МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

МЕЛЕНЬ Віталій Ігорович

Багатоміткова класифікація атрибутів мистецьких об'єктів на основі методів перенесення навчання згорткових нейронних мережах / Multi-label Classification of Artistic Object Attributes Based on Transfer Learning Methods in Convolutional Neural Networks

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-41
В.І. Мелень
_____

Науковий керівник:
д.т.н., доцент Х.В. Ліп'яніна-Гончаренко
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ **Н.В. Дзюбановська**

ТЕРНОПІЛЬ – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
МЕЛЕНЮ Віталію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Багатоміткова класифікація атрибутів мистецьких об'єктів на основі методів перенесення навчання згорткових нейронних мережах / Multi-label Classification of Artistic Object Attributes Based on Transfer Learning Methods in Convolutional Neural Networks

керівник роботи д.т.н., доцент Х.В. Ліп'яніна-Гончаренко
затверджений наказом по університету від 01 грудня 2025 року № 894.
2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Основні питання, які потрібно розробити
- Провести аналіз предметної області атрибуції мистецьких об'єктів, сформувати таксономію атрибутів та виконати огляд існуючих підходів до багатоміткової класифікації мистецьких зображень
- Дослідити структуру та особливості набору даних iMet Collection 2019
- Реалізувати багатоміткову модель класифікації атрибутів мистецьких об'єктів на основі згорткової нейронної мережі Xception з використанням методів перенесення навчання
- Налаштувати процедури порогування та оцінювання якості за F_2-мірою на валідаційній вибірці, проаналізувати розподіл передбачених атрибутів
- Оцінити ефективність розробленого модуля

5. Перелік графічного матеріалу у роботі
- Структурна схема модуля багатоміткової класифікації атрибутів
- Блок-схема алгоритму навчання моделі
- Блок-схема алгоритму інференсу та формування топ-N атрибутів
- Таксономія атрибутів мистецького об'єкта (багатомітковий опис)
- Залежність значення макро-F_2 від порогового значення для бінаризації ймовірностей

- Структура програмного проєкту та інформаційні зв’язки між компонентами

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент _____ В.І. Мелень
підпис

Керівник роботи _____ д.т.н., доцент Х.В. Ліп’яніна-Гончаренко
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Багатоміткова класифікація атрибутів мистецьких об’єктів на основі методів перенесення навчання в згорткових нейронних мережах» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп’ютерні науки» освітньої програми «Комп’ютерні науки» написана обсягом 67 сторінок і містить 15 рисунків, 5 таблиць, 2 додатки та 31 використане джерело.

Актуальність роботи зумовлена стрімким зростанням обсягів цифрових колекцій музеїв та галерей, необхідністю автоматизованої атрибуції мистецьких об’єктів та підвищення якості пошуку в інформаційних системах цифрової культурної спадщини. За наявності сотень атрибутів для одного зображення застосування традиційних ручних підходів стає надмірно трудомістким та суб’єктивним, що обґрунтовує доцільність використання методів глибинного навчання для багатоміткової класифікації.

Об’єктом дослідження є програмний модуль багатоміткової класифікації атрибутів мистецьких об’єктів на основі згорткових нейронних мереж. Предметом дослідження є методи побудови та налаштування моделей глибинного навчання для задач багатоміткової атрибуції мистецьких зображень, зокрема на основі архітектури Xception та датасету iMet Collection 2019.

Метою роботи є розроблення та експериментальна оцінка програмного модуля багатоміткової класифікації атрибутів мистецьких об’єктів на основі згорткової нейронної мережі Xception із використанням підходів перенесення навчання.

Методами дослідження є аналіз предметної області цифрової культурної спадщини та структури набору даних iMet Collection 2019, методи глибинного навчання і перенесення навчання в згорткових нейронних мережах, аугментація зображень, k-fold крос-валідація, оптимізація порога прийняття рішення за F_2-мірою, а також аналіз помилок класифікації.

У роботі спроєктовано конвеєр підготовки даних, навчання та інференсу багатоміткової моделі, реалізовано програмний модуль класифікації на основі попередньо натренованої моделі Xception, проведено дослідження впливу порогів прийняття рішень для багатоміткових прогнозів. На валідаційній вибірці набору iMet Collection 2019 досягнуто значення макро-$F_2 = 0{,}764$, проаналізовано розподіл передбачених атрибутів і типові помилки моделі.

Результати роботи апробовано на X Міжнародній студентській конференції «Наука сьогодення: від досліджень до стратегічних рішень» (м. Одеса, 27 лютого 2026 р.) та можуть бути використані для автоматизації цифрової каталогізації й пошуку в інформаційних системах музейних та мистецьких колекцій.

Ключові слова: БАГАТОМІТКОВА КЛАСИФІКАЦІЯ; МИСТЕЦЬКІ ЗОБРАЖЕННЯ; ЗГОРТКОВА НЕЙРОННА МЕРЕЖА; XCEPTION; ПЕРЕНЕСЕННЯ НАВЧАННЯ; IMET COLLECTION 2019; F_2-МІРА.

ANNOTATION

Qualification work on the topic «Multi-label Classification of Artistic Object Attributes Based on Transfer Learning Methods in Convolutional Neural Networks» for Bachelor's degree in speciality 122 «Computer Science» educational and professional program «Computer Science» is written on 67 pages and it contains 15 figures, 5 tables, 2 annexes and 31 sources.

The work is devoted to the problem of automatic multi-label attribution of artistic objects in large-scale digital collections of museums and galleries. This problem is important for improving the quality and efficiency of metadata creation and search in digital heritage information systems, since manual annotation of numerous attributes for each image is time-consuming and subjective.

The object of research is a software module for multi-label classification of artistic object attributes based on convolutional neural networks. The subject of research is methods for building and tuning deep learning models for multi-label attribution of artistic images, in particular using the Xception architecture and the iMet Collection 2019 dataset.

The purpose of the work is to develop and experimentally evaluate a software module for multi-label classification of artistic object attributes using the Xception convolutional neural network with transfer learning.

The research methods include analysis of the problem domain of digital cultural heritage and the structure of the iMet Collection 2019 dataset, deep learning and transfer learning methods for convolutional neural networks, image augmentation, k-fold cross-validation, decision threshold optimization with respect to the F2-score, as well as error analysis of the classifier.

As a result, a full pipeline for data preparation, model training and inference was designed; a multi-label classifier module based on a pre-trained Xception model was implemented; the influence of decision thresholds for multi-label predictions was investigated. A macro F2-score of 0.764 was achieved on the validation subset of iMet

Collection 2019, and the distribution of predicted attributes and typical model errors was analysed.

The results were approbated at the X International Student Conference «Science of Today: from Research to Strategic Decisions» (Odesa, 27 February 2026) and can be used to support automated digital cataloguing and retrieval in information systems for museum and art collections.

Keywords: MULTI-LABEL CLASSIFICATION; ARTISTIC IMAGES; CONVOLUTIONAL NEURAL NETWORK; XCEPTION; TRANSFER LEARNING; IMET COLLECTION 2019; F2-SCORE.

ЗМІСТ

ВСТУП

Активна цифровізація культурної спадщини призводить до стрімкого зростання обсягів музейних, архівних та галерейних колекцій, представлених у вигляді зображень. Для забезпечення пошуку, фільтрації, аналітики та побудови віртуальних експозицій такі колекції потребують формалізованого опису — атрибуції мистецьких об’єктів за змістовими, стилістичними, культурно-історичними та матеріально-технічними ознаками. На відміну від класичних задач одно-класової класифікації, кожен мистецький об’єкт характеризується множиною атрибутів, що природно формує багатоміткову постановку задачі.

Традиційні підходи до атрибуції, що ґрунтуються на ручному заповненні каталогів фахівцями або на використанні простих ознак і правил, є трудомісткими, погано масштабуються на сотні тисяч зображень і не забезпечують достатньої узгодженості описів. Крім того, мистецькі зображення мають специфічні властивості: довгохвостий розподіл атрибутів (велика кількість рідкісних міток), варіативність ракурсів і якості оцифрування, наявність складних візуальних патернів (орнаментика, фактура, стилізація). Це ускладнює застосування класичних алгоритмів комп’ютерного зору та вимагає спеціалізованих рішень для багатоміткової класифікації.

Сучасні досягнення глибинного навчання, насамперед згорткові нейронні мережі та перенесення навчання (transfer learning), відкрили можливість автоматизованої атрибуції мистецьких об’єктів на основі великих репрезентативних наборів даних. Прикладом є набір iMet Collection 2019, у якому атрибути музейних зображень подано у вигляді багатоміткової розмітки, а якість рішень оцінюється метриками, чутливими до пропуску атрибутів (зокрема F_2-мірою). Водночас значна частина наявних рішень орієнтована на конкурсні постановки й не пропонує відтворюваних програмних модулів, готових до інтеграції в інформаційні системи цифрових колекцій.

Розробка відтворюваного програмного модуля, що реалізує повний конвеєр «дані – модель – порогування – оцінювання» для багатоміткової атрибуції

мистецьких об’єктів на основі згорткової нейронної мережі Xception та набору даних iMet Collection 2019, є актуальним завданням. Такий модуль має забезпечувати коректну роботу з довгохвостим розподілом атрибутів, враховувати особливості музейних зображень і демонструвати достатню якість за метрикою F_2, що важливо для практичних сценаріїв цифрової каталогізації.

Мета роботи — розробити та експериментально оцінити програмний модуль автоматизованої багатоміткової атрибуції мистецьких об’єктів на основі згорткової нейронної мережі Xception.

Для досягнення мети поставлено такі завдання:

1. Провести аналіз предметної області та сучасних підходів до багатоміткової атрибуції мистецьких зображень.
2. Дослідити структуру та особливості набору даних iMet Collection 2019 і сформувати інформаційне забезпечення модуля.
3. Спроєктувати архітектуру програмного модуля, що охоплює конвеєр підготовки даних, навчання моделі та інференсу.
4. Реалізувати багатоміткову модель класифікації на основі згорткової нейронної мережі Xception з використанням перенесення навчання.
5. Налаштувати процедури порогування й оцінювання якості за F_2-мірою на валідаційній вибірці, проаналізувати розподіл передбачених атрибутів та типові помилки моделі.
6. Оцінити ефективність розробленого модуля та сформулювати рекомендації щодо його застосування в задачах цифрової каталогізації мистецьких колекцій.

Предмет дослідження – процеси автоматизованої багатоміткової атрибуції мистецьких об’єктів за їхніми зображеннями з використанням методів глибинного навчання.

Об’єкт дослідження – програмний модуль багатоміткової класифікації атрибутів мистецьких об’єктів на основі згорткової нейронної мережі Xception та набору даних iMet Collection 2019.

Апробація результатів дослідження здійснена в межах X Міжнародної студентської конференції «Наука сьогодення: від досліджень до стратегічних рішень», яка відбулася в місті Одесі 27 лютого 2026 року.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області «атрибуція мистецьких об’єктів»

Мистецькі об’єкти у цифрових колекціях (музейні фонди, архіви, електронні каталоги, віртуальні виставки) потребують формалізованого опису, який забезпечує пошук, фільтрацію, групування та аналітичну обробку. На відміну від ситуацій, де достатньо віднести зображення до одного класу, у сфері культурної спадщини типовим є багатовимірний опис: один і той самий об’єкт одночасно характеризується змістом, культурним контекстом, стилістичними ознаками, матеріально-технічними характеристиками та низкою супровідних атрибутів, що відображають умови його цифрового відтворення. Тому предметна область природно формалізується як задача атрибуції, у якій результатом є множина атрибутів, а не одинична категорія.

Практика цифрової каталогізації показує, що атрибути не є однорідними: вони формують семантичні групи, які описують різні аспекти об’єкта та мають різну “спостережуваність” за зображенням. У межах цієї роботи доцільно застосовувати таксономічне подання атрибутів, яке уніфікує терміни й структурує простір міток, а також підтримує інтерпретацію результатів автоматизованої атрибуції. Таксономію атрибутів мистецького об’єкта наведено на рисунку 1.1, де групи атрибутів розподілено за змістовою, стилістичною, культурно-історичною, матеріально-технічною та цифровою (службовою) ознаками.

Згідно з рисунком 1.1, змістові та іконографічні атрибути відображають те, що саме зображено та в якій формі це подано. До цієї групи належать:

- сюжет/мотив (наприклад, релігійний/міфологічний, історичний/побутовий, алегоричний/символічний);
- тип сцени/композиція (портрет — індивідуальний або груповий; пейзаж/інтер’єр; натюрморт);
- персонажі та об’єкти (людина з уточненням статі/віку/ролі, тварини/рослини, предмети/артефакти, архітектура/ландшафт).

Ці атрибути є ключовими для семантичного пошуку та тематичного групування, однак їхнє автоматичне визначення ускладнюється художньою умовністю, стилізацією та неоднозначністю іконографічних ознак.

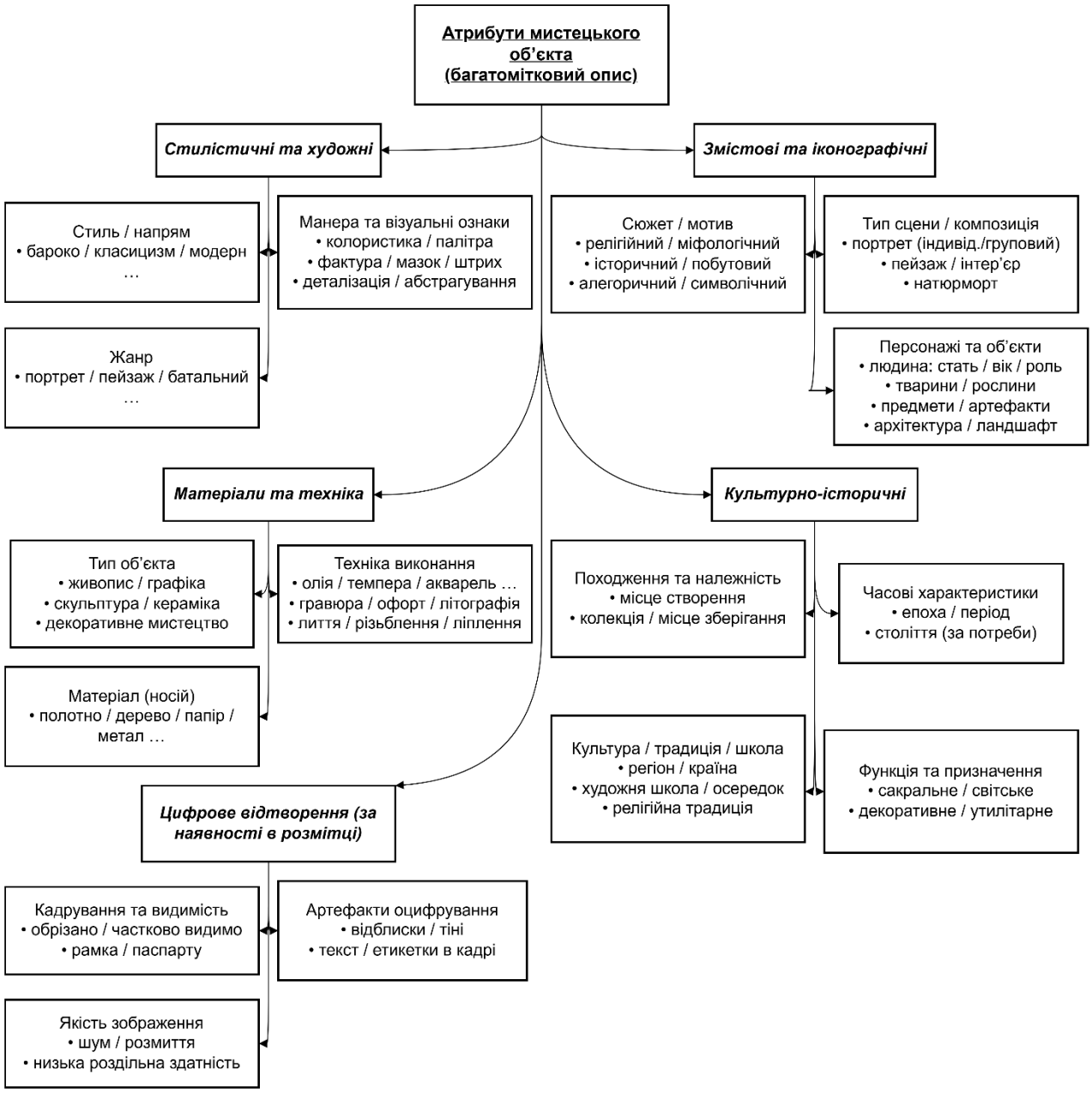


Рисунок 1.1 – Таксономія атрибутів мистецького об’єкта (багатомітковий опис)

Стилістичні та художні атрибути фіксують, як саме виконано твір і які візуальні закономірності є визначальними. На рисунку 1.1 ця група включає стиль/напрям (наприклад, бароко, класицизм, модерн тощо), жанр (портрет,

пейзаж, батальний тощо), а також манеру та візуальні ознаки, зокрема колористику/палітру, фактуру (мазок/штрих) і рівень деталізації/абстрагування. Для цієї групи характерна висока варіативність проявів: однаковий стиль може реалізовуватися через різні техніки, а візуальні ознаки суттєво залежать від якості репродукції (освітлення, баланс білого, контрастність), що підсилює вимоги до нормалізації даних і стійкості алгоритмів.

Культурно-історичні атрибути описують контекст походження та функціональне призначення об’єкта. Як показано на рисунку 1.1, до них належать часові характеристики (епоха/період, за потреби — століття), культура/традиція/школа (регіон/країна, художня школа/осередок, релігійна традиція), походження та належність (місце створення, колекція/місце зберігання), а також функція та призначення (сакральне/світське, декоративне/утилітарне). Значна частина цих атрибутів є слабко спостережуваною лише за зображенням (особливо “місце створення” чи “колекція”), тому в реальних інформаційних системах вони часто поєднуються з метаданими (каталожні записи, інвентарні описи). Водночас у задачі візуальної атрибуції ці атрибути залишаються важливими як цільові мітки, що можуть частково відновлюватися за непрямими ознаками (типова орнаментика, композиційні канони, матеріали, стильові маркери).

Матеріали та техніка є групою атрибутів, що відображає матеріально-технологічну природу твору. Відповідно до рисунка 1.1, вони включають тип об’єкта (живопис/графіка; скульптура/кераміка; декоративне мистецтво), матеріал (носій) (полотно, дерево, папір, метал тощо) і техніку виконання (олія/темпера/акварель; гравюра/офорт/літографія; лиття/різьблення/ліплення). Для автоматизованого розпізнавання ця група є принциповою, оскільки матеріал і техніка впливають на фактуру, характер ліній, зернистість та інші візуальні маркери; водночас помилки оцифрування (шум, розмиття) можуть маскувати ці ознаки, що потребує врахування якості зображень на етапі підготовки даних.

Окремо в таксономії виділено атрибути цифрового відтворення (за наявності у розмітці), які не описують мистецький зміст безпосередньо, але

суттєво впливають на спостережуваність інших атрибутів і можуть бути корисними для контролю якості даних. На рисунку 1.1 до них віднесено кадрування та видимість (обрізано/частково видимо, наявність рамки/паспарту), якість зображення (шум/розмиття, низька роздільна здатність) та артефакти оцифрування (відблиски/тіні, текст/етикетки в кадрі). У практичних наборах даних ці фактори часто проявляються як систематичні джерела похибок і мають бути враховані при аналізі помилок, доборі стратегій попередньої обробки та формуванні навчальних/валідаційних підвибірок.

Таким чином, предметна область атрибуції мистецьких об’єктів характеризується багатовимірністю опису, різнорідністю семантичних груп атрибутів та складністю візуального спостереження окремих характеристик. Додаткові ускладнення пов’язані з варіативністю ракурсів і якості репродукцій, домішками фону, відмінностями технік виконання та нерівномірністю частот атрибутів (довгохвостий розподіл), коли невелика частина міток трапляється часто, а значна кількість — рідко. У типових постановках кількість атрибутів, релевантних одному об’єкту, є змінною, але зазвичай зосереджується у невеликому діапазоні (кілька міток на зразок), що необхідно враховувати при побудові правил формування прогнозованої множини атрибутів та при інтерпретації результатів моделі.

З огляду на структуру атрибутів, подану на рисунку 1.1, атрибуція мистецьких об’єктів природно формалізується як багатоміткова задача з великою кількістю класів, нерівномірним розподілом міток і різною “спостережуваністю” атрибутів за зображенням. Це означає, що вимоги до підходу мають охоплювати не лише вибір моделі комп’ютерного зору, а й обґрунтування функції втрат, методів компенсації дисбалансу, стратегії перетворення ймовірностей у кінцевий набір атрибутів та протоколу оцінювання якості. У зв’язку з цим у наступному підрозділі здійснюється огляд і порівняльний аналіз існуючих аналогів, що застосовуються для багатоміткової атрибуції мистецьких зображень, з виділенням рішень, найбільш придатних для відтворюваної реалізації в межах бакалаврської роботи.

1.2 Огляд і аналіз існуючих аналогів

Існуючі аналоги автоматизованої атрибуції мистецьких об’єктів за зображеннями сформувалися на перетині двох методологічних підходів: багатоміткової класифікації та методів комп’ютерного зору. На відміну від одно-класової класифікації, де кожному зображенню відповідає один клас, у багатомітковій постановці кожен об’єкт описується множиною атрибутів, тому вихід моделі має вигляд вектора $y \in \{0,1\}^K$, де K— кількість атрибутів. Така постановка зумовлює специфічні труднощі:

- нерівномірність частот атрибутів і «довгий хвіст» рідкісних міток;
- домінування негативних міток над позитивними у кожному прикладі;
- наявність взаємозв’язків (кореляцій) між атрибутами;
- неповнота та шум у розмітці.

У підсумку ефективність аналогів визначається не лише архітектурою, а й протоколом підготовки даних, вибором функції втрат, стратегією перетворення ймовірностей на множину атрибутів, а також узгодженням навчання з обраними метриками якості [1- 3].

Ключовим чинником зіставлення аналогів є репрезентативні набори даних і відтворювані протоколи. Значний вплив на розвиток підходів має набір iMet Collection 2019, у якому атрибути музейних зображень формалізовано як багатоміткову задачу з великою кількістю атрибутів та великим обсягом прикладів [4]. Конкурсна постановка iMet підкреслює прикладну мету: автоматичне формування атрибутів для покращення пошуку, групування та навігації у цифрових колекціях [5]. Поряд із цим у літературі використовуються агреговані мистецькі колекції (наприклад, OmniArt), де поєднано зображення та різні типи метаданих; такі дані природно підтримують багатозадачні режими навчання і дозволяють формувати спільні візуальні подання для кількох характеристик творів [6]. Додаткову роль відіграють великі корпуси на кшталт ART500K/DeepArt, орієнтовані на формування подань стилю і змісту та подальше доменне перенесення на спеціалізовані задачі атрибуції [7]. Для

музейних сценаріїв показовими є й практико-орієнтовані виклики (наприклад, Rijksmuseum Challenge), що демонструють високу неоднорідність експонатів і суттєву варіативність умов оцифрування [8]. Огляди з комп'ютерного зору для образотворчого мистецтва узагальнюють, що мистецькі зображення мають відмінну природу ознак порівняно зі «звичайними» фотографіями: на сприйняття впливають техніка виконання, матеріали, реставраційні втручання, умови зйомки та цифрової обробки; це посилює доменний зсув і ускладнює пряме застосування рішень, отриманих на загальних наборах [9-12].

У частині алгоритмічних рішень домінує парадигма перенесення навчання: використання попередньо навчених згорткових нейронних мереж як універсальних екстракторів візуальних ознак з подальшим пристосуванням до цільової задачі [13]. Емпірично показано, що нижні та середні шари таких мереж здебільшого кодують загальні структури (контури, локальні текстури), а верхні шари є більш спеціалізованими; це обґрунтовує практику часткового фіксування параметрів і подальшого тонкого налаштування вибраних шарів [14]. Для багатоміткової атрибуції типово застосовується схема «базова мережа → агрегування ознак → вихідний шар розмірності K із сигмоїдною активацією», де кожен атрибут оцінюється як ймовірність належності; подальшим етапом є перетворення вектора ймовірностей у кінцевий набір атрибутів за визначеним правилом [4, 25].

Найпоширенішими архітектурними аналогами як базові мережі є ResNet (завдяки стабільній оптимізації глибоких моделей) [15], DenseNet (через інтенсивне повторне використання ознак і поліпшену передачу градієнтів) [18], а також EfficientNet, що забезпечує високу якість за помірних обчислювальних витрат завдяки збалансованому масштабуванню глибини, ширини та роздільної здатності [19]. У ролі порівняльних еталонів часто застосовуються VGG та Inception, оскільки вони формують відтворювані базові лінії і дозволяють коректно оцінювати приріст від сучасніших архітектур [16-17]. Для задач мистецької класифікації (стиль, жанр, авторство) показано, що результати суттєво залежать від режимів оптимізації, регуляризації та протоколу

налаштування, навіть за однакової архітектури; отже, «архітектура» не є єдиним визначальним чинником якості [10-12].

Окремим критичним аспектом аналогів у багатомітковій постановці є дисбаланс атрибутів і «перевага негативів». У більшості прикладів кількість відсутніх атрибутів значно перевищує кількість наявних, а рідкісні мітки мають недостатню кількість позитивних прикладів. Це призводить до того, що стандартна бінарна крос-ентропія може спрямовувати навчання на легкі негативні випадки та знижувати здатність відновлювати рідкісні атрибути. Для компенсації цього ефекту в аналогах застосовуються спеціалізовані функції втрат: фокусована втрата, що підсилює внесок складних прикладів [20], класово-збалансована втрата, яка враховує «ефективну кількість» прикладів для класів і обмежує домінування частотних атрибутів [22], та асиметрична втрата для багатоміткової класифікації, що зменшує домінування легких негативів і практично підвищує чутливість до позитивів [21]. Вибір метрики якості також є принциповим: у музейних сценаріях пропуск релевантного атрибуту часто є суттєвішим за появу зайвого, тому застосовуються метрики з підвищеною вагою повноти, зокрема F_β при $\beta > 1$ (наприклад, F_2), що прямо використовується в постановці iMet [4, 5].

Ще однією відмінністю аналогів є стратегія перетворення ймовірностей у множину атрибутів. Використання фіксованого порога (типово 0,5) є простим, але часто неадекватним у разі дисбалансу та різної каліброваності вихідних оцінок. Тому застосовується оптимізація порога на валідаційній вибірці, пороги для окремих атрибутів, а також методи, які навчаються виконувати перехід «оцінки → прогнози» як окремий етап моделювання. До таких підходів віднесено нейромережеві стратегії порогування, що покращують відповідність дискретизації цільовим метрикам [23]. Додатково запропоновано гладкі сурогатні функції втрат, наближені до F-міри (зокрема sigmoidF1), які дозволяють краще узгоджувати навчання з кінцевим критерієм якості [24]. У практиці атрибуції це означає, що приріст якості часто досягається не лише зміною

архітектури, а й ретельним налаштуванням порогування та узгодженням оптимізації з метрикою.

На інженерному рівні аналоги широко використовують штучне розширення даних і об’єднання моделей. Штучне розширення підвищує стійкість до змін масштабу, освітлення та часткових перекриттів, а об’єднання прогнозів кількох моделей або кількох варіантів одного зображення знижує варіативність і підвищує узагальнення, що особливо помітно у конкурсних постановках. Водночас об’єднання моделей збільшує обчислювальну вартість інференсу, тому в прикладних системах потребує балансування між якістю та продуктивністю. Показано, що «сильні базові лінії» на основі стандартних згорткових архітектур у поєднанні з коректними протоколами навчання й об’єднанням прогнозів здатні забезпечувати конкурентні результати і повинні розглядатися як обов’язковий еталон при порівнянні зі складнішими підходами [25].

Перспективні аналоги останніх років включають візуальні трансформери та мультимодальні моделі зображення–текст. Візуальні трансформери демонструють високу якість за умови масштабного попереднього навчання і подальшого перенесення [28], а мультимодальні моделі (наприклад, контрастивні моделі зображення–текст) формують узагальнювані подання, корисні для мало-вибіркового перенесення та семантичної інтерпретації [29]. Проте для відтворюваного бакалаврського дослідження доцільним є фокус на згорткових нейронних мережах із перенесенням навчання як на методологічно зрілих і ресурсно передбачуваних рішеннях, де існують усталені підходи до контролю дисбалансу та налаштування порогування [4, 19-25].

Отже, аналіз аналогів дозволяє виділити практично значущі орієнтири: по-перше, перенесення навчання в згорткових нейронних мережах є базовою парадигмою для мистецького домену; по-друге, якість багатоміткової атрибуції визначається сукупністю рішень — архітектурою, функцією втрат, протоколом навчання та стратегією перетворення ймовірностей у кінцевий набір атрибутів; по-третє, контроль дисбалансу та метрики, чутливі до пропусків, є критичними

для музейних сценаріїв; по-четверте, штучне розширення даних і об’єднання моделей забезпечують приріст якості, але потребують узгодження з обчислювальними обмеженнями. Порівняльну характеристику ключових аналогів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння підходів до багатоміткової атрибуції мистецьких об’єктів (аналоги)

Аналог / джерело	Набір даних / домен	Базовий підхід	Критерій оцінювання	Переваги	Обмеження / ризики
iMet Collection 2019 [4, 5]	Музейні зображення; велика кількість атрибутів	Згорткова нейронна мережа з перенесенням навчання; сигмоїдний вихід на K атрибутів	F_β, зокрема F_2	Відтворювана постановка; практична релевантність для колекцій	Дисбаланс атрибутів; потреба у налаштуванні порогування
OmniArt [6]	Агреговані колекції з метаданими	Багатозадачне навчання зі спільним поданням	Метрики за підзадачами (F-міра/точність)	Використання взаємодоповнювальних сигналів	Неоднорідність даних; складніший протокол оцінювання
ART500K / DeepArt [7]	Великі мистецькі корпуси (стиль/зміст)	Формування подань і перенесення у цільові задачі	Класифікація/ пошук подібності	Кращі подання стилю; придатність для попереднього навчання	Вимогливість до ресурсів; доменна невідповідність атрибутам
Rijksmuseum Challenge [8]	Музейні експонати; різні типи об’єктів	Моделі комп’ютерного зору для музейних сценаріїв	Залежить від задачі	Реалістичні умови; музейна специфіка	Висока варіативність; складність стандартизації протоколу

Продовження таблиці 1.1.

Аналог/ джерело	Набір даних/ домен	Базовий підхід	Критерій оцінювання	Переваги	Обмеження/ ризики
Архітектура ResNet [15]	Загальний домен → перенесення на мистецтво	Глибокі залишкові мережі; тонке налаштування	F-міра/ mAP тощо	Стабільна базова лінія; висока відтворюваність	Чутливість до режиму налаштування і регуляризації
Архітектура EfficientNet [19]	Загальний домен → перенесення на мистецтво	Збалансоване масштабування; перенесення навчання	F-міра/ mAP	Висока якість за помірних ресурсів	Необхідність ретельного підбору роздільної здатності/регуляризації
Фокусована втрата [20]	Дисбаланс у задачах класифікації	Модифікація функції втрат для підсилення складних прикладів	Опосередковано через F-міру/mAP	Підвищує чутливість до складних прикладів	Потреба налаштування параметрів
Асиметрична втрата [21]	Багатомітková класифікація з домінуванням негативів	Різне зважування позитивної/негативної частини втрати	mAP / F-міра	Зменшує домінування легких негативів	Чутливість до параметрів та особливостей даних
Нейромережеве порогування [23]	Загальні багатомітkові задачі	Навчання переходу «оцінки → мітки»	F-міра	Покращує дискретизацію виходів	Ризик переобучення на валідації; додатковий компонент
Гладка сурогатна втрата до F-міри [24]	Загальні багатомітkові задачі	Оптимізація, узгоджена з F-орієнтованим критерієм	F-міра	Краще узгодження навчання з метрикою	Потреба обережної інтерпретації та налаштування

Узагальнюючи, аналіз існуючих аналогів засвідчує, що найбільш відтворюваними та практично результативними підходами до багатоміткової атрибуції мистецьких об’єктів є рішення на основі перенесення навчання у згорткових нейронних мережах, де підсумкова якість визначається не лише вибором архітектури, а й коректною організацією навчання: урахуванням дисбалансу атрибутів через спеціалізовані функції втрат, узгодженням оцінювання з метриками, чутливими до пропусків, та обґрунтованою стратегією перетворення ймовірностей у кінцевий набір міток; отже, подальше дослідження доцільно спрямувати на побудову відтворюваного конвеєра «дані–модель–порогування–оцінювання» з контролем дисбалансу і перевіркою впливу ключових компонентів на результати.

1.3 Постановка задачі дослідження та вимоги

Актуальність обраної тематики зумовлена стрімкою цифровізацією культурної спадщини та зростанням обсягів музейних, архівних і галерейних колекцій, представлених у вигляді зображень. Як показано в підрозділі 1.1, мистецькі об’єкти в таких колекціях потребують не просто «прикріплення» до одного класу, а багатовимірної атрибуції, яка одночасно відображає зміст, стиль, культурно-історичний контекст, матеріально-технічні характеристики та умови цифрового відтворення. У ручному режимі підтримувати настільки детальний опис для сотень тисяч зображень практично неможливо: процес є трудомістким, дорогим і схильним до суб’єктивних розбіжностей між фахівцями. Це формує запит на автоматизовані інтелектуальні рішення, здатні масштабовано підтримувати атрибуцію мистецьких об’єктів у цифрових колекціях.

З огляду на структуровану таксономію атрибутів, наведену в підрозділі 1.1, задача атрибуції природно формалізується як багатоміткова класифікація з великою кількістю класів та довгохвостим розподілом міток. Для такої постановки характерні специфічні труднощі: домінування негативних ознак у векторах міток, рідкісні атрибути з малою кількістю позитивних прикладів,

часткова спостережуваність деяких характеристик лише за зображенням, а також варіативність якості й ракурсів репродукцій. За відсутності спеціалізованого алгоритмічного та програмного забезпечення ці фактори призводять до низької повноти відновлення атрибутів, особливо для культурно-історичних та матеріально-технічних ознак, що суттєво знижує цінність цифрових колекцій як джерела для пошуку, аналітики та дослідницьких задач.

Огляд існуючих аналогів у підрозділі 1.2 показує, що найбільш результативні підходи до автоматизованої атрибуції мистецьких об’єктів базуються на перенесенні навчання в згорткових нейронних мережах та використанні великих репрезентативних наборів, зокрема iMet Collection 2019. Водночас на практиці якість таких рішень визначається не лише обраною архітектурою, а й повнотою конвеєра «дані – підготовка – навчання – порогування – оцінювання». Значна частина робіт зосереджується на порівнянні мереж (ResNet, EfficientNet, Xception тощо), тоді як питання дисбалансу атрибутів, узгодження навчання з F-орієнтованими метриками та оптимізації правила перетворення ймовірностей на множину міток залишаються недостатньо формалізованими та важко відтворюваними.

Особливої актуальності набуває побудова відтворюваного програмного модуля, який би поєднував сучасні підходи багатоміткової класифікації з урахуванням специфіки мистецького домену. Як випливає з підрозділу 1.2, у наявних дослідженнях часто відсутній детальний опис конвеєра підготовки даних, не наводяться параметри аугментацій, не фіксуються критерії вибору порогів та не здійснюється системний аналіз характеру помилок на рівні конкретних об’єктів. У результаті повторне використання або адаптація таких рішень для інших колекцій (зокрема українських музейних фондів) ускладнюється, а порівняння якості різних підходів стає некоректним.

У цьому контексті актуальним є створення цілісного програмного модуля багатоміткової атрибуції, який, з одного боку, спирається на перевірені архітектури глибинного навчання та репрезентативний датасет iMet, а з іншого – детально описує інформаційне, алгоритмічне та програмно-технологічне

забезпечення. Такий модуль має забезпечувати прозоре керування дисбалансом атрибутів, оптимізацію F_2-міри як ключової метрики для музейних сценаріїв, емпіричний підбір порога бінаризації та можливість аналізу як кількісних, так і якісних результатів (на рівні окремих зображень). Отриманий інструментарій є важливим кроком у напрямі масштабованої цифрової каталогізації, що підтримує як внутрішні потреби установ культурної спадщини, так і наукові дослідження в галузі комп'ютерного зору для мистецтва.

Мета роботи — розробити та експериментально оцінити програмний модуль автоматизованої багатоміткової атрибуції мистецьких об'єктів на основі згорткової нейронної мережі Xception.

Для досягнення мети поставлено такі завдання:

1. Провести аналіз предметної області та сучасних підходів до багатоміткової атрибуції мистецьких зображень.
2. Дослідити структуру та особливості набору даних iMet Collection 2019 і сформувати інформаційне забезпечення модуля.
3. Спроєктувати архітектуру програмного модуля, що охоплює конвеєр підготовки даних, навчання моделі та інференсу.
4. Реалізувати багатоміткову модель класифікації на основі згорткової нейронної мережі Xception з використанням перенесення навчання.
5. Налаштувати процедури порогування й оцінювання якості за F_2-мірою на валідаційній вибірці, проаналізувати розподіл передбачених атрибутів та типові помилки моделі.
6. Оцінити ефективність розробленого модуля та сформулювати рекомендації щодо його застосування в задачах цифрової каталогізації мистецьких колекцій.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

2.1 Загальна структура проєктованого модуля

Проєктований модуль багатоміткової класифікації атрибутів мистецьких об’єктів доцільно розглядати як цілісний програмний конвеєр, який перетворює вхідні дані (зображення та їхні анотації) у вихідний результат у вигляді множини атрибутів із кількісними оцінками впевненості. Такий конвеєр має забезпечувати відтворюваність експериментів, повторне використання навченої моделі та прозорі потоки даних між етапами. На рівні архітектури модуль подано як послідовність взаємопов’язаних підсистем: завантаження даних, попередня обробка, формування навчальної та валідаційної вибірок, побудова моделі, навчання, оцінювання, збереження артефактів та інференс. Структурну схему модуля наведено на рисунку 2.1.

Рисунок 2.1 – Структурна схема модуля багатоміткової класифікації атрибутів

Як показано на рисунку 2.1, вхідними джерелами для модуля є зображення, файл міток у форматі CSV та словник атрибутів. Словник атрибутів забезпечує уніфіковане відображення між ідентифікаторами міток і їхніми текстовими назвами та використовується як під час формування навчальних векторів, так і на етапі інтерпретації результатів. Блок «Завантажувач даних» виконує зчитування шляхів до зображень і відповідних анотацій із CSV, контролює цілісність входу та формує узгоджені структури даних для подальшого опрацювання. На цьому етапі доцільно виконувати первинну перевірку: наявність файлів, коректність форматів, узгодженість кількості записів, а також коректність атрибутів відносно словника.

Блок «Попередня обробка» забезпечує приведення зображень до стандартного вигляду, необхідного для подачі на вхід нейромережевої моделі. Типово сюди належать зміна розміру, нормалізація піксельних значень і, за потреби, операції штучного розширення навчальних даних, що підвищують стійкість моделі до варіативності ракурсів, освітлення та якості репродукцій. Паралельно з обробкою зображень виконується перетворення текстових/ідентифікаційних анотацій у багатоміткове подання — вектор розмірності K, де кожна позиція відповідає наявності або відсутності певного атрибута. Результатом цього етапу є підготовлені тензори зображень і відповідні багатоміткові вектори міток.

Далі, відповідно до рисунка 2.1, виконується формування підвибірок «Навчання і валідація даних», що є необхідним для коректної оцінки узагальнювальної здатності моделі та уникнення завищення показників якості. Навчальна підвибірка використовується для оптимізації параметрів, тоді як валідаційна — для контролю перенавчання, підбору гіперпараметрів і налаштування правил формування кінцевого набору міток (наприклад, порогів дискретизації).

Блок «Побудова моделі» реалізує створення архітектури багатоміткової класифікації, орієнтованої на перенесення навчання. Практично це означає використання попередньо навченої згорткової нейронної мережі як екстрактора

ознак та додавання вихідного шару розмірності K, який формує оцінки для кожного атрибута. Вибір конфігурації моделі (вхідний розмір, частина мережі, що донавчається, регуляризація, параметри оптимізації) визначає баланс між якістю, стійкістю та обчислювальними витратами.

Блок «Навчання моделі» виконує оптимізацію параметрів на навчальній вибірці та формує основні артефакти відтворюваності: ваги моделі, конфігурацію гіперпараметрів і журнал навчання (динаміка функції втрат і метрик). У межах цього етапу доцільно застосовувати механізми керування процесом навчання (контроль найкращої моделі, раннє завершення, адаптацію швидкості навчання), оскільки багатоміткова постановка та дисбаланс атрибутів підвищують ризик перенавчання і нестабільності.

Блок «Оцінювання» призначений для розрахунку метрик на валідаційній вибірці та формування підсумкового блоку «Результати та метрики» (див. рисунок 2.1). Саме на цьому етапі кількісно встановлюється якість моделі за критеріями багатоміткової класифікації та, за потреби, виконується налаштування правила перетворення вектора ймовірностей у кінцевий набір міток (наприклад, підбір порога або вибір топ-N атрибутів). Отримані значення метрик і параметри дискретизації мають бути зафіксовані як частина експериментального протоколу, оскільки вони безпосередньо визначають прикладну інтерпретацію результатів атрибуції.

Блок «Збереження моделі» забезпечує серіалізацію навченої моделі та пов’язаних артефактів (ваги, конфігурація, словник атрибутів, параметри попередньої обробки та порогування), що дозволяє виконувати повторний інференс без повторного навчання та підтримує відтворюваність експериментів. Завершальним етапом є «Інференс», під час якого модуль застосовується до нових зображень: після стандартної попередньої обробки обчислюється вектор ймовірностей для Kатрибутів, виконується дискретизація за визначеним правилом і формується вихідний блок «Результати атрибуції» (див. рисунок 2.1), який подається як перелік атрибутів із відповідними оцінками впевненості.

Отже, структурна схема на рисунку 2.1 відображає логічно завершений цикл “дані → модель → оцінювання → застосування”, у межах якого кожен блок має чітко визначені вхідні та вихідні дані: від зображень і CSV-міток до тензорів, багатоміткових векторів, параметрів моделі, метрик та підсумкових результатів атрибуції. Така організація забезпечує модульність реалізації, контроль якості на валідації та можливість практичного використання навченої моделі в задачах цифрової каталогізації мистецьких колекцій.

2.2 Алгоритмічне забезпечення

Алгоритмічне забезпечення проєктованого модуля ґрунтується на багатомітковій постановці задачі, у якій кожному зображенню мистецького об’єкта x_iвідповідає вектор міток $y_i \in \{0,1\}^K$, де K— кількість атрибутів таксономії (див. рисунок 1.1). Метою є побудова відображення $f(x_i;\theta)$, яке формує вектор оцінок належності $p_i \in [0,1]^K$для всіх атрибутів. Для цього застосовується нейромережева архітектура на основі перенесення навчання: попередньо навчена згорткова нейронна мережа використовується як екстрактор ознак, а вихідний шар розмірності Kіз сигмоїдною активацією забезпечує незалежне (за виходом) оцінювання ймовірностей атрибутів. Надалі ймовірнісні оцінки перетворюються у кінцеву множину міток за визначеним правилом (порогування або вибір топ-N), що відповідає практичним вимогам атрибуції цифрових колекцій.

Підготовка даних у межах алгоритму включає: зчитування зображень і таблиці міток, формування відповідності «ідентифікатор атрибута ↔ назва» за словником, кодування міток у форматі multi-hot та нормалізацію зображень до уніфікованого тензорного представлення. Особливістю предметної області є довгохвостий розподіл частот атрибутів і домінування негативних міток над позитивними в кожному прикладі; це зумовлює необхідність контролю дисбалансу та коректного налаштування процедури дискретизації виходів. На етапі формування вибірок дані поділяються на навчальну й валідаційну частини,

причому валідаційний набір використовується не лише для оцінювання узагальнення, а й для добору порогів або інших параметрів перетворення вектора ймовірностей у множину атрибутів.

Навчання моделі реалізується як ітеративний процес оптимізації параметрів θна навчальній вибірці за допомогою оптимізатора градієнтного типу. Для кожного пакета даних виконується прямий прохід, обчислюється функція втрат і виконується зворотне поширення похибки. У базовому варіанті для багатоміткової класифікації використовується бінарна крос-ентропія, яка агрегує внесок усіх Kатрибутів. З огляду на дисбаланс атрибутів, у рамках алгоритму доцільно передбачити можливість модифікації функції втрат (наприклад, вагове зважування для рідкісних атрибутів або фокусування на складних прикладах), що підвищує повноту відновлення малочастотних міток. Контроль якості здійснюється на валідаційному наборі після кожної епохи: обчислюються метрики, аналізується динаміка втрат і, за умови покращення, зберігаються найкращі параметри моделі. Такий підхід забезпечує відтворюваний протокол вибору моделі та зменшує ризик перенавчання.

Загальну логіку алгоритму навчання та оцінювання в модулі відображено на рисунку 2.2. На схемі показано послідовність: завантаження і підготовка даних, формування навчальної/валідаційної підвибірок, побудова моделі, налаштування функції втрат і оптимізатора, ітеративне навчання на пакетах із проміжним оцінюванням на валідації, збереження найкращих ваг, а також завершальний етап налаштування порогів і збереження моделі разом із параметрами.

Після завершення навчання важливою складовою алгоритмічного забезпечення є налаштування правила формування міток на основі ймовірнісних виходів моделі. Оскільки сигмоїдний вихід формує p_iяк неперервні оцінки, для отримання кінцевої множини атрибутів необхідна дискретизація. У модулі передбачено два базові режими: (1) порогування, коли атрибут jвважається наявним, якщо $p_{ij} \geq t$(глобальний поріг) або $p_{ij} \geq t_j$(поріг за атрибутом); (2) вибір топ-N, коли у відповідь повертаються Nатрибутів із найбільшими

значеннями ймовірності. Порогування забезпечує гнучкість і дозволяє балансувати точність і повноту, тоді як топ-N формує стабільну за розміром відповідь, що може бути корисним у прикладних сценаріях каталогізації. Параметри порогування доцільно визначати на валідаційному наборі за обраною метрикою багатоміткової якості, оскільки саме правило дискретизації істотно впливає на частку пропущених і помилково доданих атрибутів.

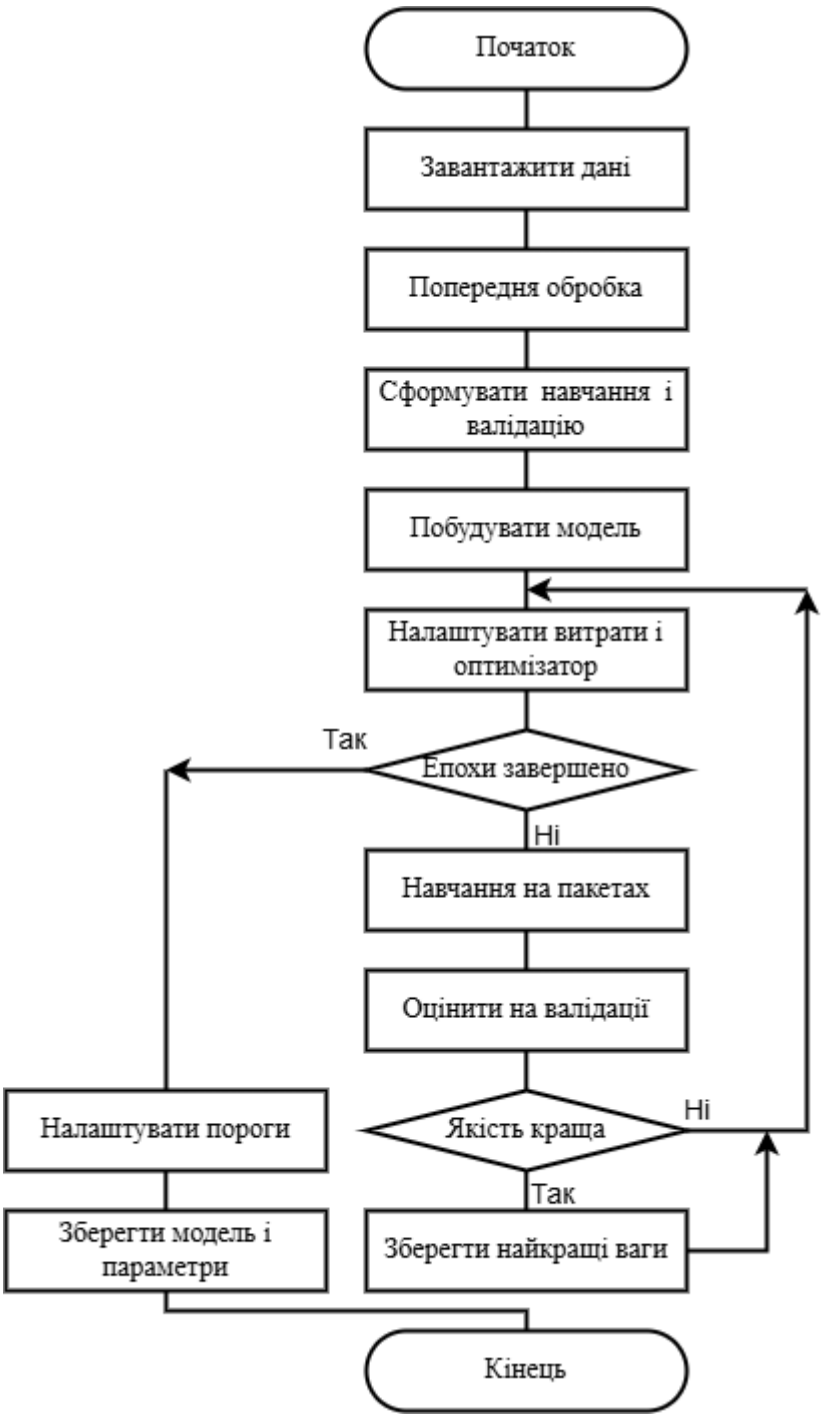


Рисунок 2.2 – Блок-схема алгоритму навчання моделі

Алгоритм інференсу, який реалізує застосування збереженої моделі до нових зображень і формування вихідного результату атрибуції, наведено на рисунку 2.3. Схема відображає завантаження моделі та словника атрибутів, зчитування і попередню обробку зображення, обчислення вектора ймовірностей, застосування правила формування міток (поріг або топ-N), формування списку атрибутів та додавання оцінок впевненості. Завершальним кроком є повернення результату атрибуції в структурованому вигляді, придатному для подальшого збереження або інтеграції в інформаційну систему цифрової колекції.

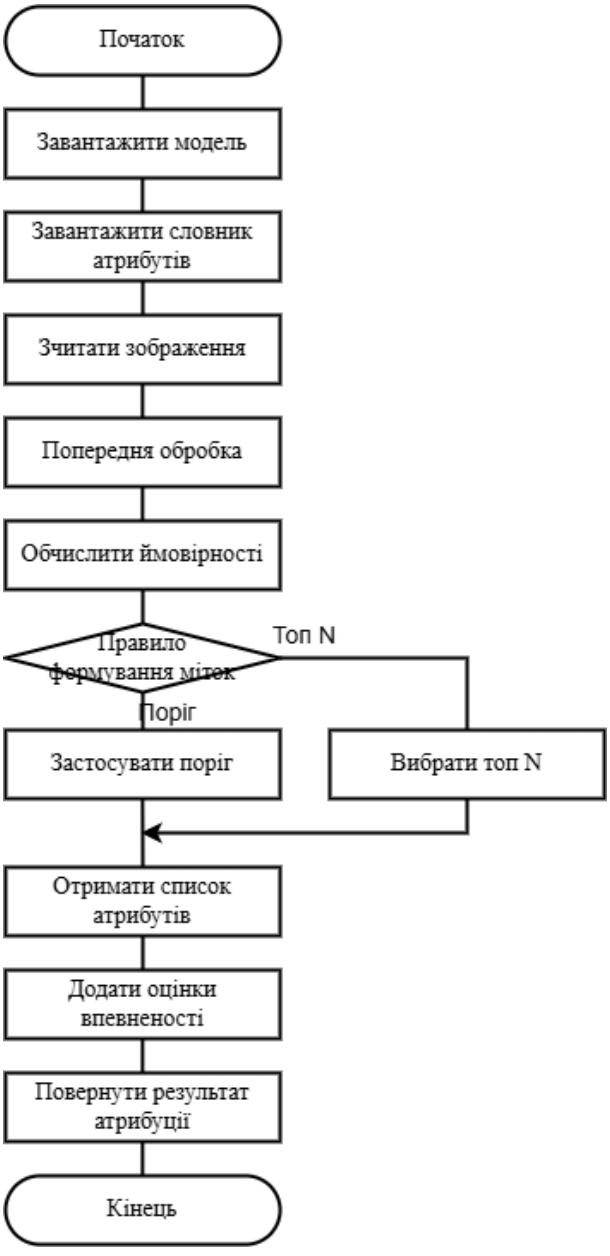


Рисунок 2.3 – Блок-схема алгоритму інференсу та формування топ-N атрибутів

Таким чином, алгоритмічне забезпечення модуля охоплює два взаємодоповнювальні контури:

- контур навчання та оцінювання, який забезпечує отримання оптимальних параметрів моделі, контроль узагальнення і фіксацію артефактів відтворюваності (див. рисунок 2.2);

- контур інференсу, який забезпечує прикладне застосування моделі й інтерпретоване представлення результатів у вигляді множини атрибутів з оцінками впевненості (див. рисунок 2.3).

Узгодження цих контурів через налаштування порогів/правил формування міток і збереження параметрів дозволяє забезпечити стабільну якість атрибуції та повторюваність результатів у межах цифрових колекцій.

2.3 Інформаційне забезпечення

Інформаційне забезпечення програмного модуля базується на наборі даних iMet Collection 2019, у якому кожен об’єкт подано парою *«зображення – множина атрибутів»*. Атрибути задано у вигляді списку ідентифікаторів (attribute_ids), що в процесі підготовки даних перетворюється на багатоміткове подання *multi-hot* (вектор довжини N, де N— кількість атрибутів). У межах датасету визначено 1103 атрибути, що формалізує задачу як багатоміткову класифікацію з високою розмірністю вихідного простору та потенційною розрідженістю міток. Загальний обсяг вибірки становить 155 531 зображення, розподілених на підвибірки: 109 274 для навчання, 7 443 для валідації та 38 814 для тестування (таблиця 2.1). Даний розподіл забезпечує достатній обсяг навчальних прикладів для формування узагальнювальної здатності моделі та водночас дозволяє здійснювати контроль перенавчання на незалежній валідаційній множині. Первинний опис та характеристика набору iMet Collection 2019 наведені у джерелі [30].

Важливою складовою інформаційного забезпечення є аналіз метаданих зображень (розмірів, співвідношення сторін, інтенсивнісних статистик), оскільки

ці характеристики визначають вибір стратегії попередньої обробки та параметрів підготовки пакетів для навчання. За результатами аналізу встановлено, що для всіх зображень у навчальній частині мінімальний розмір однієї зі сторін є стандартизованим і дорівнює 300 пікселів: для випадків $width \geq height$виконується $height = 300$, а для $width \leq height$— $width = 300$. Аналогічна властивість підтверджується і для тестового набору. Така структурна особливість даних є практично значущою, оскільки дозволяє реалізувати уніфіковане правило приведення зображень до цільового формату з мінімальними втратами просторової інформації та прогнозованим споживанням пам'яті під час навчання.

Таблиця 2.1 – Опис набору даних iMet Collection 2019

Показник	Значення
Загальна кількість зображень	155 531
Train (навчальна підвибірка)	109 274
Val (валідаційна підвибірка)	7 443
Test (тестова підвибірка)	38 814
Кількість атрибутів (класів)	1 103
Тип розмітки	багатоміткова (multi-label), multi-hot кодування

Окремо розглянуто випадки екстремальних співвідношень сторін, які можуть ускладнювати коректне масштабування. На рисунок 2.4 наведено приклади витягнутих зображень із $aspect_ratio > 3$або $aspect\_ratio < \frac{1}{3}$, для яких пряме приведення до квадратного розміру через масштабування призводить до суттєвих геометричних спотворень та втрати семантично важливих локальних патернів (орнаменти, написи, фрагменти декору). З огляду на це застосовано диференційований підхід до обробки: для зображень із крайніми значеннями співвідношення сторін спочатку виконується центроване обрізання до 300×300, після чого — масштабування до цільового розміру (наприклад, 156×156); для решти зображень здійснюється лише масштабування до цільового розміру без обрізання. Така стратегія знижує ризик «розтягування» деталей та забезпечує більш стабільні умови для навчання згорткових шарів.

Візуальна різноманітність об’єктів та фонових умов додатково ілюструється прикладами зображень із навчальної вибірки (рисунок 2.5), де спостерігаються відмінності в освітленні, матеріалах, текстурах, а також у композиції та орієнтації об’єктів. Наявність поворотів, часткових обрізань і варіативності масштабу підсилює вимоги до нормалізації та стабільності конвеєра завантаження даних, а також обґрунтовує використання потокового читання зображень (streaming) і пакетної обробки. З позиції ресурсної ефективності доцільним є завантаження зображень «на льоту» з подальшою стандартизацією формату та нормалізацією інтенсивностей, що зменшує витрати оперативної пам’яті та прискорює навчання при роботі з великими наборами даних.

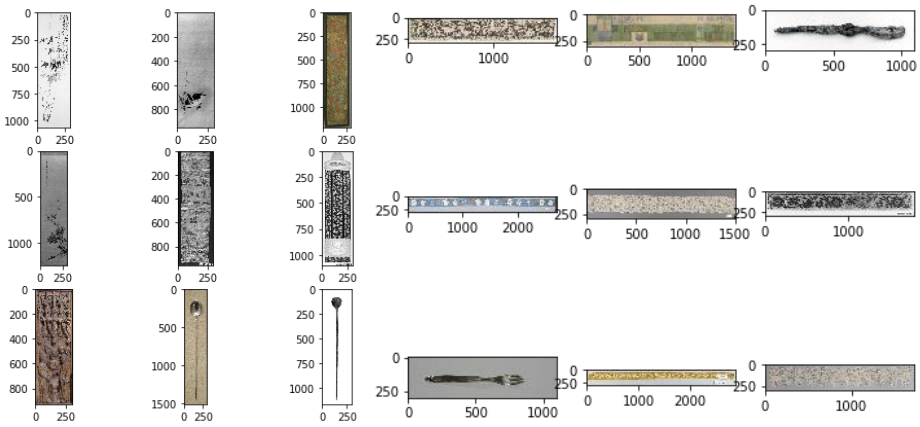


Рисунок 2.4 – Приклади зображень із екстремальним співвідношенням сторін (витягнуті/вузькі зображення)

Рисунок 2.5 – Приклади зображень із навчальної вибірки, що демонструють варіативність об’єктів і умов зйомки

Суттєвим фактором, який визначає вимоги до інформаційного забезпечення та критерії оцінювання моделі, є розподіл міток. Для багатоміткових даних iMet характерний довгохвостий розподіл: мала кількість атрибутів зустрічається часто, тоді як значна частина атрибутів є рідкісними. Таким чином, інформаційне забезпечення у даній роботі включає не лише формальний опис набору даних, а й аналітичне обґрунтування структури даних та процедур їх підготовки, що безпосередньо впливає на якість і відтворюваність подальших експериментів.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ТА ТЕСТУВАННЯ

3.1 Реалізація програмного забезпечення

Програмну реалізацію модуля багатоміткової класифікації здійснено в середовищі Python (Додаток А) із використанням бібліотек глибинного навчання TensorFlow/Keras, а також інструментів оброблення даних NumPy та Pandas. Обчислювальна складова орієнтована на виконання на графічному процесорі (GPU), що забезпечує суттєве скорочення часу навчання згорткової нейронної мережі на великомасштабному наборі зображень iMet Collection 2019 [30]. Реалізацію організовано у вигляді відтворюваного програмного конвеєра, який охоплює повний цикл роботи з даними: зчитування метаданих, кодування міток у формат *multi-hot* розмірності $N\_CLASSES = 1103$, формування потокового датасету tf.data з паралельною обробкою, ініціалізацію моделі, навчання з журналюванням метрик, збереження найкращих ваг і виконання інференсу.

Загальну архітектуру програмного рішення та інформаційні зв’язки між його компонентами подано на рисунок 3.1. Схема відображає логічну декомпозицію системи на функціональні підсистеми: блок даних і метаданих, модуль підготовки даних (data/), модуль моделі (model/), підсистему навчання (train/), модуль валідації та підбору порога (reports/), блок тестування та підсистему інференсу (infer/). Така модульна організація забезпечує чітке розмежування відповідальностей між компонентами, підвищує читабельність коду та спрощує експериментування з різними конфігураціями моделі.

У верхній частині схеми представлено джерела даних: зображення навчальної, валідаційної та тестової вибірок, файл відповідності ідентифікаторів атрибутів їх назвам (labels.csv), а також розширений файл метаданих (weird_images_w_labels.csv), що містить інформацію про розміри зображень, статистики інтенсивностей та перелік атрибутів для кожного зразка. Ці дані надходять до підсистеми підготовки даних, де здійснюється зчитування метаданих, формування багатоміткового векторного подання міток та попередня обробка зображень.

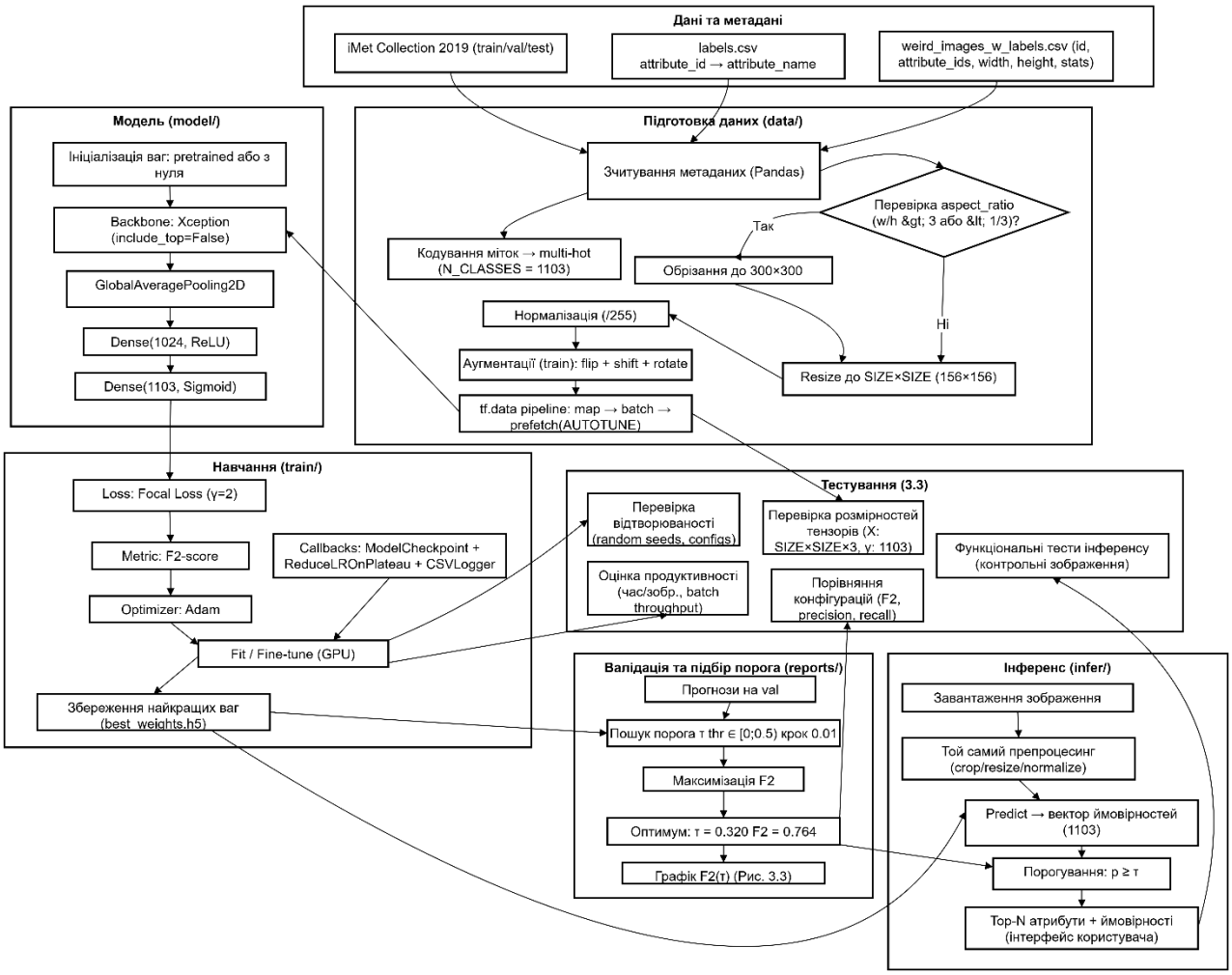


Рисунок 3.1 – Структура програмного проєкту (дерево файлів/модулів) та інформаційні зв’язки між компонентами

Ключовим елементом підготовки даних є перевірка співвідношення сторін зображення $aspect\_ratio = \frac{width}{height}$. Якщо виконується умова $aspect\_ratio >$ Забо $aspect\_ratio < \frac{1}{3}$, застосовується обрізання до розміру 300×300 пікселів із подальшим масштабуванням до цільового розміру 156×156. В інших випадках виконується лише масштабування без обрізання. Такий підхід дозволяє мінімізувати геометричні спотворення та забезпечити однорідність вхідних тензорів. Після масштабування виконується нормалізація інтенсивностей пікселів шляхом приведення їх до інтервалу $[0; 1]$. Для підвищення узагальнювальної здатності моделі на етапі навчання застосовуються аугментації: горизонтальне віддзеркалення, випадкові зсуви по горизонталі та вертикалі, а також випадкові повороти в заданому діапазоні кутів. Усі ці операції

інтегровані в конвеєр tf.data, що включає етапи map → batch → prefetch(AUTOTUNE) і забезпечує ефективну потокову обробку.

Архітектурною основою моделі є згорткова нейронна мережа Xception без верхнього класифікаційного шару (include_top=False). Після блоку глибинних згорткових шарів застосовується шар глобального середнього згортання (GlobalAveragePooling2D), що зменшує просторову розмірність ознак, далі — повнозв'язний шар із 1024 нейронами та активацією ReLU, і фінальний шар із 1103 нейронами та сигмоїдною активацією. Сигмоїдна функція дозволяє інтерпретувати вихід кожного нейрона як незалежну ймовірність належності до відповідного атрибута.

Підсистема навчання реалізує двоетапну стратегію: початкове «розігрівання» моделі з фіксацією частини шарів та подальше тонке налаштування (fine-tuning) усіх параметрів. Як функцію втрат використано focal loss з параметром $\gamma = 2$, що зменшує вплив легко класифікованих прикладів і частково компенсує дисбаланс міток. Основною метрикою оцінювання обрано F_2-міру, яка надає більшої ваги повноті (recall) порівняно з точністю (precision) і є релевантною для багатоміткових задач із великою кількістю рідкісних класів. Для контролю навчання використовуються механізми ModelCheckpoint (збереження найкращих ваг), ReduceLROnPlateau (адаптивне зменшення швидкості навчання) та CSVLogger (фіксація журналу метрик). Збережені ваги (best_weights.h5) передаються до модулів валідації та інференсу.

Блок валідації та підбору порога (reports/) реалізує процедуру пошуку оптимального фіксованого порога τ у діапазоні $[0; 0{,}5)$ з кроком 0,01. Для кожного значення порога обчислюється метрика F_2, після чого обирається значення, що максимізує її. За результатами експерименту оптимальним виявився поріг $\tau = 0{,}320$ із відповідним значенням $F_2 = 0{,}764$, що підтверджує доцільність емпіричного налаштування правила бінаризації ймовірностей.

Підсистема тестування забезпечує перевірку коректності реалізації: відповідність розмірностей тензорів (SIZE×SIZE×3 для зображень та 1103 для міток), відтворюваність експериментів за фіксованих випадкових зерен,

функціональні тести інференсу на контрольних зображеннях та оцінку продуктивності (час оброблення одного зображення та пропускна здатність пакетного режиму). Підсистема інференсу використовує той самий препроцесинг, що й на етапі навчання, формує вектор ймовірностей розмірності 1103 і застосовує правило $p \geq \tau$ для визначення фінального набору атрибутів, який може бути поданий користувачу у вигляді списку топ-N міток із відповідними значеннями ймовірності.

Таким чином, структурно-функціональна організація, подана на рисунку 3.1, відображає цілісну архітектуру програмного модуля, що поєднує підготовку даних, навчання глибинної моделі, емпіричну оптимізацію порогових параметрів, тестування та прикладний інференс у межах єдиного відтворюваного конвеєра.

3.2 Конвеєр підготовки даних та інференсу

Конвеєр підготовки даних для задачі багатоміткової класифікації зображень культурної спадщини побудовано на базі механізму tf.data, що дозволяє організувати потокове завантаження, попередню обробку й подальший інференс без використання окремих графічних інтерфейсів. Усі операції — від читання файлів до подачі батчів у модель — реалізовані програмно, у вигляді послідовності функцій Python/TensorFlow, що забезпечує відтворюваність експериментів та спрощує подальше тестування.

Вихідні мітки зображень подано у вигляді рядків із переліком індексів атрибутів (наприклад, "51 616 734 813"), які перетворюються на багатоміткові вектори-індикатори довжини (N_{\text{classes}} = 1103). Для кожного зображення формується рядок матриці categorical_labels, де елементи з індексами, що входять до відповідного списку, набувають значення 1, решта — 0. У результаті одержано масив розмірності ((109237 \times 1103)), який використовується як цільова змінна під час навчання та інференсу. Такий формат

є стандартним для багатоміткової класифікації й забезпечує сумісність із обраною функцією втрат (focal loss) та метрикою (F_2).

Завантаження зображень і прив'язка до міток реалізовані у функції load_img_label(). На вхід вона отримує ідентифікатор зображення та його багатомітковий вектор, читає файл формату PNG з каталогу навчальних даних і декодує його у тензор. Для запобігання спотворенням у випадку "нестандартних" співвідношень сторін запроваджено правило: якщо відношення ширини до висоти $\frac{w}{h} > 3$ або $\frac{w}{h} < \frac{1}{3}$, до зображення застосовується випадкове кадрування до розміру 300×300 пікселів. У протилежному випадку зображення використовують без попереднього кадрування. На наступному кроці всі зображення масштабуються до фіксованого розміру 156×156 пікселів і нормуються множенням інтенсивностей на $(1/255)$, що переводить значення у діапазон $([0;1])$.

Аугментація даних здійснюється у функції _augment(), параметри якої задаються окремо для навчального та валідаційного режимів. У навчальному режимі використано конфігурацію tr_cfg, яка включає: випадкове горизонтальне віддзеркалення, зсуви по горизонталі та вертикалі в межах 20% від розміру зображення, а також випадкове обертання в діапазоні $([-15^\circ; +15^\circ])$. Поєднання цих перетворень формує інваріантність моделі до помірних геометричних деформацій та виконує роль регуляризатора, зменшуючи ризик перенавчання на окремих фрагментах колекції. Для валідаційної вибірки використовується спрощена конфігурація val_cfg (масштабування та нормування без аугментацій), що забезпечує стабільну та зіставну оцінку якості.

Створення потоків даних для навчання, валідації та подальшого інференсу реалізовано функцією create_dataset(). На основі списків імен файлів і відповідних багатоміткових векторів формується об'єкт tf.data.Dataset. У навчальному режимі до нього послідовно застосовуються операції shuffle_and_repeat (перемішування й повторення вибірки), map із викликом load_img_label() та _augment() (з параметром num_parallel_calls для паралельної обробки), batch (формування батчів заданого розміру) і prefetch(AUTOTUNE)

(попереднє завантаження наступних батчів). У режимі інференсу використовується аналогічна послідовність без стохастичних аугментацій; завдяки цьому навчання та тестування експлуатують один і той самий конвеєр, відрізняючись лише параметрами препроцесингу.

Загальні налаштування препроцесингу для навчального й валідаційного/тестового режимів узагальнені в таблиця 3.1.

Таблиця 3.1 – Параметри конвеєра tf.data для різних режимів роботи

Етап обробки	Навчання	Валідація/ інференс
Зчитування зображень	PNG → тензор RGB	PNG → тензор RGB
Обробка співвідношення сторін	випадкове кадрування до (300\times300) при (w/h>3) або (w/h<1/3)	те саме
Масштабування зображення	resize до (156\times156)	resize до (156\times156)
Нормування інтенсивностей	множення на (1/255)	множення на (1/255)
Геометричні аугментації	горизонтальне віддзеркалення, зсуви 0.2, обертання ([-15^\circ; +15^\circ])	відсутні
Формування потоків даних	shuffle_and_repeat → map → batch → prefetch	repeat → map → batch → prefetch

Наступним важливим елементом реалізації є безпосередньо архітектура моделі, яка інтегрується з описаним конвеєром даних. Як базову згорткову мережу використано Xception без верхньої (класичної) частини (include_top=False). На виході згорткового “стрижня” застосовано шар глобального усереднення ознак GlobalAveragePooling2D, після якого розміщено повнозв’язний шар на 1024 нейрони з активацією ReLU та вихідний шар Dense(1103) з сигмоїдною активацією, що формує вектор імовірностей для кожного атрибуту. Загальна кількість параметрів моделі становить 24 090 231, із яких 24 035 703 є тренованими, а 54 528 — нетренованими (параметри нормалізації). Попередньо навчені ваги завантажуються з зовнішнього файлу, що зменшує час збіжності та покращує початкову якість.

Текстове представлення архітектури, сформоване стандартною функцією Keras model.summary(), доцільно включити до пояснювальної записки як окремий рисунок (рисунок 3.2). Для цього у середовищі виконання (Kaggle/Google Colab) викликається model.summary(), отриманий консольний вивід зберігається як зображення (скріншот) і вставляється в документ із відповідним підписом. Такий рисунок фіксує повний перелік шарів, їх вихідні розмірності та кількість параметрів і виконує роль “машинно-читаної специфікації” архітектури, що особливо важливо при відсутності графічних структурних схем.

In [23]:
```
custom_weight = True
training = False
```

In [24]:
```
def create_model(input_shape, n_out):
    input_tensor = tf.keras.layers.Input(shape=input_shape)
    base_model = tf.keras.applications.Xception(include_top=False,
                   weights=None,
                   input_tensor=input_tensor)
    if(not custom_weight):
        base_model.load_weights('../input/xception/xception_weights_tf_dim_ordering_tf_kernels_no
top.h5')
#     x = Conv2D(32, kernel_size=(1,1), activation='relu')(base_model.output)
#     x = Flatten()(x)
    x = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
    # x = Dropout(0.5)(x)
    x = tf.keras.layers.Dense(1024, activation='relu')(x)
    # x = Dropout(0.5)(x)
    final_output = tf.keras.layers.Dense(n_out, activation='sigmoid', name='final_output')(x)
    model = tf.keras.Model(input_tensor, final_output)
    if(custom_weight):
        model.load_weights('../input/imetpretrained/imet_xception_val_f2_0.686829-f2_0.611568.h
5')

    return model
```

In [25]:
```
model = create_model(input_shape=(SIZE, SIZE, 3), n_out=N_CLASSES)
model.summary()
```

Рисунок 3.2 – Фрагмент коду функції create_model() для побудови Xception-подібної згорткової нейронної мережі з попередньо натренованими вагами

Агреговані характеристики основних блоків моделі наведено в таблиця 3.2, що дозволяє акцентувати увагу на співвідношенні параметрів між базовою згортковою частиною та доданою класифікаційною “головою”.

Таблиця 3.2 – Узагальнена структура моделі

Блок моделі	Тип шарів	Кількість параметрів	Частка від загальної кількості	Призначення
Згортковий стрижень на базі Xception	послідовності Conv2D, Separable Conv2D, BatchNorm, MaxPooling	20 861 480	≈ 86,6 %	виділення високорівневих візуальних ознак
Перехідний повнозв’язний шар	GlobalAverage Pooling2D + Dense(1024, ReLU)	2 098 176	≈ 8,7 %	узагальнення ознак у компактне представлення
Вихідний багатомітковий класифікатор	Dense(1103, sigmoid)	1 130 575	≈ 4,7 %	оцінка імовірностей належності до атрибутів
Разом	–	24 090 231	100 %	повна модель для інференсу атрибутів

Таким чином, у підрозділі 3.2 детально описано реалізацію програмного конвеєра підготовки даних на основі tf.data та архітектуру багатоміткової моделі класифікації, без використання графічних схем чи інтерфейсних екранів. Отримана специфікація є достатньою для відтворення експериментів і подальшого аналізу якості; у наступному підрозділі 3.3 буде розглянуто

процедури тестування, обрані метрики та результати оцінювання побудованої моделі.

3.3 Тестування та результати перевірки

Для кількісної оцінки якості моделі було сформовано валідаційну вибірку на основі 50 % усіх навчальних даних. За допомогою конвеєра tf.data (підрозділ 3.2) зображення завантажувалися батчами розміром 512, що дало змогу опрацювати 6784 приклади без проміжного збереження в оперативній пам’яті. Для кожного зразка обчислювали ймовірність належності до 1103 можливих атрибутів та вектор істинних міток у one-hot поданні.

Оскільки задача є багатомітковою, основною метрикою було обрано макро-F_2-міру, яка сильніше штрафує втрати за рахунок неповного покриття істинних міток (низький recall). На основі попередньо збережених предиктів валідаційної вибірки (lastFullValPred розмірності 6784×1103) та істинних міток (lastFullValLabels) було реалізовано окрему функцію my_f2, що рахує кількість істинно позитивних, хибно позитивних та хибно негативних спрацювань і повертає глобальне значення F_2 для всієї вибірки.

Далі виконувався пошук оптимального порога бінаризації ймовірностей. Функція find_best_fixed_threshold перебирала порогові значення $t \in [0; 0{,}49]$з кроком 0,01 (усього 50 ітерацій) і для кожного порога обчислювала значення F_2. За результатами пошуку найкращим виявився поріг $t^* = 0{,}32$, якому відповідає максимальне значення метрики $F_2 = 0{,}764$. Графічно залежність F_2 від порога показано на рисунку 3.3: крива має чіткий максимум у діапазоні 0,30–0,34, а відхилення порога в бік менших значень призводить до зростання кількості хибно позитивних міток, у бік більших – до втрати істинних міток (падіння recall).

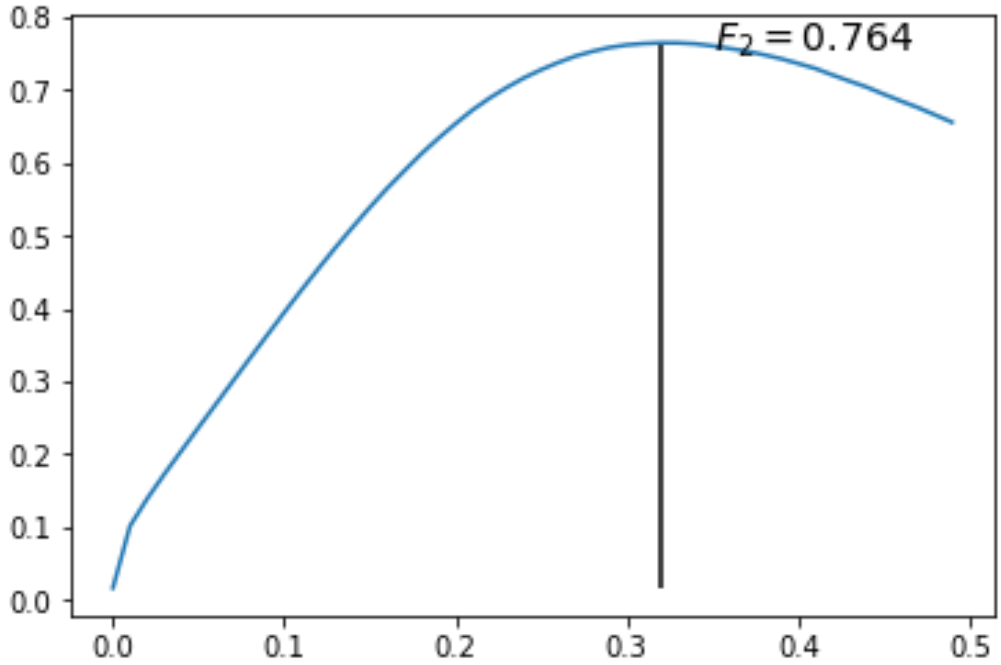


Рисунок 3.3 – Залежність значення макро-F_2 від порогового значення tдля бінаризації ймовірностей; максимум $F_2 = 0{,}764$ досягається при порозі $t^* = 0{,}32$

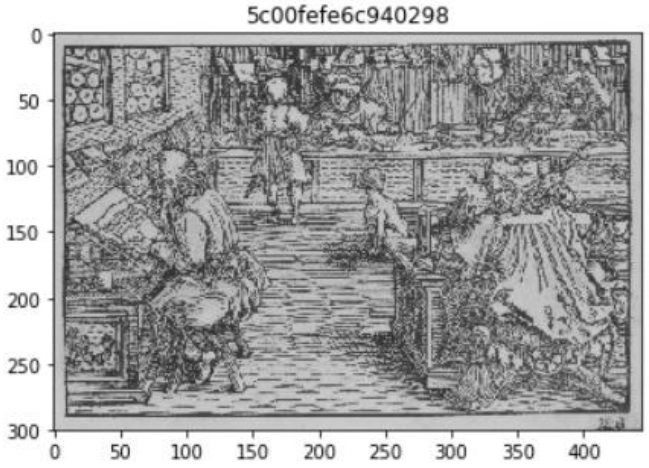
Отриманий поріг t^* використовувався як фіксоване значення для подальшого інференсу на тестовому наборі змагання. Для цього було сформовано датасет submit_ds на основі 7443 тестових зображень. Інференс з використанням попередньо натренованих ваг моделі (model.predict_generator) складав 117 кроків і тривав близько 30 секунд, що відповідає середньому часу передбачення ~0,25 с на один батч. Після застосування порога t^*до векторів ймовірностей для кожного зображення формувався список індексів атрибутів, який експортувався у файл submission.csv для подальшого завантаження на платформу змагання.

Для аналізу характеру передбачень було побудовано розподіл кількості присвоєних атрибутів на одне зображення (таблиця 3.3). Середня кількість міток на об’єкт становить 4,6, при цьому 66,4 % зображень отримали від 2 до 5 атрибутів, а лише близько 3 % – більше 10 міток. Це свідчить про те, що обрана комбінація порога t^*та архітектури Xception не схильна до надмірної гіперпередбачуваності (assign-all), але водночас зберігає достатньо високе покриття семантичних ознак.

Таблиця 3.3 – Розподіл кількості передбачених атрибутів на одне тестове зображення

Кількість міток на зображенні	Кількість зображень
1	317
2	1160
3	1430
4	1345
5	1010
6	768
7	495
8	311
9	244
10	142
11	98
12	51
13	34
14	17
15	10
16	7
17	3
18	1

Для якісного аналізу було розглянуто кілька характерних прикладів відповідності та розбіжностей між істинними й передбаченими мітками (рисунки 3.4–3.9). На рисунку 3.4 наведено сцену майстерні. Істинний набір атрибутів містить 5 міток (культура, присутність собак, чоловіків, робочий контекст, написання тексту), тоді як модель передбачила 4 мітки, з яких лише 2 збігаються з істинними. Таким чином, локальний recall становить 40 % (2/5), precision – 50 % (2/4); помилки стосуються як культури, так і деталізації сюжету («dogs», «working», «writing»).




```
True labels = culture::german, tag::dogs, tag::men, tag::working, tag::writing
Predicted labels = culture::american, culture::german, tag::men, tag::women
```

Рисунок 3.4 – Приклад передбачення для сцени майстерні: істинні мітки (5 атрибутів) та передбачені моделлю мітки (4 атрибути) з двома збігами

На рисунку 3.5 показано гравюру з британським написом. Тут істинно задано 3 атрибути, модель повертає 4 мітки, причому 2 з них збігаються з еталоном («culture::british», «tag::men»). Відповідно, локальний recall дорівнює 66,7 % (2/3), precision – 50 % (2/4); додаткові мітки («satire», «women») відображають стилістичні особливості зображення, які не були включені до ручної розмітки.

Приклад на рисунку 3.6 демонструє гірший випадок: істинні мітки описують італійський об’єкт з бісером та написами (3 атрибути), натомість модель пропонує три альтернативні культурні кластери («byzantine», «egyptian», «roman»), жоден з яких не збігається з еталоном. Локальні значення precision та recall тут дорівнюють 0 %, що вказує на типову помилку класифікації культурно споріднених артефактів.


```
True labels = culture::british, tag::inscriptions, tag::men
Predicted labels = culture::british, tag::men, tag::satire, tag::women
```

Рисунок 3.5 – Британська сатирична гравюра: порівняння істинних (3) та передбачених (4) атрибутів із двома збігами та двома надлишковими тегами


```
True labels = culture::italian, tag::beads, tag::inscriptions
Predicted labels = culture::byzantine, culture::egyptian, culture::roman
```

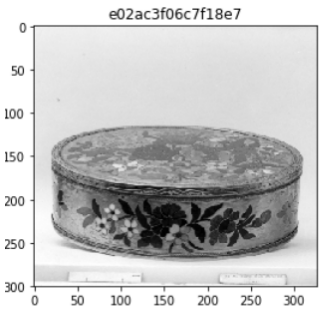
Рисунок 3.6 – Приклад культурної плутанини: італійський артефакт з написами, для якого модель передбачає альтернативні культурні кластери без жодного збігу з еталоном

На рисунку 3.7 наведено орнаментальні рамки французького походження. Еталон містить 2 мітки («culture::french», «tag::decorative elements»), тоді як модель повертає 6 атрибутів, з яких 2 збігаються з істинними. Таким чином, локальний recall дорівнює 100 % (2/2), а precision – лише 33 % (2/6): модель правильно «перекриває» еталонні ознаки, але додає низку споріднених декоративних тегів («architecture», «design elements», «flowers», «ornament»), що збільшує семантичну насиченість опису ціною зниження точності.


```
True labels = culture::french, tag::decorative elements
Predicted labels = culture::french, tag::architecture, tag::decorative elements, tag::design e
lements, tag::flowers, tag::ornament
```

Рисунок 3.7 – Орнаментальні рамки французької культури: повне покриття двох еталонних атрибутів та додавання чотирьох додаткових декоративних тегів

Приклад на рисунку 3.8 ілюструє випадок змішаного культурного контексту. Істинний опис включає 5 атрибутів (німецька культура, квіти, листя, утилітарний об’єкт, «turkish or venice»), тоді як модель повертає 6 міток, з яких 3 збігаються з еталоном. Локальний recall становить 60 % (3/5), precision – 50 % (3/6). Правильно захоплюються матеріально-функціональні ознаки («flowers», «leaves», «utilitarian objects»), однак культурна приналежність зміщується в бік французьких/паризьких тегів.




```
True labels = culture::german, tag::flowers, tag::leaves, tag::utilitarian objects, culture::t
urkish or venice
Predicted labels = culture::french, culture::paris, tag::boxes, tag::flowers, tag::leaves, ta
g::utilitarian objects
```

Рисунок 3.8 – Утилітарний об’єкт з квітковим орнаментом: частковий збіг культурно-функціональних атрибутів (3 з 5) за наявності зміщення культурних тегів

Нарешті, на рисунку 3.9 показано грецьку чашу з фігурами. Істинний набір містить 4 атрибути, модель передбачає також 4, причому збігаються 2 («culture::greek», «tag::cups»). Локальні значення precision та recall однакові – 50 % (2/4); помилки стосуються деталізації сюжету (специфічні фігури й персонажі замінюються більш загальним тегом «men» та додатковою культурною міткою «attic»).


```
True labels = culture::greek, tag::cups, tag::greek figures, tag::satyrs
Predicted labels = culture::attic, culture::greek, tag::cups, tag::men
```

Рисунок 3.9 – Грецька чаша з фігуративною сценою: збіг двох із чотирьох еталонних атрибутів та помилки у деталізації сюжету й додатковій культурній мітці

Кількісний аналіз (значення F_2 для валідаційної вибірки, розподіл кількості міток, локальні precision/recall на окремих прикладах) демонструє, що модель Xception із налаштованим порогом $t^* = 0{,}32$ забезпечує прийнятний баланс між повнотою та точністю багатоміткової класифікації атрибутів культурних об’єктів. Водночас приклади з рисунків 3.4–3.9 виявляють систематичні зони помилок – насамперед у розрізненні близьких культурних кластерів та у надлишковому присвоєнні декоративних тегів, – які необхідно враховувати при подальшому вдосконаленні моделі.

ВИСНОВКИ

Узагальнюючи результати проведеного дослідження, можна зробити висновок, що всі поставлені завдання виконано, а мети роботи досягнуто.

1. Щодо першого завдання, було здійснено системний аналіз предметної області атрибуції мистецьких об’єктів на основі зображень, сформовано таксономію атрибутів (змістові, стилістичні, культурно-історичні, матеріально-технічні та атрибути цифрового відтворення) та обґрунтовано формалізацію задачі у вигляді багатоміткової класифікації з довгохвостим розподілом міток. Проаналізовано сучасні підходи до багатоміткової атрибуції мистецьких зображень, зокрема рішення на основі перенесення навчання в згорткових нейронних мережах, спеціалізовані функції втрат для умов дисбалансу класів та стратегії порогування, узгоджені з F-орієнтованими метриками якості. Це дозволило виокремити методологічно зрілу парадигму, на якій ґрунтується подальша розробка модуля.

2. У межах другого завдання досліджено структуру та особливості набору даних iMet Collection 2019, що слугує базою інформаційного забезпечення модуля. Визначено загальний обсяг даних (155 531 зображення), розподіл на навчальну, валідаційну та тестову підвибірки, кількість атрибутів (1103 мітки) та характер розмітки (multi-hot подання). Проаналізовано статистики розмірів зображень і співвідношень сторін, виявлено групу зображень із екстремальним aspect ratio, для яких обґрунтовано застосування спеціальної стратегії попередньої обробки (кадрування 300×300 з подальшим масштабуванням). Показано, що iMet має довгохвостий розподіл міток та значний дисбаланс атрибутів, що безпосередньо впливає на вибір функцій втрат і процедур оцінювання.

3. Третє завдання реалізовано шляхом проєктування цілісної архітектури програмного модуля, яка охоплює конвеєр підготовки даних, навчання моделі та інференсу. Запропоновано структурну декомпозицію на функціональні підсистеми (підготовка даних, модуль моделі, підсистема навчання, модуль

валідації та порогування, блок тестування та інференсу), описано інформаційні потоки між ними та зафіксовано єдиний відтворюваний конвеєр на основі механізму tf.data. Така архітектура забезпечує модульність, повторюваність експериментів, узгодженість препроцесингу на етапах навчання й інференсу та можливість подальшого розширення модуля.

4. У межах четвертого завдання реалізовано багатоміткову модель класифікації на основі згорткової нейронної мережі Xception з використанням перенесення навчання. Побудовано модель із виключенням верхнього класифікаційного шару, додано GlobalAveragePooling2D та двошаровий класифікаційний «хвіст» (Dense(1024, ReLU) + Dense(1103, sigmoid)). Загальна кількість параметрів моделі становить 24 090 231, з яких 24 035 703 є тренованими. Застосовано двоетапну стратегію навчання (початкове донавчання верхніх шарів та подальше fine-tuning), використано функцію втрат типу focal loss для часткової компенсації дисбалансу та налаштовано оптимізатор і механізми керування навчанням (ModelCheckpoint, ReduceLROnPlateau, CSVLogger), що забезпечило стабільну збіжність на великомасштабному датасеті.

5. П’яте завдання виконано шляхом налаштування процедур порогування та оцінювання якості моделі за F_2-мірою на валідаційній вибірці, а також аналізу розподілу передбачених атрибутів і типових помилок. На основі предиктів для 6784 валідаційних прикладів було реалізовано перебір глобального порога в діапазоні [0; 0,49] із кроком 0,01 та обчислення F_2. Встановлено, що оптимальним є поріг $t^* = 0{,}32$, для якого досягнуто значення $F_2 = 0{,}764$. Додатково досліджено розподіл кількості передбачених міток на зображення: більшість об’єктів отримують від 2 до 5 атрибутів, середня кількість міток становить близько 4–5, а випадки з понад 10 атрибутами є поодинокими. Проведений якісний аналіз окремих прикладів виявив типові зони помилок – плутанину між культурно спорідненими кластерами, частковий пропуск сюжетних деталей та надлишкове присвоєння декоративних тегів.

6. Шосте завдання реалізовано через інтегральну оцінку ефективності розробленого модуля та формулювання рекомендацій щодо його застосування в задачах цифрової каталогізації мистецьких колекцій. Показано, що модуль на основі Xception і конвеєра tf.data забезпечує прийнятний компроміс між якістю (F_2 = 0,764 на валідаційній підвибірці) та обчислювальною вартістю (інференс тестового набору за десятки секунд у пакетному режимі). Отримані результати свідчать про практичну придатність модуля для автоматизованого поповнення або уточнення атрибутивних описів у великих музейних колекціях. Рекомендовано застосовувати модуль як допоміжний інструмент для фахівців-кураторів, поєднуючи його з ручною верифікацією для складних або рідкісних атрибутів, а також розглядати подальший розвиток у напрямі адаптації до локальних колекцій, введення атрибутно-залежних порогів та використання мультимодальної інформації (зображення + текстові метадані) для підвищення точності культурно-історичної атрибуції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tsoumakas G., Katakis I. Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining. 2007. Vol. 3(3). P. 1–13. DOI: 10.4018/jdwm.2007070101.

2. Zhang M.-L., Zhou Z.-H. A Review on Multi-Label Learning Algorithms. IEEE Transactions on Knowledge and Data Engineering. 2014. Vol. 26(8). P. 1819–1837. DOI: 10.1109/TKDE.2013.39.

3. Tarekegn A. N., Ullah M., Cheikh F. A. Deep Learning for Multi-Label Learning: A Comprehensive Survey. arXiv:2401.16549. 2024. URL: https://arxiv.org/abs/2401.16549 (дата звернення: 23.02.2026).

4. Zhang C., Kaeser-Chen C., Vesom G., Choi J., Kessler M., Belongie S. The iMet Collection 2019 Challenge Dataset. arXiv:1906.00901. 2019. DOI: 10.48550/arXiv.1906.00901. URL: https://arxiv.org/abs/1906.00901 (дата звернення: 23.02.2026).

5. iMet Collection 2019 – FGVC6 (Kaggle Competition). URL: https://kaggle.com/competitions/imet-2019-fgvc6 (дата звернення: 23.02.2026).

6. Strezoski G., Worring M. OmniArt: Multi-task Deep Learning for Artistic Data Analysis. arXiv:1708.00684. 2017. URL: https://arxiv.org/abs/1708.00684 (дата звернення: 23.02.2026).

7. Mao H., Cheung M., She J. DeepArt: Learning Joint Representations of Visual Arts. ACM International Conference on Multimedia (MM). 2017. URL: https://deepart.hkust.edu.hk/ART500K/art500k.html (дата звернення: 23.02.2026).

8. Mensink T., van Gemert J. The Rijksmuseum Challenge: Museum-Centered Visual Recognition. Proceedings of the International Conference on Multimedia Retrieval (ICMR ’14). 2014. DOI: 10.1145/2578726.2578791.

9. Castellano G., Vessio G. Deep learning approaches to pattern extraction and recognition in paintings and drawings: an overview. Neural Computing and Applications. 2021. DOI: 10.1007/s00521-021-05893-z.

10. Zhao W., Zhou D., Qiu X., Jiang W. Compare the performance of the models in art classification. PLOS ONE. 2021. Vol. 16(3). e0248414. DOI: 10.1371/journal.pone.0248414.

11. Saleh B., Elgammal A. Large-Scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature. International Journal for Digital Art History. 2016. No. 2. DOI: 10.11588/dah.2016.2.23376.

12. Karayev S., Trentacoste M., Han H., Agarwala A., Darrell T., Hertzmann A., Winnemoeller H. Recognizing Image Style. Proceedings of the British Machine Vision Conference (BMVC). 2014. DOI: 10.5244/C.28.122.

13. Pan S. J., Yang Q. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering. 2010. Vol. 22(10). P. 1345–1359. DOI: 10.1109/TKDE.2009.191.

14. Yosinski J., Clune J., Bengio Y., Lipson H. How transferable are features in deep neural networks? Advances in Neural Information Processing Systems (NeurIPS 2014). 2014. URL: https://arxiv.org/abs/1411.1792 (дата звернення: 23.02.2026).

15. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90.

16. Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). 2016. P. 2818–2826. DOI: 10.1109/CVPR.2016.308.

17. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556. 2014. DOI: 10.48550/arXiv.1409.1556. URL: https://arxiv.org/abs/1409.1556 (дата звернення: 23.02.2026).

18. Huang G., Liu Z., van der Maaten L., Weinberger K. Q. Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). 2017. DOI: 10.1109/CVPR.2017.243.

19. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of Machine Learning Research. 2019. Vol. 97. P. 6105–

6114. URL: https://proceedings.mlr.press/v97/tan19a.html (дата звернення: 23.02.2026).

20. Lin T.-Y., Goyal P., Girshick R., He K., Dollár P. Focal Loss for Dense Object Detection. arXiv:1708.02002. 2017. DOI: 10.48550/arXiv.1708.02002. URL: https://arxiv.org/abs/1708.02002 (дата звернення: 23.02.2026).

21. Ridnik T., Ben-Baruch E., Zamir N., Noy A., Friedman I., Protter M., Zelnik-Manor L. Asymmetric Loss for Multi-Label Classification. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021). 2021. P. 82–91. DOI: 10.1109/ICCV48922.2021.00015.

22. Cui Y., Jia M., Lin T.-Y., Song Y., Belongie S. Class-Balanced Loss Based on Effective Number of Samples. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019). 2019. DOI: 10.1109/CVPR.2019.00949.

23. Draszawka K., Szymański J. From Scores to Predictions in Multi-Label Classification: Neural Thresholding Strategies. Applied Sciences. 2023. Vol. 13(13). Art. 7591. DOI: 10.3390/app13137591.

24. Bénédict G., Koops H. V., Odijk D., de Rijke M. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. Transactions on Machine Learning Research. 2022. DOI: 10.48550/arXiv.2108.10566. URL: https://arxiv.org/abs/2108.10566 (дата звернення: 23.02.2026).

25. Wang Q., Jia N., Breckon T. P. A Baseline for Multi-Label Image Classification Using an Ensemble of Deep Convolutional Neural Networks. 2019 IEEE International Conference on Image Processing (ICIP). 2019. DOI: 10.1109/ICIP.2019.8803793. URL: https://arxiv.org/abs/1811.08412 (дата звернення: 23.02.2026).

26. Chua T.-S., Tang J., Hong R., Li H., Luo Z., Zheng Y. NUS-WIDE: a real-world web image database from National University of Singapore. Proceedings of the ACM International Conference on Image and Video Retrieval. 2009. DOI: 10.1145/1646396.1646452.

27. Lin T.-Y., Maire M., Belongie S., et al. Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014. Lecture Notes in Computer Science. 2014. Vol. 8693. P. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

28. Dosovitskiy A., Beyer L., Kolesnikov A., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929. 2020. DOI: 10.48550/arXiv.2010.11929. URL: https://arxiv.org/abs/2010.11929 (дата звернення: 23.02.2026).

29. Radford A., Kim J. W., Hallacy C., et al. 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 (дата звернення: 23.02.2026).

30. Zhang C., Kaeser-Chen C., Vesom G., Choi J., Kessler M., Belongie S. The iMet Collection 2019 Challenge Dataset. arXiv preprint arXiv:1906.00901, 2019. URL: https://arxiv.org/abs/1906.00901

32. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп'яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

33. Мелень В., Патраманська А., Пилипів Л., Ліп'яніна-Гончаренко Х.. Комплексне застосування методів штучного інтелекту для розпізнавання та генерації об'єктів культурної спадщини. *У матеріалах X Міжнародної мультидисциплінарної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень».* С. 90–92.

Додаток А

Програмний код

```
import os
import functools
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split

tf.enable_eager_execution()
AUTOTUNE = tf.data.experimental.AUTOTUNE

# ------------------------------------------------------------------------
# 1. Константи та завантаження метаданих
# ------------------------------------------------------------------------

SIZE = 156
BATCH_SIZE = 64

train_val_df = pd.read_csv("../input/imetmetadata/weird_images_w_labels.csv")
labels_df    = pd.read_csv("../input/imet-2019-fgvc6/labels.csv")
N_CLASSES    = len(labels_df)

# ------------------------------------------------------------------------
# 2. Формування багатоміткових векторів (multi-hot)
# ------------------------------------------------------------------------

categorical_labels = np.zeros((len(train_val_df), N_CLASSES), dtype=np.float32)
for i, s in enumerate(train_val_df["attribute_ids"]):
   idx = [int(k) for k in s.split()]
   categorical_labels[i, idx] = 1.0

# Розбиття на навчальну / валідаційну вибірки
train_ids, val_ids, y_train, y_val = train_test_split(
   train_val_df["id"].values,
   categorical_labels,
   test_size=0.2,
   random_state=42
)

# ------------------------------------------------------------------------
# 3. Конвеєр підготовки зображень (tf.data) з урахуванням aspect ratio
# ------------------------------------------------------------------------

def load_img_label(image_id, label):
   img_bytes = tf.read_file("../input/imet-2019-fgvc6/train/" + image_id + ".png")
   img = tf.image.decode_png(img_bytes, channels=3)

   shape = tf.shape(img)
   w = tf.cast(shape[1], tf.float32)
   h = tf.cast(shape[0], tf.float32)

   # Для екстремальних співвідношень сторін виконується кроп 300×300
   img = tf.cond(
      tf.logical_or(w / h > 3.0, w / h < 1.0 / 3.0),
      lambda: tf.image.random_crop(img, size=[300, 300, 3]),
      lambda: img
```

```
    )

    # Масштабування до розміру 156×156 та проста аугментація
    img = tf.image.resize_images(img, [SIZE, SIZE])
    img = tf.image.random_flip_left_right(img)
    img = tf.cast(img, tf.float32) / 255.0

    return img, label

def make_dataset(ids, labels, shuffle=True):
    ds = tf.data.Dataset.from_tensor_slices((ids, labels))
    if shuffle:
        ds = ds.shuffle(buffer_size=len(ids)).repeat()
    ds = ds.map(load_img_label, num_parallel_calls=AUTOTUNE)
    ds = ds.batch(BATCH_SIZE).prefetch(AUTOTUNE)
    return ds

train_ds = make_dataset(train_ids, y_train, shuffle=True)
val_ds   = make_dataset(val_ids,   y_val,   shuffle=False)

# ---------------------------------------------------------------------
# 4. Функція втрат (focal loss) та метрика F2-score
# ---------------------------------------------------------------------

gamma = 2.0
eps   = tf.keras.backend.epsilon()

def focal_loss(y_true, y_pred):
    pt = y_pred * y_true + (1.0 - y_pred) * (1.0 - y_true)
    pt = tf.keras.backend.clip(pt, eps, 1.0 - eps)
    ce = -tf.log(pt)
    fl = tf.pow(1.0 - pt, gamma) * ce
    return tf.keras.backend.mean(tf.keras.backend.sum(fl, axis=1))

def f2_score(y_true, y_pred):
    beta2 = 4.0  # beta = 2 => beta^2 = 4
    y_pred_bin = tf.round(tf.clip_by_value(y_pred, 0.0, 1.0))

    tp = tf.reduce_sum(y_true * y_pred_bin, axis=1)
    fp = tf.reduce_sum((1.0 - y_true) * y_pred_bin, axis=1)
    fn = tf.reduce_sum(y_true * (1.0 - y_pred_bin), axis=1)

    precision = tp / (tp + fp + eps)
    recall    = tp / (tp + fn + eps)

    return tf.reduce_mean((1.0 + beta2) * precision * recall /
                          (beta2 * precision + recall + eps))

# ---------------------------------------------------------------------
# 5. Архітектура моделі Xception з класифікаційним «хвостом»
# ---------------------------------------------------------------------

def create_model(input_shape=(SIZE, SIZE, 3), n_out=N_CLASSES):
    inp = tf.keras.layers.Input(shape=input_shape)
    base = tf.keras.applications.Xception(
        include_top=False,
        weights=None,          # у роботі використовувались попередньо навчені ваги
        input_tensor=inp
    )
    x = tf.keras.layers.GlobalAveragePooling2D()(base.output)
```

```
    x = tf.keras.layers.Dense(1024, activation="relu")(x)
    out = tf.keras.layers.Dense(n_out, activation="sigmoid", name="final_output")(x)
    model = tf.keras.Model(inp, out)
    return model

model = create_model()
model.compile(
    loss=focal_loss,
    optimizer=tf.keras.optimizers.Adam(lr=3e-4),
    metrics=[f2_score]
)

# ---------------------------------------------------------------------
# 6. Навчання моделі з використанням callback-функцій
# ---------------------------------------------------------------------

steps_per_epoch = len(train_ids) // BATCH_SIZE
val_steps       = len(val_ids)   // BATCH_SIZE

callbacks = [
    tf.keras.callbacks.ModelCheckpoint(
        "imet_xception_best.h5",
        monitor="val_loss",
        save_best_only=True,
        save_weights_only=True,
        verbose=1
    ),
    tf.keras.callbacks.ReduceLROnPlateau(
        monitor="val_loss",
        factor=0.5,
        patience=2,
        verbose=1
    ),
    tf.keras.callbacks.EarlyStopping(
        monitor="val_loss",
        patience=5,
        restore_best_weights=True,
        verbose=1
    )
]

history = model.fit(
    train_ds,
    epochs=20,
    steps_per_epoch=steps_per_epoch,
    validation_data=val_ds,
    validation_steps=val_steps,
    callbacks=callbacks,
    verbose=1
)

# ---------------------------------------------------------------------
# 7. Пошук оптимального порога за F2-мірою на валідаційній вибірці
# ---------------------------------------------------------------------

# Отримання предиктів на підмножині валідаційних даних
val_pred = model.predict(
    val_ds.take(val_steps),
    verbose=1
)
```

```
n_val      = val_pred.shape[0]
y_val_true  = y_val[:n_val]

beta_f2 = 2.0

def f2_numpy(y_true, y_prob, thr):
    y_hat = (y_prob >= thr).astype("int32")
    tp = ((y_true == 1) & (y_hat == 1)).sum()
    fp = ((y_true == 0) & (y_hat == 1)).sum()
    fn = ((y_true == 1) & (y_hat == 0)).sum()
    p = tp / (tp + fp + 1e-15)
    r = tp / (tp + fn + 1e-15)
    return (1 + beta_f2**2) * p * r / (beta_f2**2 * p + r + 1e-15)

best_thr, best_f2 = 0.0, 0.0
for thr in np.arange(0.0, 0.5, 0.01):
    score = f2_numpy(y_val_true, val_pred, thr)
    if score > best_f2:
        best_thr, best_f2 = thr, score

print("Best threshold:", best_thr, "F2 =", best_f2)

# ---------------------------------------------------------------------
# 8. Інференс на тестовій вибірці та формування multi-label виходу
# ---------------------------------------------------------------------

test_ids = pd.read_csv("../input/imet-2019-fgvc6/sample_submission.csv")["id"].values

def load_test_img(image_id):
    img_bytes = tf.read_file("../input/imet-2019-fgvc6/test/" + image_id + ".png")
    img = tf.image.decode_png(img_bytes, channels=3)

    shape = tf.shape(img)
    w = tf.cast(shape[1], tf.float32)
    h = tf.cast(shape[0], tf.float32)

    img = tf.cond(
        tf.logical_or(w / h > 3.0, w / h < 1.0 / 3.0),
        lambda: tf.image.random_crop(img, size=[300, 300, 3]),
        lambda: img
    )

    img = tf.image.resize_images(img, [SIZE, SIZE])
    img = tf.cast(img, tf.float32) / 255.0
    return img

test_ds = tf.data.Dataset.from_tensor_slices(test_ids)
test_ds = test_ds.map(load_test_img, num_parallel_calls=AUTOTUNE)
test_ds = test_ds.batch(BATCH_SIZE).prefetch(AUTOTUNE)

test_pred = model.predict(test_ds, verbose=1)

attr_ids = []
for row in test_pred:
    idx = np.where(row >= best_thr)[0]
    attr_ids.append(" ".join(str(i) for i in idx))

submit = pd.DataFrame({"id": test_ids, "attribute_ids": attr_ids})
submit.to_csv("submission.csv", index=False)
```

Додаток Б

Апробація отриманих результатів

УДК 082:001
Н 44

Голова оргкомітету: Коренюк І.О.
Верстка: Білоус Т.В.
Дизайн: Бондаренко І.В.

Рекомендовано до видання Вченою Радою Інституту науково-технічної інтеграції та співпраці. Протокол № 7 від 26.02.2026 року.

Конференцію зареєстровано Державною науковою установою «УкрІНТЕІ» в базі даних науково-технічних заходів України та бюлетені «План проведення наукових, науково-технічних заходів в Україні» (Посвідчення № 468 від 10.06.2025).

Н 44 **Наука сьогодення: від досліджень до стратегічних рішень**: матеріали X Міжнародної студентської наукової конференції, м. Одеса, 27 лютого, 2026 рік / ГО «Молодіжна наукова ліга». — Вінниця: ТОВ «УКРЛОГОС Груп», 2026. — 194 с.

ISBN 978-617-8582-23-4
DOI 10.62732/liga-inter-27.02.2026

Викладено матеріали учасників X Міжнародної мультидисциплінарної студентської наукової конференції «Наука сьогодення: від досліджень до стратегічних рішень», яка відбулася 27 лютого 2026 року у місті Одеса, Україна.

УДК 082:001

ISBN 978-617-8582-23-4

СЕКЦІЯ 14.
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

СЕКЦІЯ 15.
ТРАНСПОРТ ТА ТРАНСПОРТНІ ТЕХНОЛОГІЇ

СЕКЦІЯ 16.
ФІЛОЛОГІЯ ТА ЖУРНАЛІСТИКА

СЕКЦІЯ 17.
ПЕДАГОГІКА ТА ОСВІТА

Віталій Мелень, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Анастасія Патраманська, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Людмила Пилипів, здобувач вищої освіти факультету комп'ютерних інформаційних технологій
Західноукраїнський національний університет, Україна

Науковий керівник: Христина Ліп'яніна-Гончаренко, д-р.техн.наук, доцент, доцент кафедри інформаційно-обчислювальних систем і управління
Західноукраїнський національний університет, Україна

КОМПЛЕКСНЕ ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ТА ГЕНЕРАЦІЇ ОБ'ЄКТІВ ЦИФРОВОЇ КУЛЬТУРНОЇ СПАДЩИНИ

В епоху глобальної цифровізації суспільства питання збереження, дослідження та популяризації культурної спадщини набуває особливої актуальності. Сучасні музеї, галереї та архіви стикаються з проблемою обробки величезних масивів візуальних даних, які потребують каталогізації, детального атрибутування та реставрації у цифровому форматі. Вирішення цих завдань традиційними методами вимагає значних людських та часових ресурсів. Відтак, впровадження методів штучного інтелекту, зокрема згорткових нейронних мереж (ЗНМ) та генеративних моделей, є перспективним напрямком автоматизації цих процесів [1].

Метою даного дослідження є розробка концептуальної моделі комплексного програмного забезпечення, що об'єднує модулі розпізнавання, класифікації та генерації візуального контенту для ефективного управління об'єктами цифрової культурної спадщини.

Запропонована система складається з трьох ключових взаємопов'язаних підсистем, кожна з яких вирішує специфічний клас завдань.

Перша підсистема відповідає за багатоміткову класифікацію атрибутів мистецьких об'єктів. На відміну від традиційної однокласової класифікації, твори мистецтва мають складну семантику і вимагають одночасного визначення кількох атрибутів: епохи, стилю, жанру, автора та техніки виконання. Для вирішення цього завдання застосовано методи перенесення навчання (transfer learning) на базі глибоких згорткових нейронних мереж (наприклад, архітектур ResNet або EfficientNet). Використання попередньо натренованих моделей дозволяє суттєво зменшити час навчання та підвищити точність класифікації навіть на обмежених наборах даних [2].

Оцінка ефективності модуля багатоміткової класифікації здійснюється за допомогою метрики точності (Accuracy), що розраховується за формулою (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

де: TP – істинно позитивні результати; TN – істинно негативні результати; FP – хибно позитивні результати; FN – хибно негативні результати.

Другою важливою складовою комплексу є програмний модуль розпізнавання архітектурних стилів. Архітектурні пам'ятки є невід'ємною частиною культурного надбання. Модуль призначений для автоматичної ідентифікації стилістичної приналежності будівель (готика, бароко, модерн, конструктивізм тощо) за їх фотографіями. Архітектура цього модуля також базується на комп'ютерному зорі, проте його алгоритми оптимізовані для виявлення специфічних просторових патернів, геометричних форм та елементів декору, притаманних різним епохам.

Третя підсистема – програмний модуль генерації візуального контенту. Якщо перші два модулі виконують аналітичну функцію, то даний модуль розширює можливості взаємодії з цифровою спадщиною через синтез нових зображень. За допомогою генеративно-змагальних мереж (GAN) та дифузійних моделей стає можливим відтворення втрачених фрагментів картин, колоризація історичних чорно-білих фотографій будівель, а також генерація ілюстративних матеріалів для віртуальних екскурсій та освітніх платформ [3].

Функціональний розподіл та взаємозв'язок розроблених модулів представлено у таблиці (табл. 1).

Таблиця 1

Функціональні можливості модулів системи

Назва модуля	Базові технології	Основне призначення	Сфера застосування
Аналіз мистецьких об'єктів	ЗНМ, Transfer Learning, Multi-label	Визначення автора, стилю, епохи, техніки	Цифрові архіви, онлайн-каталоги музеїв
Розпізнавання архітектури	Комп'ютерний зір, виділення ознак	Класифікація архітектурних стилів будівель	Інтерактивні туристичні карти, реставрація
Генерація контенту	GAN, дифузійні моделі (Stable Diffusion)	Відновлення пошкоджених артефактів, створення візуалізацій	Освітні платформи, VR/AR виставки

Комплексна архітектура розробленої системи дозволяє передавати дані між модулями. Наприклад, результати класифікації архітектурного стилю пошкодженої будівлі (модуль 2) можуть слугувати вхідними параметрами (умовами) для модуля генерації (модуль 3), що дозволить алгоритму максимально автентично реконструювати її первісний вигляд.

Висновки. Розроблена концепція інтегрованого програмного комплексу демонструє високий потенціал застосування методів машинного навчання у гуманітарній сфері. Об'єднання інструментів багатоміткової класифікації мистецьких творів, розпізнавання архітектурних стилів та генерації візуального контенту створює потужну екосистему для цифрової культурної спадщини.

Застосування перенесення навчання у згорткових нейронних мережах забезпечує високу точність каталогізації, а генеративні технології відкривають нові шляхи для реставрації та візуалізації втрачених пам'яток. Подальші дослідження будуть спрямовані на програмну реалізацію запропонованої архітектури та її тестування на реальних наборах музейних даних.

Список використаних джерел:

1. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Запоріжжя : ЗНТУ, 2018. 341 с.
2. Kornilova O., & Mykhalchuk M. Application of Transfer Learning Methods for Image Classification in Digital Humanities. *Journal of Information Technologies*, 2021. Vol. 15(2). P. 45–52.
3. Goodfellow I., Pouget-Abadie J., Mirza M., et al. Generative Adversarial Networks. *Communications of the ACM*, 2020. Vol. 63(11). P. 139–144.