreal.gemius.com/ua/show/P0C	1	VakıfBank - Banking service							
https://adreal.gemius.com/ua/show/P_0	1	Hero Wars - Mobile puzzle ad							
https://adreal.gemius.com/ua/show/P0C	1	Arbitrage UP Стартовий курс по арбітражу трафіку Online course							
https://adreal.gemius.com/ua/show/P0C	1	hint - Online quiz/test							

Рисунок 3.21 – Частковий вміст файлу звіту

3.5 Контрольна вибірка та ручне оцінювання результатів класифікації

Для експериментального оцінювання результатів класифікації системи сформовано контрольну вибірку зі 100 посилань на рекламні креативи. До цієї вибірки увійшли креативи з остаточного звіту системи після обробки. Для кожного рекламного матеріалу система вписує сутності у форматі «Brand || Product || Category» в один стовпчик.

Ручна розмітка відбувалася шляхом порівняння результату роботи системи з реальним змістом креативу за посиланням. Для кожної з шуканих трьох сутностей окремо зазначалося, чи збігається відповідь системи з експертною оцінкою. Для цього використовувалися поля «brand_ok», «product_ok» та «category_ok» зі значеннями «yes» або «no» відповідно до того чи відповідає реальний зміст рекламного матеріалу із тим, що знайшла система. Також слід уточнити що, якщо система та експерт не знайшли однієї з сутностей на креативі, то експерт також пише «yes».

Окремо було сформовано поле «expert_usable», яке відповідає індикатору придатності $\delta(y_i)$, який був описаний в підрозділі 2.2. Результат вважався придатним лише тоді, коли система заповнила Brand або Product, а відповідна сутність була підтверджена ручною оцінкою як правильна. Якщо коректною була лише Category, то в такому випадку результат не вважатиметься придатним для «expert_usable», так як Category у поточній задачі є допоміжною корисною сутністю, але не необхідний.

Отже, ручна експертна розмітка використовується у двох випадках. По-перше, для оцінювання правильності сутностей Brand, Product та Category. По-друге, розмітка потрібна для перевірки наскільки автоматичний індикатор придатності взагалі узгоджується з реальною оцінкою. На рисунку 3.22 показано частковий вміст файлу після ручної експертної розмітки.

Link	Status	data	pred_brand	pred_product	pred_category	indicator_pred	expert_usable	quality_group	meaningful_correct_fields	brand_ok	product_ok	category_ok
https://	1	InfoCar.site - Headlight polishing s	InfoCar.site	-	Headlight polish	1	0	Not usable (cate	Category	no	yes	yes
https://	1	PixVerse V5.5 Model Al video gener	PixVerse	V5.5 Model	Al video generati	1	1	BPC correct	Brand; Product; Category	yes	yes	yes
https://	1	Упаковочка - Packaging supplies	Упаковочка	-	Packaging suppl	1	1	Usable by Brand	Brand; Category	yes	yes	yes
https://	1	KARMA - Drone frame components	KARMA	-	Drone frame con	1	1	Usable by Brand	Brand; Category	yes	no	yes
https://	1	DS Engines Game creation tool ac-	DS Engines	Game creation to	1	1	Usable by Brand	Product; Category	yes	yes	yes	
https://	1	Opel Frontera Car	Opel	Frontera	Car	1	1	BPC correct	Brand; Product; Category	yes	yes	yes
https://	1	VELO - Nicotine pouches	VELO	-	Nicotine pouches	1	1	Usable by Brand	Brand	yes	no	no
https://	1	myIQ - Online quiz ad	myIQ	-	Online quiz ad	1	1	Usable by Brand	Brand; Category	yes	no	yes
https://	1	- Car headlight assembly	-	-	Car headlight as	0	0	Not usable		yes	yes	no
https://	1	VakifBank - Banking service	VakifBank	-	Banking service	1	0	Not usable (cate	Category	no	no	yes
https://	1	Hero Wars - Mobile puzzle ad	Hero Wars	-	Mobile puzzle ad	1	1	Usable by Brand	Brand; Category	yes	no	yes
https://	1	Arbitrage UP Стартувий курс по арбі	Arbitrage UP	Стартувий курс	Online course	1	1	BPC correct	Brand; Product; Category	yes	yes	yes
https://	1	hint - Online quiz/test	hint	-	Online quiz/test	1	1	Usable by Brand	Brand; Category	yes	yes	yes
https://	1	- Batteries	-	-	Batteries	0	0	Not usable		yes	yes	no
https://	1	- Investment academy course	-	-	Investment acad	0	0	Not usable		no	no	no
https://	1	- Stock trading app	-	-	Stock trading ap	0	0	Not usable (cate	Category	yes	yes	yes
https://	1	Tvei Coffee Streblum? - Coffee	Tvei Coffee St	-	Coffee	1	0	Not usable (cate	Category	no	yes	yes
https://	1	ROBEAUTY reMAX Skincare cosmet	ROBEAUTY	reMAX	Skincare cosmet	1	1	BPC correct	Brand; Product; Category	yes	yes	yes

Рисунок 3.22 – Частковий вміст файлу звіту після ручної розмітки

3.6 Аналіз результатів експериментального оцінювання роботи системи

Для перевірки роботи системи було взяти 400 посилань на рекламні креативи. Частина посилань працювала, а частина – ні. Система згенерувала звіт з полями Link, Status, Product, що відповідає зазначеній формулі (2.8), при цьому колонка Product містить результат класифікації y_i із 3 сутностями як розділяються символами «||». Сутності, яка мультимодальна модель не змогла класифікувати, позначено як «-». Було зроблено дві ітерації обробки. Усі метрики для аналізу були взяті з файлу «history_metrics.csv». Спочатку можна подивитися загальну характеристику тестового набору в таблиці 3.1.

Таблиця 3.1 – Загальна характеристика тестового набору

Показник	Значення
Усього посилань	400
Унікальних посилань	400
Успішно завантажено креативів	300
Неробочих або недоступних посилань	100
Банерів серед завантажених	289
Відео серед завантажених	11
Частка завантажених креативів	75.00%
Частка неробочих посилань	25.00%

Продовження таблиці 3.1

Частка банерів	96.33%
Частка відео	3.67%

Видно, що з 400 посилань вдалося вивантажити 300, у свою чергу 100 посилань були невалідними і мають статус 0 у звіті. Більшість завантажених креативів банери, їх кількість від усіх 96.33%, відео реклами в цій вибірці набагато менше, усього 3.67% вивантажених файлів. Тому результат для зображень є більш показовим.

Тепер наведемо таблицю 3.2 з метриками другої ітерації, а після порівняємо її з першою ітерацією.

Таблиця 3.2 – Результати другої ітерації системи

Показник	Значення
Усього посилань	400
Завантажено креативів	300
Неробочих посилань	100
Банерів	289
Відео	11
Оброблено через API	91
Взято з історії	209
Результативно класифіковано	229
Результативно класифіковано банерів	220
Результативно класифіковано відео	9
Знайдено дублікатів банерів	1
Знайдено дублікатів відео	0
Вартість API, USD	0.026246
Час API-класифікації, с	78.94
Час pipeline-обробки, с	32.04

У другій ітерації видно, що кількість посилань не змінилася, як і кількість доступних для обробки креативів. Також зазначено, що з історії взяти 209 посилань у другій ітерації, це пов'язано з тим, що в першій було результативно класифіковано за індикатором придатності $\delta(y_i)$ та записано в історію ці 209 креативів. А інші 91 пішли на повторну обробку, так як за індикатором (2.11) не вважалися придатно класифікованими. Дублікатів в такому малому наборі небагато, тому вплив дедуплікації тут несуттєвий. Треба подивитися на відповідні дані з першої ітерації та порівняти, тому нижче наведено таблицю 3.3 порівнянь першої та другої ітерації.

Таблиця 3.3 – Порівняння першої та другої ітерації

Показник	Перша ітерація	Друга ітерація	Зміна
Усього посилань	400	400	0
Завантажено креативів	300	300	0
Неробочих посилань	100	100	0
Взято з історії	0	209	209
Оброблено через API	300	91	-209
Результативно класифіковано	209	229	20
Результативно класифіковано банерів	201	220	19
Результативно класифіковано відео	8	9	1
Дублікатів банерів	2	1	-1
Вартість API, USD	0.088239	0.026246	-70.25%
Час API-класифікації, с	256.79	78.94	-69.26%
Час pipeline-обробки, с	71.27	32.04	-55.04%

Як можна побачити, саме перша ітерація створила історію класифікації, після чого у другій ітерації цей список посилань система просто зібрала та перевикористала. Можна побачити суттєве зменшення кількості креативів, які пішли на API, а також зменшення ціни та часу обробки на другій ітерації. Дублікатів стало на один менше через те, що один з репрезентантів було

результативно класифіковано, тому на його дублікат це теж розповсюдилося, в той же час інший репрезентант не був придатним згідно умови індикатора $\delta(y_i)$, тому і його дублікат залишився в другій ітерації.

Проводячи експертне ручне маркування та оцінку результатів звіту другої ітерації, а саме 100 довільних посилань (в яку було включено усі 11 відео), стало зрозуміло, що якість класифікації показує відносно хороший результат. Як видно на таблиці 3.4, більше половини відповідей збігаються з реальним маркуванням.

Таблиця 3.4 – Оцінка ручної розмітки 100 посилань із відповідями у сформованому звіті після другої ітерації

Показник	Кількість
Brand збігається з ручною оцінкою	76
Product збігається з ручною оцінкою	79
Category збігається з ручною оцінкою	88
Brand і Product збігаються одночасно	65
Brand, Product і Category збігаються одночасно	61
Є коректно визначений Brand або Product	64

За результатами оцінки видно, що найвищий показник має сутність Category, це логічно, оскільки для визначення до якої категорії належить рекламний креатив достатньо просто розуміти семантику. У свою чергу, щоб знайти бренд або ж продукт, треба щоб він був явно конкретизований.

Після експертної оцінки та розмітки можна побудувати матрицю невідповідностей, яка показана в таблиці 3.5.

Таблиця 3.5 – Матриця невідповідностей індикатора придатності

	Експерт: придатний	Експерт: непридатний
Індикатор: придатний	64	10
Індикатор: непридатний	0	26

Також треба порахувати метрики Accuracy, Precision, Recall та F1-score, які було розібрано в підрозділі 2.7.2. Спочатку треба порахувати, що загальна кількість дорівнює 100:

$$N = TP + FP + FN + TN = 64 + 10 + 0 + 26 = 100$$

Тепер рахуються всі чотири метрики по черзі за відповідними формулами:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$= (64 + 26) / (64 + 26 + 10 + 0) = 90 / 100 = 0.90,$$

$$Precision = TP / (TP + FP) = 64 / (64 + 10) = 64 / 74 \approx 0.86,$$

$$Recall = TP / (TP + FN) = 64 / (64 + 0) = 64 / 64 = 1.00,$$

$$F1 = 2 \cdot Precision \cdot Recall / (Precision + Recall)$$

$$= 2 \cdot 0.86 \cdot 1.00 / (0.86 + 1.00) \approx 0.93.$$

Отже можна отримати значення зазначені в таблиці 3.6.

Таблиця 3.6 – Метрики роботи індикатора придатності

Метрика	Значення
Accuracy	0.90
Precision	0.86
Recall	1.00
F1-score	0.93

З матриці невідповідностей добре видно, що індикатор не відкинув жодного результату системи, що за ручною розміткою мав коректно визначений Brand або Product. Але було 10 хибнопозитивних випадків, коли індикатор прийняв результат як придатний через наявність Brand або Product, але експертний аналіз показав, що ці сутності класифікувалися некоректно або їх не було в креативі. Інші результати в цілому відповідають істині.

Значення Accuracy = 0.90 вказує на те, що індикатор зазвичай відповідає експертній оцінці. Recall = 1.00 означає, що за індикатором система не пропустила жодного результату, який вручну був визнаний придатним за Brand або Product. Звісно він може пропускати, але в даній вибірці цього не сталося. Precision = 0.86 показує, що невелика частина результатів, визначених

індикатором як придатні, після ручної перевірки виявилася некоректною. Це пояснюється тим, що за індикатором система перевіряє сам факт наявності Brand або Product, але не може самостійно оцінити їх змістовну правильність.

Тож, індикатор придатності є доволі корисним автоматичним фільтром порожніх або незадовільних (визначається списком заборонених значень) відповідей. Він не є гарантом правильності класифікації, але для автономності в умовах невизначеності множини класів є корисним інструментом.

Звісно треба враховувати, що вибірка, на якій були пораховані ці метрики, має в собі всього 100 посилань, у випадку цієї задачі доволі складно розмічати дані, оскільки немає можливості виділити конкретну множину класів, а через ручну роботу така розмітка займає доволі багато часу.

3.7 Обмеження реалізованої системи та напрями подальшого розвитку

Першою найважливішою приміткою можна назвати невідповідні посилання, вони постійно видаляються один за одним, тому в старих збірках посилань будуть працювати одиниці. Іноді аналітикам потрібно проаналізувати збірку посилань реклами за конкретний старий період, а її ніяк більше не отримати.

Другою проблемою є нерівномірний набір, особливо якщо вибірка невелика. Зазвичай такі збірки посилань мають в собі 30 000 посилань та більше, але обробка такої кількості посилань може займати декілька годин, а якщо більшість з них будуть відео, то і пів дня. Але слід розуміти, що ця система була розроблена саме на такі об'єми даних, оскільки ручна класифікація такої кількості креативів займала б тижні.

Також проблемою є мала кількість дублікатів, але знову ж таки ця тестова вибірка не є достатньо великою для того щоб в ній попалося багато схожих рекламних матеріалів для більш глибокого аналізу. Цей механізм більше розрахований на велику кількість посилань.

Ще треба уточнити, що далеко не завжди виходить отримати відповідь по усім 3 сутностям, найчастіше якогось одного стовпця чи нема чи там написано щось загальне, але для аналізу цього зазвичай вистачає. В цій тестовій вибірці результати доволі хороші, на другій ітерації 229 креативів з 300 класифіковано придатно, це понад 76%, це відсоток креативів, результат класифікації яких задовольняє індикатор $\delta(y_i)$, тобто він дорівнює 1.

Також важливо мати на увазі, що попри проведення ручної експертної оцінки, контрольна вибірка зі 100 посилань не може вважатися повноцінним benchmark-набором, який можна часто зустріти для задач класифікації. Пов'язано це з тим, що рекламні креативи не можна поділити на конкретну множину класів, тому і створити такі промарковані датасети доволі складно. Ще ускладнює оцінку те, що відповідь подається у вигляді тексту, модель може написати іншою мовою, або не дописати букву, це утворює доволі багато нюансів щодо оцінки якості класифікації, тому так важлива ручна експертиза.

Вибірка зі 100 посилань дозволяє оцінити результати системи в межах цієї роботи, однак для більш точного статистичного аналізу варто збільшити кількість розмічених даних і залучити ще кількох незалежних експертів.

3.8 Висновки до розділу 3

У третьому розділі було розібрано архітектуру системи для обробки та класифікації рекламних креативів. Було описано, як працює модуль вивантаження креативів, після чого розібрано модуль обробки та класифікації. Визначено, що вхідними даними є множина посилань L , а вихідними даними є множина результатів обробки та класифікації R .

Було побудовано та детально розібрано функціональну схему та sequence diagram. Також було досліджено роботу алгоритмів обробки

зображень та відео, а після побудовано їх прості схеми. Для пошуку дублікатів були використані рHash, embeddings та модель CLIP, і розбір на кадри.

В кінці систему було запущено та взято 400 посилань, з яких 300 змогли обробитися. Після запуску двох ітерацій було порівняно результати, стало видно, що метод збереження посилань в історію суттєво зменшив навантаження на систему. Алгоритм дедуплікації не дав значного зменшення креативів для обробки, оскільки вибірка була не велика. В цілому було видно доволі хороший відсоток класифікованих за індикатором $\delta(y_i) = 1$.

Додатково було проведено ручне оцінювання контрольної вибірки зі 100 посилань. За результатом перевірки Brand збігався з ручною оцінкою у 76% випадків, Product – у 79%, Category – у 88%. Одночасно всі три сутності були визначені правильно у 61% випадків. Метрики індикатора придатності становили Accuracy = 0.90, Precision = 0.86, Recall = 1.00 та F1-score = 0.93.

Отримані результати показують, що реалізована система здатна автоматизувати значну частину процесу обробки рекламних креативів для їх подальшого структурування. Також важливо, що якість класифікації залежить від інформативності самого креативу, наявності там бренду або продукту, а також від можливостей обраної мультимодальної моделі та сформованої текстової інструкції для неї.

РОЗДІЛ 4 ФУНКЦІОНАЛЬНО-ВАРТІСНИЙ АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

У цьому розділі буде проведено оцінювання основних характеристик програмного забезпечення, а саме системи автоматизованої мультимодальної класифікації рекламних креативів на основі аналізу зображень та відео.

Функціонально-вартісний аналіз є системним методом дослідження, спрямованим на оптимізацію співвідношення між споживчою цінністю (корисністю) об'єкта та сукупними витратами на його розробку і впровадження. Ключове завдання цього підходу полягає в мінімізації грошових та часових інвестицій за умови повного збереження (або підвищення) заданої ефективності та якості функціонування системи.

Інформаційним фундаментом методу є поєднання техніко-конструкторських та економічних показників об'єкта. Завдяки такому всебічному охопленню стає можливим точне структурування витрат, що дозволяє акцентувати увагу на розробці найбільш рентабельних та корисних функцій системи із низькою собівартістю.

Алгоритм функціонально-вартісного аналізу включає в себе визначення послідовності етапів розробки продукту, далі йде визначення повних річних витрат та кількості робочих годин, а також визначення джерел витрат. Останнім кроком є остаточний розрахунок вартості створення програмного продукту.

4.1 Постановка задачі проектування

Застосування функціонально-вартісного аналізу в цій роботі спрямоване на техніко-економічне обґрунтування програмного продукту для

автоматизованої мультимодальної класифікації рекламних креативів. Ефективність фінального продукту залежить від синергії всіх його складових, тому кожна підсистема аналізується на відповідність індивідуальним вимогам. Для забезпечення конкурентоспроможності розроблений застосунок має відповідати наступному комплексу технічних вимог:

1. Ергономічний та зрозумілий користувацький інтерфейс.
2. Можливість роботи на стандартному ПК користувача.
3. Обробка великих масивів рекламних креативів із формуванням структурованого звіту.
4. Модульність структури для легкого масштабування та супроводу.
5. Оптимальна собівартість розробки та впровадження програмного продукту.

4.2 Обґрунтування функцій програмного продукту

Головна функція F_0 – розробка програмного продукту для автоматизованої мультимодальної класифікації рекламних креативів, який забезпечує вивантаження рекламних матеріалів, обробку банерних зображень і відеокреативів, пошук дублікатів, використання історії попередніх класифікацій та формування структурованого звіту. Беручи дану функцію як основну, можна виділити наступні функції програмного застосунку:

F_1 – вибір мови програмування;

F_2 – вибір способу отримання результату класифікації;

F_3 – вибір способу розгортання програмного продукту.

Кожна з представлених функцій має декілька варіантів для реалізації.

Функція F_1 :

- 1) (а) мова програмування Python;
- 2) (б) мова програмування C++.

Функція F_2 :

- 1) (а) використання зовнішньої мультимодальної моделі через API;
- 2) (б) розробка власної моделі класифікації.

Функція F_3 :

- 1) (а) локальний desktop-застосунок;
- 2) (б) веб-застосунок.

Варіанти реалізації основних функцій наведені у морфологічній карті системи (рисунок 4.1).

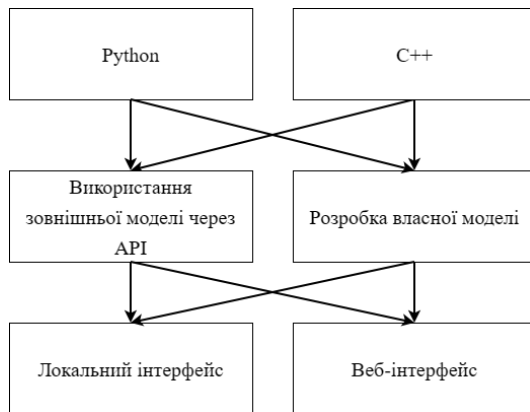


Рисунок 4.1 – Морфологічна карта

Дана морфологічна карта відображає всі можливі комбінації варіантів для наведених основних функцій. На основі даної карти будується позитивно-негативна матриця, яка наведена в таблиці 4.1.

Таблиця 4.1 – Позитивно-негативна матриця

Функції	Варіанти реалізації	Переваги	Недоліки
F_1	А	Простий синтаксис, велика кількість бібліотек з зрозумілими детальними документаціями для роботи з таблицями та векторами даних.	Велика витрата операційної пам'яті, інтерпретована мова, що вилітає через помилки в вже працюючому додатку, низька швидкість виконання.

Продовження таблиці 4.1

F_1	Б	Максимальна продуктивність та швидкість, контроль над пам'яттю та ресурсами комп'ютера.	Довгий і складний процес розробки, відсутність автоматичної безпеки типів і пам'яті.
F_2	А	Не потрібно самостійно навчати модель; швидше впровадження.	Є залежність від зовнішнього API, інтернет-з'єднання та вартості запитів.
	Б	Повний контроль над моделлю та логікою класифікації.	Потрібна велика навчальна вибірка та ресурси для навчання.
F_3	А	Просте локальне розгортання, зручний запуск користувачем, не потрібне серверне середовище.	Менш зручно для багатокористувацької роботи.
	Б	Зручний доступ через браузер, краще підходить для багатьох користувачів.	Потрібні сервер, база даних, хостинг і складніше адміністрування.

Результати аналізу позитивно-негативної матриці дозволяють обґрунтовано виключити певні варіанти реалізації функцій із подальшого розгляду. Було встановлено, що ці альтернативи не забезпечують виконання ключових завдань, поставлених перед проєктуємою системою, і є неефективними для її фінальної архітектури.

Для функції F_1 виключимо варіант б, оскільки використання C++ ускладнює розробку та підтримку системи, а також менш зручне для інтеграції

з бібліотеками обробки зображень, відео, таблиць та API. Для функції F_2 доцільно залишити варіант а, оскільки у межах цієї роботи не виконується самостійне навчання нейронної мережі. Класифікація рекламних креативів реалізується шляхом звернення до вже навченої зовнішньої мультимодальної моделі через API. Для функції F_3 доцільно залишити варіант а, оскільки локальний desktop-застосунок відповідає фактичній реалізації програмного продукту, простіший у розгортанні та не потребує серверної інфраструктури.

Таким чином, будемо розглядати такий варіант реалізації програмного продукту:

$$F_1a - F_2a - F_3a$$

4.3 Обґрунтування системи параметрів програмного продукту

Спираючись на результати аналізу, проведеного у попередньому підрозділі, було сформовано систему базових критеріїв для оцінювання ступеня функціональної реалізації програмного продукту. Зазначені параметри виступають основою для розрахунку коефіцієнта технічного рівня, що дозволяє забезпечити об'єктивний порівняльний аналіз альтернативних варіантів побудови програмної системи автоматизованої класифікації рекламних креативів. Для подальшого моделювання та вибору оптимального архітектурного рішення було виокремлено такі показники:

- 1) x_1 – середній час обробки одного креативу;
- 2) x_2 – об'єм оперативної пам'яті;
- 3) x_3 – час розробки коду (трудомісткість розробки системи);
- 4) x_4 – об'єм програмного коду.

Базові відмітки технічного рівня параметрів (гірші, середні та кращі) сформовано на основі технічного завдання замовника та умов розгортання

програмного застосунку. Відповідні градації показників узагальнено в таблиці 4.2.

Таблиця 4.2 – Параметри програмного продукту

Назва	Умовні позначення	Одиниці виміру	Значення параметра		
			Гірші	Середні	Кращі
Час відгуку системи	x_1	Секунди	5	2	0,5
Об'єм оперативної пам'яті	x_2	Мегабайти	2500	1200	500
Час розробки коду	x_3	Години	300	160	80
Об'єм програмного коду	x_4	Кількість рядків	12000	7000	3500

За заданими значеннями в таблиці 4.2 будуються графіки характеристики параметрів (рисунки 4.2–4.5).

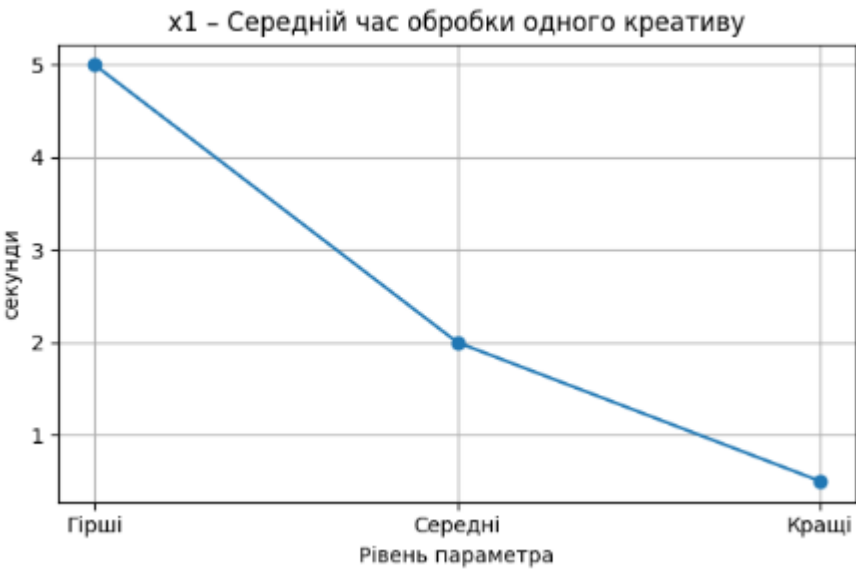


Рисунок 4.2 – Час відгуку системи

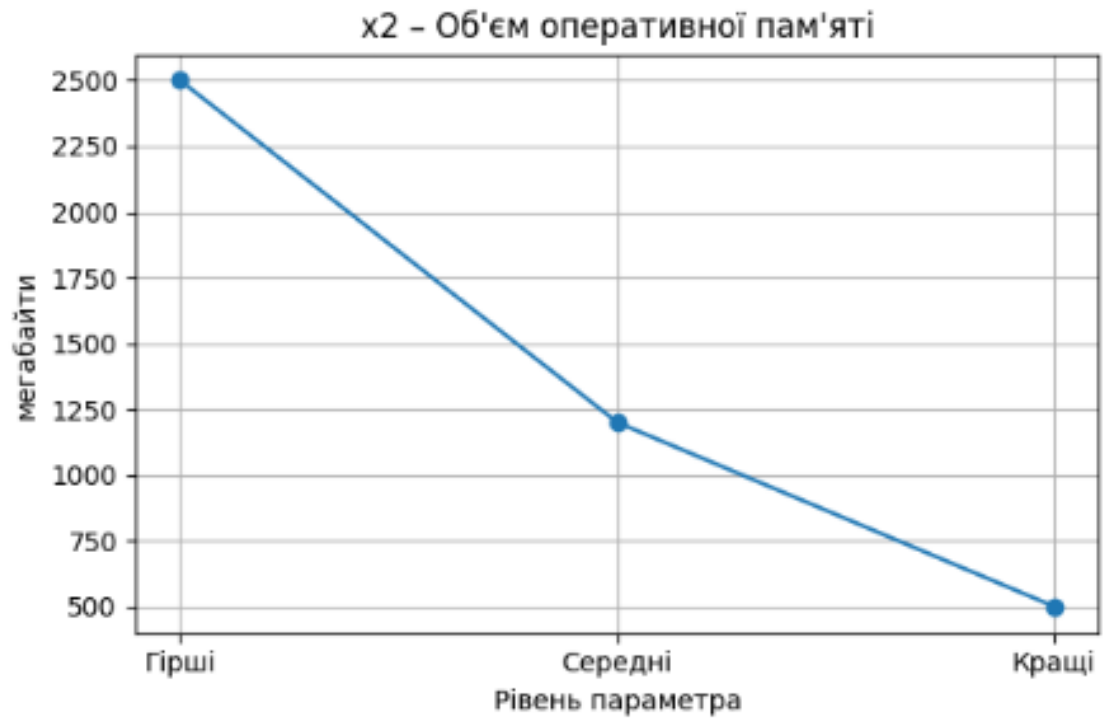


Рисунок 4.3 – Об'єм оперативної пам'яті



Рисунок 4.4 – Час розробки коду

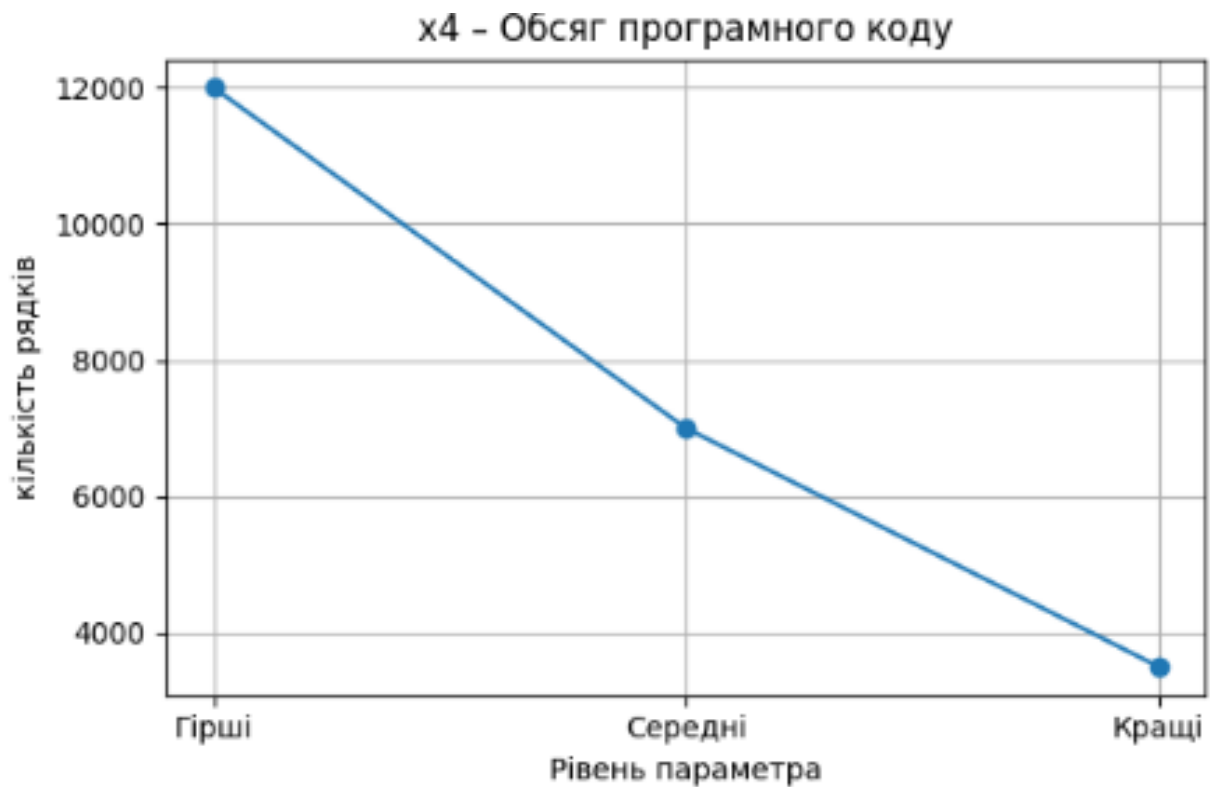


Рисунок 4.5 – Об’єм програмного коду

4.4 Аналіз експертного оцінювання параметрів

Важливість кожного визначеного техніко-економічного параметра програмної системи автоматизованої класифікації рекламних креативів встановлюється за допомогою методу попарного порівняння. Оцінювання здійснюється експертною групою, до складу якої було залучено 7 фахівців: системних аналітиків, розробників програмного забезпечення та фахівців з аналізу рекламних даних.

Після аналізу специфіки розгортання системи, зокрема обробки банерних зображень, відеокреативів, використання історії класифікацій, пошуку дублікатів та звернення до зовнішнього API, кожен експерт визначив рівень пріоритетності параметрів відповідно до поставленої мети.

Процес встановлення вагових коефіцієнтів параметрів включає виконання таких основних етапів.

1. Визначення рівня значущості кожного параметра ($x_1 - x_4$) для ефективного функціонування програмного продукту.
2. Перевірка експертних оцінок на придатність, внутрішню узгодженість та використання.
3. Визначення оцінки попарного пріоритету параметрів.
4. Обробка результатів і обчислення фінальних коефіцієнтів значущості (вагомості).

Щоб перевірити придатність експертних оцінок, визначаються відповідні статистичні показники. Сума рангів для кожного параметра розраховується за формулою:

$$R_i = \sum_{j=1}^N r_{ij}$$

де r_{ij} – ранг, присвоєний i -му параметру програмного продукту j -м експертом;

N – загальна кількість експертів ($N = 7$).

Загальна сума рангів обчислюється за такою формулою:

$$R = \frac{Nn(n+1)}{2}$$

де N – загальна кількість експертів; n – загальна кількість параметрів.

Середня сума рангів обчислюється за наступною формулою:

$$T = \frac{R}{n}$$

Для обчислення відхилення суми рангів кожного параметра від середньої суми рангів застосовується наступна формула:

$$\Delta_i = R_i - T$$

При чому сума відхилень по всіх параметрах має дорівнювати 0.

Загальну суму квадратів відхилення розраховують наступним чином:

$$S = \sum_{i=0}^N \Delta_i^2$$

Після обчислень отримали результати експертного ранжування, які наведені в таблиці 4.3.

Таблиця 4.3 – Результати експертного ранжування

Позначення параметра	Ранг параметра за оцінкою експерта							R_i	Δ_i	Δ_i^2
	1	2	3	4	5	6	7			
x_1	1	1	2	1	1	2	1	9	-8.5	72.25
x_2	3	2	3	2	3	1	2	16	-1.5	2.25
x_3	2	3	1	3	2	3	3	17	-0.5	0.25
x_4	4	4	4	4	4	4	4	28	10.5	110.25
Загальна кількість	10	10	10	10	10	10	10	70	0	185

Наступним кроком рахуємо коефіцієнт достовірності за формулою:

$$W = \frac{12S}{N^2 n^2 (n-1)} = \frac{12 \cdot 185}{7^2 \cdot 4^2 \cdot (4-1)} \approx 0.94 > W_k = 0.67$$

Ранжування можна вважати успішним, адже знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжування, проведемо попарне порівняння всіх параметрів та результати занесемо до таблиці 4.4.

Таблиця 4.4 – Попарне порівняння

Параметри		Експерти							Кінцева оцінка	Числове значення
		1	2	3	4	5	6	7		
x_1	x_2	>	>	>	>	>	<	>	>	1,5
x_1	x_3	>	>	<	>	>	>	>	>	1,5
x_1	x_4	>	>	>	>	>	>	>	>	1,5
x_2	x_3	<	>	<	>	<	>	>	>	1,5
x_2	x_4	>	>	>	>	>	>	>	>	1,5
x_3	x_4	>	>	>	>	>	>	>	>	1,5

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається за формулою:

$$a_{ij} = \begin{cases} 1.5, \text{ при } X_i > X_j \\ 1, \text{ при } X_i = X_j \\ 0.5, \text{ при } X_i < X_j \end{cases}$$

З отриманих оцінок переваги складається матриця $A = \|a_{ij}\|$. Для кожного параметра зробимо розрахунок вагомості K_b за наступними формулами:

$$K_b = \frac{b_i}{\sum_{i=1}^n b_i}$$

$$b_i = \sum_{i=1}^N a_{ij}$$

Ітерації потрібно робити до поки різниця значень коефіцієнтів не буде нижчою 2% (таблиця 4.5).

Таблиця 4.5 – Обчислення вагомості параметрів

Параметри x_i	Параметри x_j				Перша ітерація		Друга ітерація		Третя ітерація	
	x_1	x_2	x_3	x_4	b_i	K_{b_i}	b_i^1	$K_{b_i}^1$	b_i^2	$K_{b_i}^2$
x_1	1	1.5	1.5	1.5	5.5	0.3438	21.25	0.3602	77.875	0.3605
x_2	0.5	1	1.5	1.5	4.5	0.2813	16.25	0.2754	59.125	0.2737
x_3	0.5	0.5	1	1.5	3.5	0.2188	12.25	0.2076	44.875	0.2078
x_4	0.5	0.5	0.5	1	2.5	0.1563	9.25	0.1568	34.125	0.158
Сума	-	-	-	-	16	1	59	1	216	1

На третій ітерації різниця значень коефіцієнтів менше ніж 2%, тому більше ітерацій робити не потрібно.

4.5 Аналіз рівня якості варіантів реалізації функції

Для порівняльного аналізу архітектурних рішень здійснюється окреме оцінювання рівня реалізації кожної базової функції програми. Зведені техніко-економічні показники варіантів розробки наведено в таблиці 4.6, де також розраховується узагальнений коефіцієнт технічного рівня. Обчислення цього коефіцієнта виконується за формулою:

$$K_K(j) = \sum_{i=1}^n K_{b_{i,j}} B_{i,j}$$

Для функції F_2 враховано два параметри одночасно: x_1 , що характеризує час обробки одного креативу, та x_4 , що характеризує обсяг програмного коду. Це пов'язано з тим, що спосіб отримання результату класифікації впливає як на швидкість роботи системи, так і на складність її реалізації.

Таблиця 4.6 – Розрахунок показників рівня якості варіантів реалізації.

Функція	Варіант реалізації функції	Параметри	Абсолютне значення параметра	Бальна оцінка	Коефіцієнт вагомості	Коефіцієнт рівня якості
F_1	а	x_3	120	25	0,2078	5,195
	б	x_3	220	16	0,2078	3,3248
F_2	а	x_1	1,09	26	0,3605	9,373
		x_4	7000	20	0,158	3,16
	б	x_1	4	13	0,3605	4,6865
		x_4	11000	12	0,158	1,896
F_3	а	x_2	1200	20	0,2737	5,474
	б	x_2	1800	15	0,2737	4,1055

За отриманими даними визначаємо рівень якості кожного з варіантів за формулою:

$$K_K = K_{Ty}[F_{1K}] + K_{Ty}[F_{2K}] + \dots K_{Ty}[F_{zK}]$$

Визначаємо рівень якості кожного з варіантів:

$$K_{K1} = 5,1950 + 9,3730 + 3,1600 + 5,4740 = 23,2020;$$

$$K_{K2} = 3,3248 + 4,6865 + 1,8960 + 4,1055 = 14,0128.$$

Спираючись на отримані результати, можна зробити висновок, що перший варіант є кращим, оскільки він має більший коефіцієнт технічного рівня та відповідає фактичній реалізації програмного продукту.

4.6 Економічний аналіз варіантів розробки програмного продукту

Для визначення загальної вартості створення програмного продукту на початковому етапі необхідно розрахувати його трудомісткість. Процес розробки для всіх розглянутих варіантів конфігурації розподілено на два автономні завдання:

- 1) проєктування та розробка ядра обробки і класифікації рекламних креативів;
- 2) створення користувацької програмної оболонки (інтерфейсу).

За ступенем новизни перше завдання класифіковано як розробку групи А, тоді як друге завдання віднесено до групи Б. Відповідно до складності використовуваних алгоритмів, ядро обробки і класифікації рекламних креативів (завдання 1) належить до 1-ї групи складності, а програмна оболонка (завдання 2) – до 3-ї групи.

Довідкова інформація використовується для реалізації завдання 1, а для завдання 2 передбачено використання інформації у вигляді даних.

Далі виконаємо обчислення норм часу на розробку та програмування для кожного з завдань. Загальна трудомісткість обчислюється за формулою:

$$T_k = T_p \cdot K_{\Pi} \cdot K_{СК} \cdot K_M \cdot K_{СТ} \cdot K_{СТ.М}$$

де T_p – трудомісткість розробки програмного продукту;

K_{Π} – поправочний коефіцієнт;

$K_{СК}$ – коефіцієнт на складність вхідної інформації;

K_M – коефіцієнт рівня мови програмування;

$K_{СТ}$ – коефіцієнт використання стандартних модулів і прикладних програм;

$K_{СТ.М}$ – коефіцієнт стандартного математичного забезпечення.

Для першого завдання (розробка розрахунково-аналітичного ядра системи), на основі нормативів часу для задач зі ступенем новизни А та першої групи складності алгоритму, базова трудомісткість становить: $T_p = 22$ людино-дні. Щоб врахувати специфіку нормативно-довідкової інформації (посилання на рекламні креативи, службові файли, банерні зображення, відеокреативи та історія класифікацій), що використовується, при реалізації завдання, вводиться поправковий коефіцієнт: $K_{\Pi} = 1.04$. Поправковий коефіцієнт, який враховує складність контролю вхідної та вихідної інформації, рівний 1. Оскільки при розробці математичного ядра використовуються оптимізовані інструменти мови Python, врахуємо це за допомогою коефіцієнта інструментальних засобів $K_{СТ} = 0.9$. Тоді загальна трудомісткість програмування першого завдання дорівнює:

$$T_1 = 22 \cdot 1,04 \cdot 1 \cdot 0,9 = 20,592 \text{ людино-днів.}$$

Наступним кроком є проведення аналогічних обчислень для проєктування користувацької оболонки застосунку на базі локального графічного інтерфейсу. Оскільки програмна оболонка містить велику кількість подій, умов та елементів інтерфейсу, для цього завдання використовується алгоритм третьої групи складності та ступінь новизни Б. Таким чином: $T_p = 14$ людино-днів, $K_{\Pi} = 0,8$, $K_{СК} = 1$, $K_{СТ} = 0,9$. Тоді загальна трудомісткість завдання буде дорівнювати:

$$T_2 = 14 \cdot 0,8 \cdot 1 \cdot 0,9 = 10,08 \text{ людино-днів.}$$

Складаємо сумарну трудомісткість відповідних завдань для кожного з обраних варіантів реалізації програми, щоб отримати їхні повні значення в годинах (з урахуванням коефіцієнтів технічного рівня якості та офіційно затвердженого в Україні 8-годинного робочого дня):

$$T_I = (20,592 + 10,08 + 23,2020 + 10,08) \cdot 8 = 511,6320 \text{ людино-годин};$$

$$T_{II} = (20,592 + 10,08 + 14,0128 + 10,08) \cdot 8 = 438,1184 \text{ людино-годин}.$$

Найбільшу високу трудомісткість має варіант I.

Нехай, в розробці беруть участь два розробники з окладом 38 000 грн та один аналітик даних з окладом 55 000 грн. Визначимо середню зарплату за годину за формулою:

$$C_q = \frac{M}{T_m \cdot t}$$

де M – місячний оклад працівників;

T_m – кількість робочих днів на місяць;

t – кількість робочих годин в день.

Отримана середня зарплата після обчислення дорівнює:

$$C_q = \frac{38000 + 38000 + 55000}{3 \cdot 21 \cdot 8} = 259,92 \text{ грн}.$$

Після отримання середньої зарплати на годину розрахуємо заробітну плату за варіантами за формулою:

$$C_{3П} = C_q \cdot T_i \cdot K_d$$

де C_q – величина погодинної оплати праці програміста;

T_i – трудомісткість відповідного завдання;

K_d – норматив, який враховує додаткову заробітну плату.

Зарплата працівників за варіантами становить:

$$C_{зп, I} = 259,92 \cdot 511,6320 \cdot 1,3 = 172\,878,83 \text{ грн};$$

$$C_{зп, II} = 259,92 \cdot 438,1184 \cdot 1,3 = 148\,038,82 \text{ грн.}$$

Наступним кроком є нарахування єдиного соціального внеску (ЄСВ). Відповідно до чинного законодавства України, ставка нарахування на фонд оплати праці розробників становить 22%:

$$C_{від, I} = C_{зп, I} \cdot 0,22 = 172\,878,83 \cdot 0,22 = 38\,033,34 \text{ грн};$$

$$C_{від, II} = C_{зп, II} \cdot 0,22 = 148\,038,82 \cdot 0,22 = 32\,568,54 \text{ грн.}$$

Наступним кроком є розрахунок витрат на експлуатацію та оплату однієї машино-години роботи обчислювальної техніки. Враховуючи, що одне персональне комп'ютерне місце закріплене за одним розробником із місячним окладом 38 000 грн (М) та коефіцієнтом його зайнятості (K_3) на рівні 0.2, базові витрати для однієї робочої станції визначаються за формулою:

$$C_{\Gamma} = 12 \cdot M \cdot K_3 = 12 \cdot 38\,000 \cdot 0,2 = 91\,200 \text{ грн.}$$

З урахуванням додаткової заробітної плати:

$$C_{зп} = C_{\Gamma} (1 + K_3) = 91\,200 \cdot (1 + 0,2) = 109\,440 \text{ грн.}$$

Відрахування на соціальний внесок:

$$C_{від} = C_{зп} \cdot 0,22 = 109\,440 \cdot 0,22 = 24\,076,80 \text{ грн.}$$

Амортизаційні відрахування розраховуємо при річній нормі амортизації у 25% (K_A) та вартості ЕОМ – 30000 грн ($C_{\text{ПР}}$). Позначимо коефіцієнт вартості додаткових технічних матеріалів (наприклад, мишка, клавіатура) як 1.3 ($K_{\text{ТМ}}$):

$$C_A = C_{\text{ПР}} \cdot K_A \cdot K_{\text{ТМ}} = 24\,000 \cdot 0,25 \cdot 1,3 = 7\,800 \text{ грн.}$$

Витрати на ремонт та профілактику розраховуємо як:

$$C_P = C_{\text{ПР}} \cdot K_P \cdot K_{\text{ТМ}} = 24\,000 \cdot 0,05 \cdot 1,3 = 1\,560 \text{ грн}$$

де K_P – відсоток витрат на поточні ремонти.

Ефективний годинний фонд часу ПК за рік обчислюється за формулою:

$$T_{\text{ЕФ}} = (D_K - D_B - D_C - D_P) \cdot t_3 \cdot K_B = (236) \cdot 8 \cdot 0,93 = 1\,755,84 \text{ години.}$$

де D_K – календарна кількість днів у році (в 2026 - 365);

D_B – кількість вихідних днів (в 2026 році -103);

D_C – кількість святкових днів (під час дії воєнного стану таке майже не передбачено, візьмемо 2);

D_P – кількість днів планових ремонтів устаткування (зазвичай 24 календарні дні);

t_3 – кількість робочих годин в день;

K_B – коефіцієнт використання приладу у часі протягом зміни.

Витрати на оплату електроенергії обчислюється за формулою:

$$C_{\text{ЕЛ}} = T_{\text{ЕФ}} \cdot N_C \cdot K_3 \cdot C_{\text{ЕН}} = 1\,755,84 \cdot 0,25 \cdot 0,8 \cdot 13,64 = 4\,789,93 \text{ грн.}$$

де N_C – середньо-споживча потужність приладу;

K_3 – коефіцієнт зайнятості приладу;

$C_{\text{ЕН}}$ – тариф за 1 кВт-годин електроенергії.

Накладні витрати (складають 67%) обчислюємо за формулою:

$$C_H = C_{\text{ПР}} \cdot 0,67 = 24\,000 \cdot 0,67 = 16\,080 \text{ грн.}$$

Тоді річні витрати обчислюємо за формулою:

$$C_{\text{ЕКС}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_A + C_P + C_{\text{ЕЛ}} + C_H$$

$$\begin{aligned} C_{\text{ЕКС}} &= 109440 + 24076,80 + 7800 + 1560 + 4789,93 + 16080 = \\ &= 163746,73 \text{ грн} \end{aligned}$$

Собівартість однієї машино-години ПК дорівнюватиме:

$$C_{\text{М-Г}} = \frac{C_{\text{ЕКС}}}{T_{\text{ЕФ}}} = \frac{163\,746,73}{1\,755,84} = 93,26 \text{ грн/год.}$$

Оскільки в цьому випадку всі роботи, які пов'язані з розробкою програмного продукту ведуться на ЕОМ, витрати на оплату машинного часу, в залежності від обраного варіанта реалізації, обчислюється за формулою:

$$C_M = C_{\text{М-Г}} \cdot T$$

$$C_{\text{М,І}} = C_{\text{М-Г}} \cdot T_I = 93,26 \cdot 511,6320 = 47\,713,95;$$

$$C_{\text{М,ІІ}} = C_{\text{М-Г}} \cdot T_{II} = 93,26 \cdot 438,1184 = 40\,858,20.$$

Накладені витрати складають 67% та обчислюються за формулою:

$$C_H = C_{\text{ЗП}} \cdot 0,67$$

$$C_{\text{Н,І}} = 145\,162,79 \cdot 0,67 = 97\,259,07 \text{ грн;}$$

$$C_{\text{Н,ІІ}} = 124\,305,14 \cdot 0,67 = 83\,284,45 \text{ грн.}$$

На основі комплексу розрахованих вище техніко-економічних метрик та інфраструктурних витрат стає можливим визначити повну собівартість розробки програмного продукту. Порівняльний розрахунок фінальних витрат для кожного з розглянутих варіантів конфігурації здійснюється за такою математичною залежністю:

$$C_{\text{ПП}} = C_{\text{ЗП}} + C_{\text{ВІД}} + C_{\text{М}} + C_{\text{Н}}$$

$$C_{\text{ПП, I}} = 172\,878,83 + 38\,033,34 + 47\,713,95 + 115\,828,82 = 374\,454,94 \text{ грн};$$

$$C_{\text{ПП, II}} = 148\,038,82 + 32\,568,54 + 40\,858,20 + 99\,186,01 = 320\,651,56 \text{ грн}.$$

4.7 Вибір кращого варіанту ПП техніко-економічного рівня

Розрахуємо коефіцієнт техніко-економічного рівня за формулою:

$$K_{\text{ТЕР}_j} = \frac{K_{K_j}}{C_{\Phi_j}}$$

$$K_{\text{ТЕР}_1} = \frac{23,2020}{374\,454,94} = 6,20 \cdot 10^{-5};$$

$$K_{\text{ТЕР}_2} = \frac{14,0128}{320\,651,56} = 4,37 \cdot 10^{-5}.$$

Як бачимо, найбільш ефективним є перший варіант реалізації програми з коефіцієнтом техніко-економічного рівня $6,20 \cdot 10^{-5}$.

На основі фінальних розрахунків ФВА було здійснено порівняльний аналіз альтернатив, що пройшли попередній відбір. Порівняння підтвердило,

що перша конфігурація програмного продукту є найбільш раціональною з економічного та інженерного погляду. Даний варіант забезпечує максимальне значення коефіцієнта техніко-економічного рівня, що робить його впровадження найбільш доцільним.

Обрана конфігурація системи базується на використанні мови програмування Python, зовнішньої мультимодальної моделі через API та локального desktop-застосунку. Такий підхід відповідає фактичній реалізації програмного продукту, забезпечує зручну взаємодію користувача із системою та спрощує локальне розгортання. При цьому інтернет-з'єднання потрібне для звернення до API під час класифікації нових рекламних креативів.

4.8 Висновки до розділу 4

У межах даного розділу виконано функціонально-вартісний аналіз програмного продукту для автоматизованої мультимодальної класифікації рекламних креативів. Було визначено основні функції системи, побудовано морфологічну карту та сформовано позитивно-негативну матрицю варіантів реалізації.

На основі експертного оцінювання розраховано коефіцієнти вагомості параметрів, рівень якості варіантів та витрати на розробку програмного продукту. За результатами розрахунку коефіцієнта техніко-економічного рівня найбільш доцільним визначено перший варіант реалізації: Python, зовнішня мультимодальна модель через API та локальний desktop-застосунок.

ВИСНОВКИ

У рамках цієї дипломної роботи було розглянуто задачу автоматизованої мультимодальної класифікації рекламних креативів. Після чого розроблено систему, яка перетворює неструктуровані рекламні матеріали, а саме банерні зображення та відео, у структурований результат для майбутнього аналізу. Для класифікації використовувалась зовнішня мультимодальна модель через API сервісу OpenAI [14].

У процесі розробки було проаналізовано рекламний креатив як мультимодальний об'єкт, який містить у собі багато різноманітної інформації: від тексту та логотипу, до простих об'єктів сцени та семантичного змісту. Пояснено чому підходи типу OCR, Object Detection або ж класичний Image Classification не підходили для реалізації поточної задачі, так як для розуміння реклами потрібно було бачити семантичний зміст рекламного матеріалу, щоб розуміти до чого відноситься та чи інша сутність на зображенні.

Було побудовано математичну модель системи, формалізовано вхідні дані у вигляді множини посилань, а після – множини креативів, яка була отримана за допомогою функції вивантаження посилань на рекламні матеріали. Залежно від того чи є креатив статичним (зображення) чи динамічним (відео), впроваджено відповідні функції обробки та класифікації.

Результатом класифікації встановлено сутності Brand, Product та Category. В кінці роботи системи генерується файл звіту із класифікацією рекламних матеріалів, оновлюється файл історії додаючи нові придатно класифіковані креативи за визначеним індикатором, а також рахуються операційні метрики результатів обробки.

Систему було реалізовано на мові програмування Python [25] із урахуванням загальних рекомендацій PEP 8 [26] та з використанням бібліотеки для роботи з графічним інтерфейсом Custom Tkinter, бібліотек для обробки даних PIL, imagehash, cv2 та інших. До того ж застосовувався

інструментарій API від OpenAI для підключення та роботи з хмарними мультимодальними моделями. Згенеровані системою файли звіту, історії та метрик створюються у форматі .csv для структурованого зберігання великих об'ємів даних. Розроблене програмне забезпечення було викладено у відкритий доступ у репозиторії на онлайн-платформі GitHub [27].

У межах 400 посилань, з яких вивантажилося та пройшло обробку 300 креативів, а інші 100 посилань виявилися недоступними, частка результативно класифікованих за індикатором була 229, з яких 220 зображень та 9 відео. Отримані результати показують, що система доволі швидко обробляла файли креативів на першій ітерації, у свою чергу на другій ітерації кількість оброблюваних файлів зменшилась на 209 креативів, що вказує на успішне занесення частини придатно класифікованих результатів за індикатором отриманих з першої ітерації в історію.

Окремо було оцінено роботу індикатора придатності класифікації, а також якості класифікації системи на вибірці зі 100 рекламних посилань, для яких була виконана ручна експертна розмітка після їх обробки системою. Після отримання результатів було побудовано матрицю невідповідностей, а також пораховано базові метрики Accuracy, Precision, Recall, F1-score. Було видно, що 90 креативів зі 100, які були оцінені індикатором як придатні чи непридатні, були оцінені правильно, в інших 10 класифікація була невірною і тому індикатор теж спрацював некоректно. Метрики Accuracy складала 0.90, Precision була приблизно 0.86, Recall дорівнювала 1.00, а F1-score складала 0.93. Це свідчить, що в цілому індикатор працював коректно, оскільки класифікація проходила в більшості випадків правильно. Після експертної оцінки окремо кожну з трьох сутностей можна було побачити, що бренд визначався правильно у 76%, продукт визначився правильно у 79%, а категорія взагалі у 88%. Такий великий відсоток у категорії пов'язаний з тим, що це більш широке поняття, для якого не треба знаходити конкретної назви чи логотипу на креативі.

Подальший розвиток розробленої системи може бути підвищенням підвищення точності класифікації та розширення алгоритмів обробки креативів. Доцільно вдосконалити методи обробки відео, а саме вибір ключових кадрів, для виділення тих кадрів, де є корисна інформація, та можливо використання додаткової інформації з аудіо доріжки. Наприклад, можна перетворювати аудіо відеокреативу у текст, а після додавати до промпту для більшого семантичного наповнення для класифікації. Також одним з важливих покращень може бути вдосконалення методів пошуку дублікатів з подальшим зменшенням навантаження на систему при їх обробці. Ще для більш глибокого аналізу якості роботи системи варто залучити декілька незалежних експертів для розмітки більшого об'єму посилань та перевірки результатів класифікації.

Отже, ця дипломна робота показує, що розробка та впровадження таких систем мультимодальної класифікації креативів є доволі цікавим та актуальним напрямком для подальших досліджень. Рекламні матеріали не можна розділити на конкретну множину класів, що унеможлиблює стандартні методи класифікації, а різноманіття корисної інформації на креативі та необхідність розуміння його зміст змушує відповідні використовувати мультимодальні моделі, які на вхід можуть приймати зображення та текстову інструкцію. Для великих об'ємів даних обробка стає занадто довгою та ресурсномісткою, тому треба також не забувати про методи, які зменшують навантаження на систему та час її роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Redel R. A., Zietz D., Uchikawa V., Pfitscher R. J. Analyzing the Influence of Online Advertisements on ISP Network Traffic. *Anais do XXIX Workshop de Gerência e Operação de Redes e Serviços*. 2024. P. 57–69. DOI: 10.5753/wgrs.2024.3235.
2. Hussain Z., Zhang M., Zhang X., Ye K., Thomas C., Agha Z., Ong N., Kovashka A. Automatic Understanding of Image and Video Advertisements. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA, 2017. P. 1705–1715. DOI: 10.1109/CVPR.2017.123.
3. Ващило О., Андрусів Ю. Мультиmodalність сучасного англomовного рекламного тексту. *Advanced Linguistics*. 2024. № 14. С. 48–55. DOI: 10.20535/.2024.14.315521.
4. Bose D., Hebbar R., Feng T., Somandepalli K., Xu A., Narayanan S. MM-AU: Towards Multimodal Understanding of Advertisement Videos. *Proceedings of the 31st ACM International Conference on Multimedia*. New York, NY: Association for Computing Machinery, 2023. P. 86–95. DOI: 10.1145/3581783.3612371.
5. Weng Z., Meng L., Wang R., Wu Z., Jiang Y.-G. A Multimodal Framework for Video Ads Understanding. *Proceedings of the 29th ACM International Conference on Multimedia*. New York, NY: Association for Computing Machinery, 2021. P. 4843–4847. DOI: 10.1145/3474085.3479202.
6. Oktavia F. P., Sudarto S., Erdiansyah R. Soundless Video Advertising and Purchase Decisions: An Experiment Study With and Without Sound. *Proceedings of the International Conference on Economics, Business, Social, and Humanities (ICEBSH 2021)*. Atlantis Press, 2021. P. 795–799. DOI: 10.2991/assehr.k.210805.125.
7. Smith R. An Overview of the Tesseract OCR Engine. *Proceedings of the Ninth International Conference on Document Analysis and Recognition*

- (ICDAR 2007). IEEE Computer Society, 2007. Vol. 2. P. 629–633. DOI: 10.1109/ICDAR.2007.4376991.
8. Zou Z., Chen K., Shi Z., Guo Y., Ye J. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*. 2023. Vol. 111, No. 3. P. 257–276. DOI: 10.1109/JPROC.2023.3238524.
 9. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*. 2012. Vol. 25. P. 1097–1105. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf> (last accessed: 31.05.2026).
 10. Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., Clark J., Krueger G., Sutskever I. Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021. Vol. 139. P. 8748–8763. URL: <https://proceedings.mlr.press/v139/radford21a.html> (last accessed: 31.05.2026).
 11. Zhang H., Luo Y., Ai Q., Wen Y., Hu H. Look, Read and Feel: Benchmarking Ads Understanding with Multimodal Multitask Learning. *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY: Association for Computing Machinery, 2020. P. 430–438. DOI: 10.1145/3394171.3413582.
 12. Bavaresco A., Testoni A., Fernández R. Don't Buy it! Reassessing the Ad Understanding Abilities of Contrastive Multimodal Models. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*. 2024. P. 870–879. DOI: 10.18653/v1/2024.acl-short.77.
 13. Rumiantsev O., Oliinyk Y. Evaluation of the Effectiveness of Two Approaches to Building Damage Detection with Satellite Imagery. *Information, Computing and Intelligent Systems*. 2025. No. 7. P. 61–71. DOI: 10.20535/2786-8729.7.2025.341475.

14. OpenAI. API Pricing. URL: <https://openai.com/api/pricing/> (last accessed: 31.05.2026).
15. Zhou Z., Lin K., Cao Y., Yang C. N., Liu Y. Near-Duplicate Image Detection System Using Coarse-to-Fine Matching Scheme Based on Global and Local CNN Features. *Mathematics*. 2020. Vol. 8, No. 4. Article 644. DOI: 10.3390/math8040644.
16. Gusev A., Xu J. Evolution of a Web-Scale Near Duplicate Image Detection System. *Proceedings of The Web Conference 2020*. New York, NY: Association for Computing Machinery, 2020. P. 2733–2739. DOI: 10.1145/3366423.3380031.
17. McKeown S., Aaby P., Steyven A. PHASER: Perceptual Hashing Algorithms Evaluation and Results – An Open Source Forensic Framework. *Forensic Science International: Digital Investigation*. 2024. Vol. 48. Article 301680. DOI: 10.1016/j.fsidi.2023.301680.
18. Шабан Н. В. Система пошуку нечітких дублікатів зображення : магістерська дис. : 122 Комп'ютерні науки / Шабан Надія Володимирівна. Київ, КПІ імені Ігоря Сікорського, 2018. 117 с. URL: <https://ela.kpi.ua/handle/123456789/26122> (дата звернення: 31.05.2026).
19. Gemius. gemiusAdReal. URL: <https://adreal.gemius.com/> (last accessed: 31.05.2026).
20. Apostolidis E., Adamantidou E., Metsai A. I., Mezaris V., Patras I. Video Summarization Using Deep Neural Networks: A Survey. *Proceedings of the IEEE*. 2021. Vol. 109, No. 11. P. 1838–1863. DOI: 10.1109/JPROC.2021.3117472.
21. Спекторський І. Я. Дискретна математика : навч. посіб. 2-ге вид., виправл. і доповн. Київ: ІВЦ «Видавництво Політехніка», 2004. 220 с.
22. Недашківська Н. І. Інтелектуальний аналіз даних. Практикум: навч. посіб. для студентів спеціальності 124 «Системний аналіз», освітніх програм «Системний аналіз і управління», «Системний аналіз фінансового ринку». Київ: КПІ імені Ігоря Сікорського, 2021. 106 с.

- URL: <https://ela.kpi.ua/handle/123456789/53763> (дата звернення: 31.05.2026).
23. Powers D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. *Journal of Machine Learning Technologies*. 2011. Vol. 2, No. 1. P. 37–63. URL: https://bioinfopublication.org/files/articles/2_1_1_JMLT.pdf (last accessed: 31.05.2026).
24. Grandini M., Bagli E., Visani G. Metrics for Multi-Class Classification: An Overview. *arXiv preprint arXiv:2008.05756*. 2020. DOI: 10.48550/arXiv.2008.05756.
25. Python Software Foundation. Python 3 documentation. URL: <https://docs.python.org/3/> (last accessed: 31.05.2026).
26. Van Rossum G., Warsaw B., Coghlan A. PEP 8 – Style Guide for Python Code. *Python Enhancement Proposals*. URL: <https://peps.python.org/pep-0008/> (last accessed: 31.05.2026).
27. Парненко І. О. ai-creative-classifier-diploma. *GitHub*. URL: <https://github.com/Snooby55/ai-creative-classifier-diploma> (дата звернення: 11.06.2026).

ДОДАТОК А РЕЗУЛЬТАТИ КЛАСИФІКАЦІЇ

Таблиця А.1 – класифікація з файлу звіту (Посилання приховані через політику конфіденційності)

Link (hidden because of privacy)	Status	Product
*****	1	InfoCar.site - Headlight polishing service
*****	1	PixVerse V5.5 Model AI video generation tool
*****	1	Упаковочка - Packaging supplies
*****	1	KARMA - Drone frame components
*****	1	- DS Engines Game creation tool advertisement
*****	1	Opel Frontera Car
*****	1	VELO - Nicotine pouches
*****	1	myIQ - Online quiz ad
*****	1	- - Car headlight assembly
*****	1	VakıfBank - Banking service
*****	1	Hero Wars - Mobile puzzle ad
*****	1	Arbitrage UP Стартовий курс по арбітражу трафіку Online course
*****	1	hint - Online quiz/test
*****	1	- - Batteries
*****	1	- - Investment academy course
*****	1	- - Stock trading app
*****	1	Tvei Coffee Streblum? - Coffee
*****	1	ROBEAUTY reMAX Skincare cosmetics
*****	1	Multisearch - Sales/marketing tool advertisement
*****	1	- - Dental implants services
*****	1	OTTO - Online retailer
*****	1	Project Entropy - Mobile/PC strategy game ad
*****	1	JustClass - Online testing platform
*****	1	Yahoo Finance - Finance app ad
*****	1	CitySites - Senior living services
*****	1	Parimatch FIRST Sports betting
*****	1	Ivs Kolo - Investment services
*****	1	BST - -
*****	1	Mazda CX-3 Car
*****	1	Mladenis - Bedding set
*****	1	Raiffeisen Bank MyRaif Banking app
*****	1	- - Car accessory
*****	1	- - -
*****	1	- - Bell pepper
*****	1	Shop Pay - Payment/checkout service
*****	1	Zizlio - Used laptop
*****	1	- - Bee pollen granules
*****	1	- - Shower head

Продовження таблиці А.1

*****	1	Renault Kardian Car
*****	1	Hero Wars - Mobile game advertisement
*****	1	twocitizenship - Citizenship services
*****	1	Opera - Web browser
*****	1	Honda CR-V e:HEV Plug-in Hybrid SUV
*****	1	Постинор - Emergency contraception
*****	1	- - T-shirts
*****	1	- - Pomegranate saplings
*****	1	Mitsubishi Motors - Car
*****	1	Metkas - Storage shelving
*****	1	- - Flowers
*****	1	ROZETKA - Electronics retailer ad
*****	1	Apple iPhone 15 Pro Smartphone
*****	1	VELO VELO Nicotine pouch
*****	1	MILITARY.EU - Men's jacket
*****	1	- - Military technology training
*****	1	Stormshot From Ruins to Pirate Empire Mobile game
*****	1	Nissan Qashqai e-POWER Car
*****	1	Parabenum Parabenum.com.ua Website
*****	1	Taboola - Content recommendation platform
*****	1	- - Tool kits
*****	1	Suzuki Vitara Car
*****	1	ClickUp - Project management software
*****	1	Everest - Internet and cable TV service
*****	1	Renault Duster Car
*****	1	- - Commemorative coin
*****	1	Samsung Galaxy S25 Ultra Smartphone
*****	1	Parimatch - Sports betting
*****	1	- - Sauna/spa service
*****	1	- - Mobile game ad
*****	1	Sweet.tv - Streaming service
*****	1	- - Air freshener
*****	1	ClickUp - Project management software
*****	1	Justschool - Online language course
*****	1	777.ua - Online casino
*****	1	- - Dish drying rack
*****	1	Aristipp - Immigration/visa service
*****	1	Range Rover Evoque Car
*****	1	Kyivstar Kyivstar TV Streaming TV service
*****	1	Orbit - Chewing gum
*****	1	- - News headlines
*****	1	- 30-Day Tai Chi Plan for Beginners Fitness plan poster
*****	1	- - IQ test ad
*****	1	Kia EV4 and EV5 Electric cars

Продовження таблиці А.1

*****	1	СМФ УКРАЇНА Засувка шиберна з рейковим механічним приводом Industrial gate valve
*****	1	- - Car marketplace
*****	1	Sturm! PG87281 Portable generator
*****	1	Lineage 2 Interlude x20 server Online game server
*****	1	zlato.ua - Jewelry
*****	1	- - Cryptocurrency exchange ad
*****	1	Imperia Online - Mobile strategy game
*****	1	Nizhno.com.ua - Women's clothing
*****	1	GoITeens - Educational courses
*****	1	Dark Epoch - PC action RPG game
*****	1	- - Mobile game ad
*****	1	PRACTIK Freshpatačka Pet food
*****	1	- - News headlines
*****	1	Shopify - E-commerce platform
*****	1	Nissan Juke Car
*****	1	- - Weight loss supplement
*****	1	e-datalog - Data/log management service
*****	1	FindanLoc - Phone number lookup service
*****	1	Систейн Ультра Hefound Eye drops
*****	1	- - English proficiency test
*****	1	Wii - Gaming console
*****	1	JustClass - Online learning platform
*****	1	Maserati Grecale Car
*****	1	Hume Health Body Pod Health monitor
*****	1	- - Travel backpacks
*****	1	- - Advertisement banner
*****	1	- - Car import service
*****	1	foundation - Mobile game
*****	1	Hero Wars Dominion Era - Mobile RPG game
*****	1	Tedocs - Health supplement
*****	1	OptiMeal Expert 9100 Pet food
*****	1	MG HS SUV
*****	1	IQOS ILUMA Heated tobacco device
*****	1	- - -
*****	1	hotline - Price comparison service
*****	1	- Power bank 50000 mAh Power bank
*****	1	- - Online course platform
*****	1	COMFY - Electronics retail advertisement
*****	1	Home City - Cleaning services advertisement
*****	1	Sweet.tv - Streaming service
*****	1	- GPT AI model/assistant
*****	1	- BoycottRussians.com Activism website banner
*****	1	Kyivstar - Mobile carrier plan
*****	1	Nissan Qashqai Car

Продовження таблиці А.1

*****	1	- - Women's handbags
*****	1	Parimatch - Sports betting
*****	1	- - Bluetooth earphones
*****	1	Tefal Ingenio Cookware set
*****	1	AHCATOP - Jewelry
*****	1	myIQ - Online test
*****	1	- Інвертор 3000W з зарядкою Power inverter
*****	1	Mercedes-Benz - Car
*****	1	Infineon EZ-PD BCR Electronics development kit
*****	1	Snap-On Smile - Cosmetic dental veneer
*****	1	IQOS ILUMA Electronic cigarette
*****	1	Uklon - Ride-hailing service
*****	1	- - Greenhouse
*****	1	GB - Swimsuit
*****	1	Sweet.tv - Streaming service
*****	1	Cryptobusy Full Time Trader Online trading course
*****	1	lifecell iPhone 17 Smartphone
*****	1	ATESS Power Hybrid Solar System (100kW all-in-one hybrid inverter) Solar inverter system
*****	1	VELIKI - -
*****	1	ARPAL AM-80BD Wood chipper
*****	1	Hero Wars - Mobile game
*****	1	- 50000 mAh power bank Power bank
*****	1	GORTA - Men's sneakers
*****	1	- - Tea
*****	1	Lilith Games Rise of Kingdoms Mobile game
*****	1	- - Garbage bags
*****	1	VENETO - Mattress
*****	1	- - Mobile game
*****	1	- - Sneakers
*****	1	MetroOpinion - Online survey platform
*****	1	Sea of Conquest PC - PC game
*****	1	Citroën C4 Car
*****	1	- - -
*****	1	Park Royal - Real estate/apartments
*****	1	Guns of Glory - Mobile strategy game
*****	1	Worksection - Task management software
*****	1	Shopify - E-commerce platform
*****	1	IFAGRI ТИФО Pesticide
*****	1	GEBERIT AquaClean Bidet toilet
*****	1	- - Cryptocurrency exchange
*****	1	State of Survival - Mobile strategy game
*****	1	Temu - Online marketplace
*****	1	- - Women's tracksuit
*****	1	SOLMAR - T-shirts

Продовження таблиці А.1

*****	1	Kyivstar - Mobile carrier plan
*****	1	Starmag - Home goods
*****	1	Maserati Grecale Car
*****	1	- - Pruning shears
*****	1	- Power Bank 50000 mAh Power bank
*****	1	- - -
*****	1	Superum - Immune support supplement
*****	1	localhistory.org.ua - Book
*****	1	- - Video game
*****	1	Hero Wars - Mobile puzzle ad
*****	1	lps-school.com.ua - Online course
*****	1	ZYN - Nicotine pouches
*****	1	- - Antivirus software ad
*****	1	- 07 Server Gaming server advertisement
*****	1	monday.com - Project management software
*****	1	- - Business course
*****	1	- - -
*****	1	SKVOT Декорування інтер'єрів Online course
*****	1	Temu - Online marketplace
*****	1	Prominada UA-7470 Women's clothing
*****	1	Hero Wars Dominion Era - Mobile RPG game
*****	1	- - Caviar jars
*****	1	NKON - Home batteries
*****	1	- - Garbage bags
*****	1	Maserati Grecale Car
*****	1	- - Pain relief patch ad
*****	1	Endlesswar Impulse - Video game
*****	1	IQO.COM - Online IQ test advertisement
*****	1	Krauff Steli Cookware
*****	1	Honda CR-V SUV
*****	1	Tel IP Asterisk (IP PBX) IP PBX/telephony service
*****	1	SanatDom - Real estate advertisement
*****	1	Sea of Conquest - Mobile strategy game
*****	1	CatVision-Dev - Video game
*****	1	ICR Polska - Certification/auditing services
*****	1	WINE NOT? MARKET - Wine shop/marketplace
*****	1	Sea of Conquest - Video game
*****	1	24 Канал - Weather app/service
*****	1	- - Programming courses for children
*****	1	- - Blanket
*****	1	- - Domain registration ad
*****	1	OLX - Online marketplace advertisement
*****	1	Tefal - Cookware set
*****	1	- - Kitchen appliances
*****	1	YAD.com - Online games

Продовження таблиці А.1

*****	1	JustClass - Online education platform
*****	1	Bet On - Online gambling service
*****	1	State of Survival - Video game
*****	1	- - 3D printed parts
*****	1	Hero Wars - Mobile puzzle ad
*****	1	ClickUp - Productivity software
*****	1	HyperX - Gaming accessories
*****	1	- - Travel booking
*****	1	Makita - Mini chainsaw
*****	1	Data Bio - Corporate training
*****	1	- - -
*****	1	- - Advertisement banner
*****	1	Husqvarna - Chainsaw
*****	1	- - Energy drink
*****	1	- DeLuxe Bedding
*****	1	Citroën C4 Car
*****	1	DIZON - Car vinyl wrap
*****	1	GoDaddy - Web services
*****	1	- - Honey products
*****	1	Mriya - Food gelatin
*****	1	Motorola Moto G57 Smartphone
*****	1	Uklon - Ride-hailing service
*****	1	ANAHATVATI - Herbal supplement
*****	1	USeru - Business productivity software
*****	1	- - Alcohol addiction treatment services
*****	1	- - Mobile game
*****	1	Nissan Qashqai Car
*****	1	Project Entropy - PC strategy game ad
*****	1	myIQ - Online personality/iq test ad
*****	1	Epson EpiqVision Portable smart projector
*****	1	SKYKIPR TV - IPTV service
*****	1	- - Grain dryer
*****	1	Hero Wars - Mobile game advertisement
*****	1	WestGard - Cucumber seeds
*****	1	OLL.TV - Streaming service app
*****	1	Bodor - Laser cutting machine
*****	1	PLABASE - B2B AI solutions service ad
*****	1	Murprotec (assumed) - Home damp proofing service ad
*****	1	Bosch; COMFY - Home appliances promotion advertisement
*****	1	Mirage: Battle of Gods - Mobile RPG game ad
*****	1	Nike - Sneakers
*****	1	ARX - Travel insurance
*****	1	- - Laptop battery
*****	1	GREENPLANET - Heat pump

Кінець таблиці А.1

*****	1	Sakuri Butchers - Kitchen knives
*****	1	Kia - Car
*****	1	Garnier Fructis Hair care product
*****	1	- - -
*****	1	Зема2eu.in - Immigration services
*****	1	allo - Retail promotion
*****	1	Freya India - Online article/content ad
*****	1	Preply - Online tutoring platform ad
*****	1	Allegro Allegro Days Online marketplace promotion ad
*****	1	OKPSY PHARMA Sonterra Melatonin lozenges
*****	1	Philips - Air fryer
*****	1	Fabergé - Luxury goods
*****	1	monobank - Banking services advertisement
*****	1	Euro shopping - Rechargeable LED bulb
*****	1	World of Tanks - Video game
*****	1	Braun - Household appliance
*****	1	Nissan X-Trail e-POWER Car
*****	1	Zakupki.Prom - Public procurement platform
*****	1	- - Car LED headlight bulbs
*****	1	- - Shipping service advertisement
*****	1	- - Camera accessories
*****	1	First Saint Ukraine - Medicines
*****	1	Sweet.tv - Streaming service
*****	1	Optima - Liquid wallpaper
*****	1	LG - Washing machine and dryer
*****	1	- 50000 mAh power bank Power bank
*****	1	Торчин (Torchin) - Sauce
*****	1	Altium - Electronic design software
*****	1	Y-Jesus - Religious article
*****	1	L'Oréal Paris - Cosmetics
*****	1	Favbet - Sports betting promotion
*****	1	alivemoment.com - Photo restoration service
*****	1	Jeep Compass Car
*****	1	World of Warships - Video game
*****	1	Hyundai HYC 2524 Air compressor
*****	1	RZTK - Gaming accessories set
*****	1	Garnier - Corporate sustainability campaign
*****	1	OPIT - Online degree program
*****	1	Small business - Wooden nightstand
*****	1	- - Mini excavator
*****	1	SolaX - Home energy storage system
*****	1	SolaX - Home energy storage system

ДОДАТОК Б СЛАЙДИ ПРЕЗЕНТАЦІЇ ДО ЗАХИСТУ

1

Система автоматизованої
мультимодальної класифікації
реklamних креативів на основі
аналізу зображень та відео

ПАРНЕНКО І.О. КА-22
НАУКОВИЙ КЕРІВНИК: НЕДАШКІВСЬКА Н.І.

2

Актуальність

Навіщо потрібна система?

- ✓ зменшити ручну розмітку креативів аналітиками;
- ✓ перевести банери та відео у таблицю для подальшого аналізу;
- ✓ масштабувати обробку на великі набори посилань;
- ✓ знизити повторні витрати через історію та дедуплікацію.

Чому прості методи недостатні?

- × оптичне розпізнавання символів бачить текст, але не розуміє сцену і логотипи;
- × виявлення об'єктів, не побачить текст з можливим брендом, продуктом або категорією;
- × реклама потребує спільного аналізу тексту, зображення та контексту.

Навіщо аналітикам структуровані дані з креативів

Класифікація не є кінцевою ціллю, вона перетворює банери та відео у таблицю, з якою вже можна працювати як з даними.



Об'єкт, предмет та мета дослідження

Об'єктом дослідження є процес автоматизована обробка рекламних креативів, а саме зображень та відео.

Предметом дослідження є математичні моделі, алгоритми та програмні засоби мультимодальної класифікації рекламних креативів.

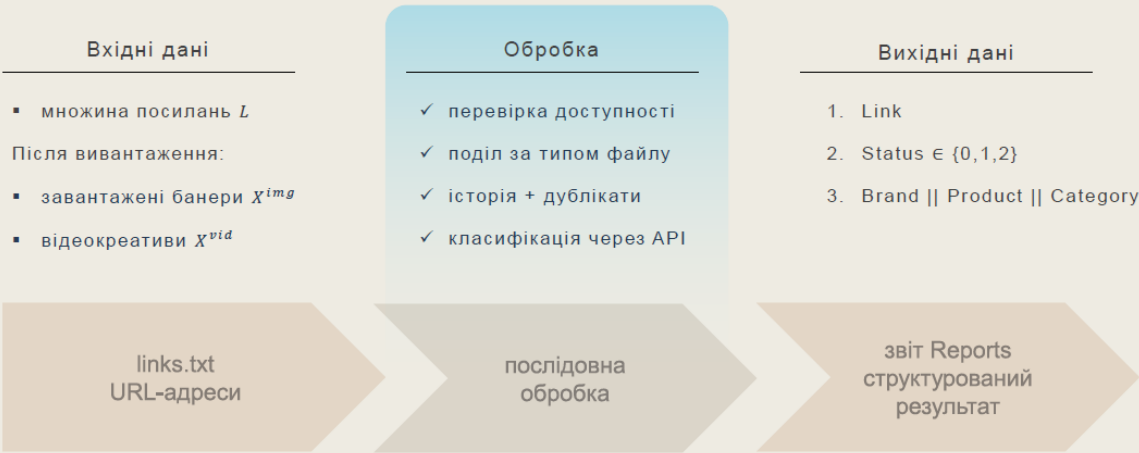
Метою роботи є розробка системи автоматизованої мультимодальної класифікації рекламних креативів для перетворення неструктурованих зображень та відео у структурований звіт із визначеними брендом, продуктом та категорією.

Основні задачі дослідження

- 1. Проаналізувати предметну область автоматизованої обробки рекламних креативів.
- 2. Формалізувати вхідні дані та вихідний результат.
- 3. Побудувати математичну модель класифікації зображень і відео.
- 4. Описати пошук схожих креативів через функцію подібності та граф дублікатів.
- 5. Реалізація модуля вивантаження, класифікації, історії та формування звіту.
- 6. Проведення експериментального оцінювання роботи системи за операційними метриками та ручною вибіркою.

Постановка задачі

Ціль даної роботи є розробка системи для мультимодальної класифікації та перетворення рекламних банерів і відеокреативів у структурований звіт, придатний для подальшої маркетингової аналітики.



Математична модель класифікації

Формалізація вхідних даних

$$L = \{l^1, l^2, \dots, l_n\}$$

$$X = \{x_i \mid x_i = d(l_i), l_i \in L, d(l_i) \neq \emptyset\}$$

$$X = X^{img} \cup X^{vid}, X^{img} \cap X^{vid} = \emptyset$$

Подання відео

$$x_i^{vid} = (f_i^1, f_i^2, \dots, f_{iT_i})$$

$$V_i = \{f_i^1, f_i^2, \dots, f_{ik}\}, k \ll T_i$$

Функція класифікації та вихідні дані

$$g: X \rightarrow Y$$

$$y_i = g(x_i) = (b_i, p_i, c_i)$$

$$r_i = \begin{cases} h(l_i), & \text{якщо } h(l_i) \neq \emptyset \\ g(x_i), & \text{якщо } h(l_i) = \emptyset \end{cases}$$

Ідея моделі в тому, щоб спочатку перевіряти історію, а зовнішню модель викликає тільки для нових або невідомих креативів.

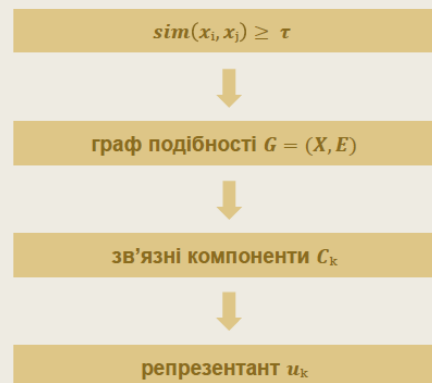
Пошук дублікатів через граф подібності

$$E = \{(x_i, x_j) \mid \text{sim}(x_i, x_j) \geq \tau\}$$

$$G = (X, E), C^1, C^2, \dots, C_m - \text{зв'язні компоненти}$$

$$u_k = \text{rep}(C_k), D = X \setminus U$$

- ❖ вершина графа – окремий креатив
- ❖ ребро – достатня схожість двох креативів
- ❖ компонента зв'язності – група дублікатів
- ❖ класифікується один репрезентант, результат переноситься на групу



результат класифікації переноситься на групу дублікатів

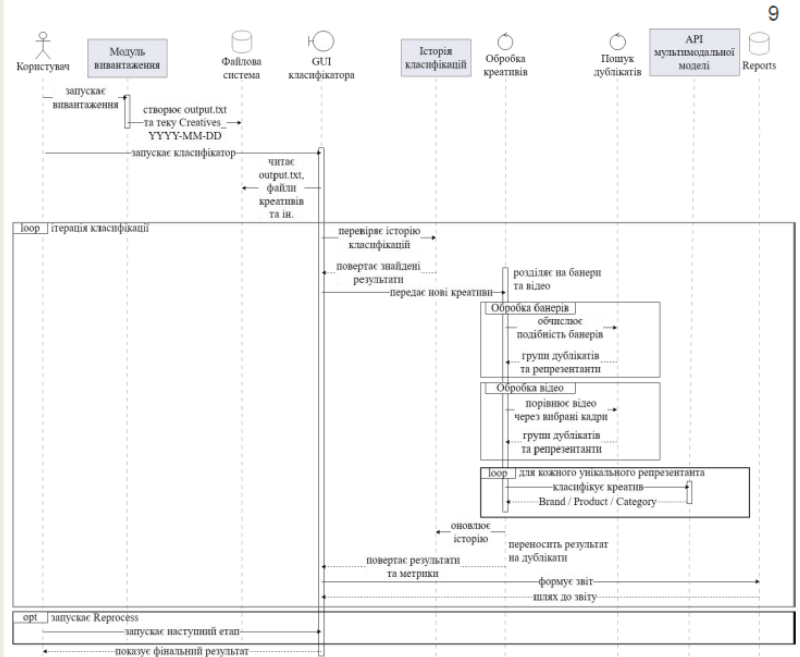
Архітектура програмної системи

Основні модулі:

1. Модуль вивантаження: links.txt, а далі генеруються output.txt + Creatives_YYYY-MM-DD
2. Модуль класифікації: GUI, історія, обробка зображень і відео

Важливо:

- API класифікатор: використовується для нових унікальних креативів
- Reports: фінальний структурований звіт і метрики

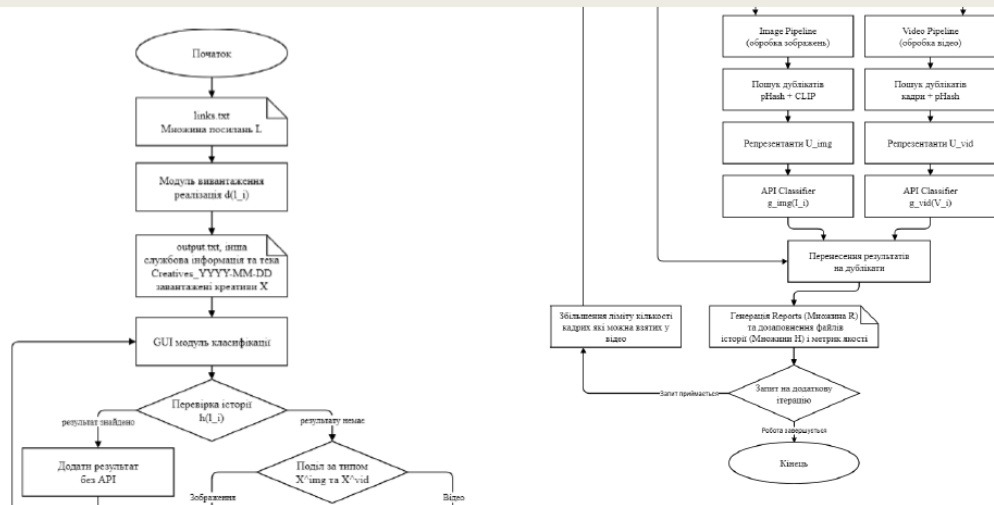


Функціональна схема роботи системи

10

Верхня частина

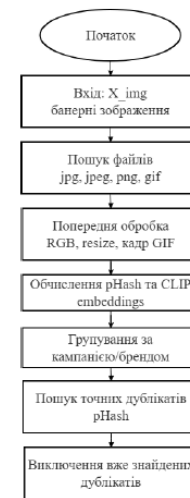
Нижня частина



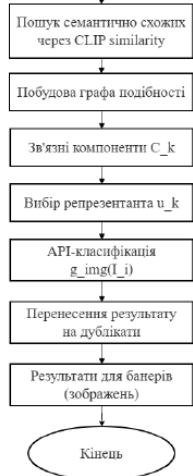
Алгоритм обробки банерних зображень

1. Пошук зображень у теці завантажених креативів.
2. Перевірка історії попередніх класифікацій.
3. Обчислення ознак подібності: rHash та векторні представлення.
4. Групування схожих банерів і вибір унікальних репрезентантів.
5. Класифікація нових унікальних зображень через API.
6. Перенесення результату на знайдені дублікати та запис у звіт.

Верхня частина



Нижня частина



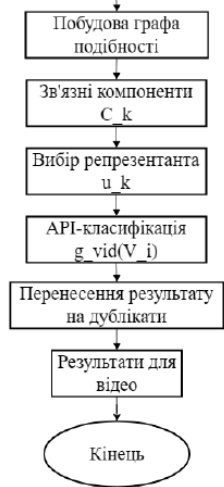
Алгоритм обробки відеокреативів

1. Відео не аналізується повністю покадрово, а формується набір вибраних кадрів.
2. Для кадрів обчислюються візуальні ознаки для порівняння відео між собою.
3. Схожі відео групуються, а для класифікації обирається один репрезентант.
4. До API передається набір кадрів одного відео та текстова інструкція.
5. Результат Brand || Product || Category записується у звіт і може бути використаний повторно.

Верхня частина



Нижня частина

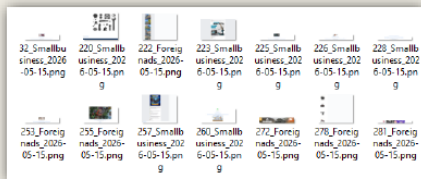


Результати вивантаження та головна сторінка модуля класифікації

Вміст файлу «output.txt»

```
link,status,brand,number,type
https://adreal.gemius.com/ua/show/cid1>,200,Smallbusiness,3,banner
https://adreal.gemius.com/ua/show/cid2>,200,Foreignads,2,banner
https://adreal.gemius.com/ua/show/cid3>,200,Smallbusiness,4,video
https://adreal.gemius.com/ua/show/cid4>,404,,,
https://adreal.gemius.com/ua/show/cid5>,404,,,
https://adreal.gemius.com/ua/show/cid6>,200,Foreignads,11,banner
https://adreal.gemius.com/ua/show/cid7>,200,Smallbusiness,12,banner
https://adreal.gemius.com/ua/show/cid8>,404,,,
https://adreal.gemius.com/ua/show/cid9>,200,Smallbusiness,15,banner
```

Вміст теки з файлами креативів
«Creatives_YYYY-MM-DD»



Головна сторінка інтерфейсу модуля класифікації

Обробка креативів

Коренева директорія:

Патка Creatives_YYYY-MM-DD:

Файл output.txt:

Тип репортю:

День: Місяць:

Рік:

Результати обробки в реальному часі через консоль

Частина виведеної у консоль інформації з обробки та класифікації креативів в кінці роботи

```
INFO: HTTP Request: POST https://api.openai.com/v1/chat/completions "HTTP/1.1 200 OK"
INFO: in=713 ($0.000891), cached=0 ($0.000000), out=124 ($0.001240), total=837 ($0.002131)
INFO: | % | Brand | Product | Category |
|---|---|---|---|
| 1 | Not found | Not found | Bell pepper |
| 2 | Not found | Not found | Bee pollen granules |
| 3 | Not found | Not found | News headlines |
| 4 | Not found | Not found | News headlines |
| 5 | Not found | Not found | Advertisement banner |
| 6 | Not found | Not found | Commemorative coin |
| 7 | Not found | Not found | Car headlight assembly |
INFO: Images summary: campaigns=26, representatives=87, duplicates=1
INFO: Додано 2 елементів із shared_list
INFO: Link auto-fill: було порожніх=91, стало порожніх=0
INFO: Видалено файл: C:\Users\user\PycharmProjects\_Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Classification\_temp\partial_image_results.csv
INFO: Видалено файл: C:\Users\user\PycharmProjects\_Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Classification\_temp\partial_video_results.csv
INFO: Видалено директорію: C:\Users\user\PycharmProjects\_Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Classification\_temp
INFO: Звіт класифікації збережено: C:\Users\user\PycharmProjects\_Study\AI_Creative_Classification\AICreativeClassification\Ads_AI-Classification\Reports\weekly report 15 May 2026 (created at 2026-05-15 02-10).csv
Загальний прогрес | in=9093 | cached=0 | out=1488 | $total=0.026246 | img 88/88 | vid 3/3: 100% | 91/91 [00:32<00:00, 2.8
```

Формулювання текстової інструкції

15

Частина схеми промпту з правилами та завданнями

```
ADMIN_PROMPT_SCHEMA = {
  "Image": {
    "task": (
      "You'll receive an image (with a built-in caption) and must classify it strictly into the table below. "
      "The entire output MUST be written in English only, regardless of the input language."
    ),
    "rules": [
      "Use caption and visual content to extract all fields.",
      "If something is not visible or unknown, write 'Not found'.",
      "Output ONLY the table described below."
    ]
  },
  "video": {
    "task": (
      "You'll receive frames from several videos (each frame labeled by its source video). "
      "The entire output MUST be written in English only, regardless of the input language."
      "For each video, use all its frames to determine a single classification."
    ),
    "rules": [
      "If ANY frame has information, use it to classify the WHOLE video.",
      "Do NOT classify frames individually - one row per video only.",
      "If something is not visible or unknown across frames, write 'Not found'.",
      "Output ONLY the table described below."
    ]
  }
}
```

Приклад швидкого додавання колонок для промпту

```
pb = PromptBuilder(admin_prompt_schema, create_if_missing=True)

if len(pb.columns) == 1:
    pb.add_column(
        "Brand",
        image_rule="company or trademark (e.g. Samsung, L'Oréal), or 'Not found'",
        video_rule="company or trademark (e.g. Samsung, L'Oréal), or 'Not found'",
    )
    pb.add_column(
        "Product",
        image_rule="model name/number (e.g. S24, A15), or 'Not found'",
        video_rule="model name/number (e.g. S24, A15), or 'Not found'",
    )
    pb.add_column(
        "Category",
        image_rule="general type (e.g. kitchen grill, gaming laptop, throat lozenges), or 'Not found'",
        video_rule="general type (e.g. kitchen grill, gaming laptop, throat lozenges), or 'Not found'",
    )
```

Експериментальні результати та метрики

16

Тестовий набір: 400 посилань; 300 креативів завантажено; 100 посилань недоступні.

Склад завантажених даних: 289 банерів і 11 відео.

Показник	1 ітерація	2 ітерація
З історії	0	209
Через API	300	91
Результативно	209	229
Вартість API	0.088 USD	0.026 USD
Час API	256.79 c	78.94 c

Ручна оцінка 100 посилань	Значення
Brand збігся	76
Product збігся	79
Category збіглася	88
Повний збіг	61
Є коректно визначений Brand або Product	64
Індикатор: Accuracy / Precision / Recall / F1	0.90 / 0.86 / 1.00 / 0.93

* індикатор придатності: хоча б Brand або Product є в результаті

Обробка креативів

Завершено: Етап 2 — середній

Витрати (\$USD, всі етапи)

\$0.114485

Ідентифіковано креативів (Downloads ~ 300)

208 (75.00% від усіх)

правильно класифіковані

229 (76.33% від **ідентифі**)

всього в історії

209 (91.27% від **правильно класифікованих**)

фактично обробилися (через API)

91 (36.33% від **ідентифі**)

дублікати

1 (1.10% від **обробл**)

посиланнями

0 (0.00% від **ідентифі**)

Показати деталі проміжку ▼

Покращення

Етап 1 → Етап 2: Δ Cost(\$USD) = +20 (+9.37%)

Історія

Етап 1 — легкий

вартість (\$USD)

\$0.088239

Ідентифіковано

300

правильно класифіковані

200 (66.67% від **ідентифі**)

всього в історії

0 (0.00% від **правильно класифікованих**)

фактично обробилися

300 (100.00% від **ідентифі**)

дублікати

2 (0.67% від **обробл**)

посиланнями

0 (0.00% від **ідентифі**)

Етап 2 — середній (current)

вартість (\$USD)

\$0.026246

Ідентифіковано

300

правильно класифіковані

229 (76.33% від **ідентифі**)

всього в історії

209 (91.27% від **правильно класифікованих**)

Відкрити ланк

Відкрити Етап 3 — легкий (failed%: batch%5)

Відкрити ланк

Аналіз результатів

- Із 400 вхідних посилань система змогла завантажити 300 креативів, тобто частка успішного вивантаження склала **75%**.
- Основну частину даних становили банерні зображення: **289 із 300**, тому система найбільше перевірялася саме на статичних креативах.
- На другій ітерації **209 креативів** було оброблено через історію, тому кількість нових API-запитів зменшилась з **300 до 91**.
- Вартість API-класифікації зменшилась з **0.088 USD до 0.026 USD** завдяки використанню історії на другій ітерації, тобто приблизно на **70%**.
- Час API-класифікації зменшився з **256.79 с до 78.94 с**, тобто приблизно на **69%**.
- Кількість результативно класифікованих креативів зросла з **209 до 229**, що показує користь повторної обробки та використання історії.
- Індикатор придатності – це бінарна ознака: 1, якщо є хоча б Brand або Product; 0, якщо обидва поля порожні або невизначені. Таку оцінку введено, щоб система автоматично відокремлювала корисні для аналітика рядки від порожніх відповідей. Сам індикатор НЕ гарантує коректності полів.
- Найкращий збіг із ручною оцінкою отримано для поля **Category – 88%**. **Precision = 0.86** показує, що більша частина результатів, визначених індикатором як придатні, після ручної перевірки виявилася коректною.

Висновок

У роботі розроблено систему автоматизованої мультимодальної класифікації рекламних креативів, яка перетворює банерні зображення та відео у структурований звіт із полями Brand, Product, Category та статусом обробки.

Формалізовано вхідні та вихідні дані системи, подання зображень і відеокреативів, функцію класифікації та механізм повторного використання попередніх результатів.

Запропоновано підхід до зменшення повторної обробки креативів через історію класифікацій і пошук дублікатів на основі графа подібності.

Реалізовано програмну систему з двох основних модулів: вивантаження рекламних матеріалів та їх подальшої класифікації через графічний інтерфейс.

Експериментальне оцінювання показало, що використання історії дозволяє зменшити кількість API-запитів, час обробки та вартість класифікації, а отримані результати можуть використовуватися як основа для подальшої маркетингової аналітики.

Перспективи подальших досліджень

- Покращення алгоритмів пошуку дублікатів для складніших випадків схожості креативів.
- Удосконалення вибору ключових кадрів відео, щоб відбирати найбільш інформативні сцени.
- Додавання аналізу аудіодоріжки відеокреативів через перетворення мовлення у текст.
- Порівняння різних мультимодальних моделей для класифікації Brand, Product і Category.
- Розширення контрольної вибірки та детальніше оцінювання якості класифікації.

ДЯКУЮ ЗА УВАГУ!

ДОДАТОК В ДОВІДКА ПРО ВПРОВАЖЕННЯ

ДП «ССМ»

м. Київ

«09» червня 2026 р.

ДОВІДКА

про використання розробленої програмної системи

Довідка видана Парненку Іллі Олексійовичу, який перебуває на посаді молодшого техніка-програміста (Junior Python Developer) Департаменту стратегічного планування та розвитку (Data Science Team), про те, що під час виконання дипломної роботи в межах виконання його трудових обов'язків ним було розроблено програмну систему автоматизованої мультимодальної класифікації рекламних креативів на основі аналізу зображень та відео.

Розроблена система використовується у робочих процесах підприємства ДП «ССМ» та належить йому як складова внутрішнього інструменту для первинної обробки даних медіа-моніторингів і підготовки результатів до завантаження у базу даних. Система забезпечує автоматизовану категоризацію рекламних креативів за брендом, продуктом і категорією, що дозволяє зменшити обсяг ручної роботи співробітників, скоротити час обробки даних та підвищити якість моніторингу рекламних активностей. Отримані результати використовуються для подальшого сегментування ринку, аналізу клієнтів і конкурентів, а також підготовки тендерних пропозицій.

Довідку видано для подання до Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Від підприємства ДП «ССМ»

Senior Data Analyst



Фарафонов С. В.

(ПІБ)