

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

ПАТРАМАНСЬКА Анастасія Василівна

**Програмний модуль класифікації архітектурних елементів
за зображенням / Software Module for Architectural Element
Classification**

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

Кваліфікаційна робота

Виконала студентка групи КН-41
А.В. Патраманська

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

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.

Завідувач кафедри
_____ **Н.В. Дзюбановська**

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 СТУДЕНТЦІ
ПАТРАМАНСЬКІЙ Анастасії Василівні**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

**Програмний модуль класифікації архітектурних елементів за зображенням
/ Software Module for Architectural Element Classification**

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 автоматизованої класифікації архітектурних елементів об'єктів культурної спадщини, визначити склад класів і чинники варіативності зображень, здійснити огляд методів класифікації зображень у комп'ютерному зорі
- Проаналізувати набір даних Architectural Heritage Elements Image64 Dataset, визначити склад класів і вимоги до передобробки зображень для навчання та інференсу
- Реалізувати два сценарії моделювання: базову CNN та модель із перенесенням навчання на основі VGG16, з порівнянням режимів замороженої бази та тонкого налаштування (fine-tuning)
- Провести експериментальне оцінювання якості моделей за метриками ассурасу, precision, recall, F1-score та виконати аналіз помилок за матрицею плутанини
- Розробити інтерфейс користувача для інференсу з підтримкою Top-k прогнозів і реалізувати механізм пояснюваності результатів на основі Grad-CAM

5. Перелік графічного матеріалу у роботі

- Узагальнена схема цифрової класифікації архітектурних елементів і факторів варіативності зображень
- Структурна схема архітектури програмного модуля

- Блок-схема алгоритму навчання моделі
- Матриця плутанини для фінальної моделі на тестовій вибірці

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	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами 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 01.04.2026 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.	

Студентка _____ А.В. Патраманська
підпис

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

АНОТАЦІЯ

Кваліфікаційна робота на тему «Програмний модуль класифікації архітектурних елементів за зображенням» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» виконана обсягом у 57 сторінок і містить 28 ілюстрацій, 2 таблиці, 3 додатки та 31 використане джерело.

Метою роботи є розроблення та експериментальне оцінювання програмного модуля багатокласової класифікації архітектурних елементів за зображеннями із застосуванням згорткових нейронних мереж і перенесення навчання, а також забезпечення інтерпретованості результатів за допомогою Grad-CAM.

Методами дослідження та розроблення обрано аналіз сучасних підходів комп'ютерного зору, проектування архітектури програмного модуля, експериментальне моделювання з порівнянням сценаріїв навчання (базова CNN, VGG16 із замороженою базою, VGG16 з тонким налаштуванням), а також оцінювання якості за метриками accuracy, precision, recall, F1-score і аналізом матриці плутанини.

У результаті виконання роботи реалізовано програмний модуль на Python із використанням TensorFlow/Keras та проведено порівняльні експерименти. Показано, що тонке налаштування VGG16 для входу 128×128 забезпечує найкращу точність на валідації 91,54% та приріст відносно базової CNN (81,45%) на +10,09 відсоткових пунктів. На тестовому наборі фінальна модель VGG16 fine-tune досягла accuracy 91,17% (loss 0,2793), macro-precision 0,89, macro-recall 0,91 та macro-F1 0,90, що свідчить про збалансовану якість розпізнавання; базова CNN демонструє нижчі показники (accuracy 68,09%, macro-F1 0,68). Аналіз Grad-CAM підтвердив фокус моделі на морфологічно значущих фрагментах архітектурних елементів, що підвищує інтерпретованість прогнозів і придатність модуля для прикладного використання.

Результати роботи можуть бути використані під час цифрової інвентаризації та каталогізації об'єктів культурної спадщини, а також як складова програмних засобів візуальної аналітики (інференс із поясненням рішення моделі).

Ключові слова: КОМП'ЮТЕРНИЙ ЗІР, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ПЕРЕНЕСЕННЯ НАВЧАННЯ, VGG16, БАГАТОКЛАСОВА КЛАСИФІКАЦІЯ, GRAD-CAM, ІНТЕРПРЕТОВНІСТЬ, КУЛЬТУРНА СПАДЩИНА.

ANNOTATION

Qualification work on the topic «Software Module for Architectural Element Classification» for the Bachelor's degree on speciality 122 «Computer Science», educational and professional program «Computer Science», is written on 57 pages and contains 28 figures, 2 tables, 3 annexes and 31 sources.

The purpose of the work is to develop and experimentally evaluate a software module for multi-class image-based classification of architectural elements using convolutional neural networks and transfer learning, and to ensure interpretability of predictions using Grad-CAM.

The research and development methods include analysis of modern computer vision approaches, software module architecture design, experimental modeling with comparison of training scenarios (baseline CNN, VGG16 with frozen base, VGG16 fine-tuning), and performance evaluation using accuracy, precision, recall, F1-score metrics and confusion matrix analysis.

As a result, a Python-based module using TensorFlow/Keras was implemented and comparative experiments were conducted. It was shown that VGG16 fine-tuning for 128×128 input provides the best validation accuracy of 91.54% and improves over the baseline CNN (81.45%) by +10.09 percentage points. On the test set, the final VGG16 fine-tune model achieved 91.17% accuracy (loss 0.2793) with macro-precision 0.89, macro-recall 0.91 and macro-F1 0.90, while the baseline CNN yields lower results (accuracy 68.09%, macro-F1 0.68). Grad-CAM analysis confirms that the model focuses on morphologically relevant image regions, increasing interpretability and supporting practical use of the module.

The results can be applied to digital inventory and cataloging of cultural heritage objects, as well as to visual analytics pipelines with explainable inference.

Keywords: COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORK, TRANSFER LEARNING, VGG16, MULTI-CLASS CLASSIFICATION, GRAD-CAM, INTERPRETABILITY, CULTURAL HERITAGE.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної області та огляд аналогів	11
1.1 Архітектурні елементи як об'єкт цифрової обробки	11
1.2 Аналіз набору даних	13
1.3 Огляд методів класифікації зображень	16
1.4 Постановка задачі.....	19
2 Алгоритмічне та інформаційне забезпечення модуля	23
2.1 Архітектура програмного модуля	23
2.2 Алгоритм навчання моделі	24
2.3 Інформаційне забезпечення.....	27
3 Програмно-технологічне забезпечення.....	30
3.1 Реалізація моделей та навчальних сценаріїв	30
3.2 Оцінювання якості, аналіз помилок та порівняння моделей	32
3.3 Інтерфейс користувача та пояснюваність Grad-CAM.....	36
Висновки	39
Список використаних джерел	42
Додаток А Програмний код	46
Додаток Б Тестування	48
Додаток В Апробація отриманих результатів.....	52

ВСТУП

Активна цифровізація культурної спадщини та зростання обсягів фотофіксації історичних споруд і їхніх фрагментів формують потребу в автоматизованих засобах аналізу візуальних даних. Для інвентаризації, каталогізації, пошуку та підготовки матеріалів до реставраційних робіт важливо мати формалізований опис не лише об'єкта в цілому, а й його морфологічно виокремлюваних складових – архітектурних елементів (куполи, дзвіниці, апсиди, аркбутани, вітражі тощо). Такі елементи виступають семантичними одиницями опису пам'яток і є основою для побудови довідкових систем, навчальних ресурсів і цифрових архівів.

Традиційне опрацювання зображень архітектурних елементів передбачає ручну експертну атрибуцію, що є трудомісткою, суб'єктивною та погано масштабується на великі колекції. Додатково задача ускладнюється варіативністю ракурсів, освітлення, наявністю оклюзій і фонових домішок, а також міжкласовою схожістю окремих елементів, що знижує відокремлюваність класів у просторі ознак. За таких умов потрібні методи, здатні автоматично формувати стійкі ієрархічні візуальні представлення та забезпечувати прийнятну точність розпізнавання в прикладних сценаріях.

Сучасні підходи комп'ютерного зору, насамперед згорткові нейронні мережі та перенесення навчання (transfer learning), демонструють високу ефективність у задачах багатокласової класифікації зображень за обмежених обсягів доменних даних [1–3]. Водночас практична цінність таких підходів у межах кваліфікаційної роботи визначається не лише показниками якості, а й відтворюваністю експериментів, наявністю завершеного програмного модуля та засобів пояснюваності результатів. Для задач культурної спадщини додатково актуальною є інтерпретованість прогнозів, зокрема через візуалізацію областей зображення, що найбільше вплинули на рішення моделі (наприклад, Grad-CAM) [4].

Отже, актуальним є розроблення програмного модуля багатокласової класифікації архітектурних елементів за зображенням на основі згорткових нейронних мереж із експериментальним обґрунтуванням вибору навчального сценарію (базова CNN та модель з перенесенням навчання), порівнянням якості та реалізацією інтерактивного інтерфейсу інференсу з можливістю пояснення прогнозів.

Мета роботи – розробити та експериментально оцінити програмний модуль автоматизованої класифікації архітектурних елементів об'єктів культурної спадщини за зображеннями із застосуванням згорткових нейронних мереж і перенесення навчання, а також забезпечити інтерпретованість результатів за допомогою Grad-CAM.

Для досягнення мети поставлено такі завдання:

1. Оцінити сучасний стан розв'язання задачі класифікації архітектурних елементів у комп'ютерному зорі та обґрунтувати вибір підходів глибинного навчання.
2. Проаналізувати набір даних архітектурних елементів, визначити склад класів і вимоги до передобробки зображень для навчання та інференсу.
3. Реалізувати два сценарії моделювання: базову CNN та модель із перенесенням навчання (VGG16) з порівнянням режимів «заморожена база» і тонкого налаштування (fine-tuning).
4. Провести експериментальне оцінювання якості моделей за метриками ассурасу, precision, recall, F1-score та виконати аналіз помилок за матрицею плутанини.
5. Розробити інтерфейс користувача для інференсу з підтримкою Top-k прогнозів та реалізувати механізм пояснюваності на основі Grad-CAM для обраної моделі.
6. Узагальнити результати, визначити практичні сценарії застосування модуля та сформулювати рекомендації щодо його використання у задачах цифрової каталогізації.

Об'єкт дослідження – процес автоматизованої класифікації архітектурних елементів об'єктів культурної спадщини за зображеннями.

Предмет дослідження – методи та програмні засоби багатокласової класифікації зображень архітектурних елементів із використанням згорткових нейронних мереж, перенесення навчання та візуальних методів пояснення прогнозів (Grad-CAM).

Методи дослідження: методи комп'ютерного зору та глибинного навчання (CNN, transfer learning, fine-tuning), методи передобробки й аугментації зображень, експериментальне оцінювання якості класифікації (accuracy, precision, recall, F1-score), аналіз помилок на основі матриці плутанини, метод пояснюваності Grad-CAM.

Практичне значення роботи полягає у створенні відтворюваного програмного модуля класифікації архітектурних елементів з інтерактивним інтерфейсом інференсу та засобами пояснюваності, придатного для використання в інформаційних системах цифрових архівів, навчальних ресурсах і задачах підтримки процесів інвентаризації та каталогізації об'єктів культурної спадщини.

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 ОГЛЯД АНАЛОГІВ

1.1 Архітектурні елементи як об'єкт цифрової обробки

Архітектурні елементи – це морфологічно виокремлювані складові архітектурного об'єкта (фасаду, інтер'єру або конструктивного вузла), що виконують конструктивну, функціональну або декоративну роль і можуть бути ідентифіковані за характерними візуальними ознаками. До таких елементів належать, зокрема, колони та пілястри, арки й склепіння, карнизи, капітелі, фронтони, балюстради, портали, наличники, віконні та дверні обрамлення, розетки, фризіві пояси, а також інші фасадні декоративні деталі. У контексті комп'ютерного зору ці об'єкти розглядаються як категорії багатокласової класифікації, де кожному зображенню або фрагменту зображення ставиться у відповідність мітка класу, яка відображає тип елемента. Цифрова обробка архітектурних елементів є актуальною для задач інвентаризації та каталогізації об'єктів культурної спадщини, підтримки реставраційних робіт, створення навчальних та довідкових систем, а також для автоматизованого пошуку подібних фрагментів у фотоколекціях і цифрових архівах.

Специфіка архітектурних елементів як об'єкта аналізу полягає у поєднанні локальних і глобальних ознак. Локальні ознаки відображають дрібні деталі форми та текстури (профіль карниза, рисунок різьблення, рельєфність каменю), тоді як глобальні ознаки пов'язані з просторовою конфігурацією елемента (кривизна арки, пропорції колони, структура обрамлення). Для моделей глибинного навчання така багаторівнева природа ознак створює вимогу до достатньої узагальнювальної здатності: модель має бути стійкою до змін масштабу, орієнтації та часткових спотворень, але водночас чутливою до дрібних морфологічних відмінностей між близькими класами.

Класифікація архітектурних елементів за зображенням ускладнюється низкою факторів, що зумовлюють варіативність вхідних даних та знижують відокремлюваність класів у просторі ознак. По-перше, значний вплив має ракурс зйомки та перспектива. Один і той самий елемент, зафіксований фронтально та

під кутом, може демонструвати різні контурні характеристики, зміну пропорцій і часткове зникнення симетричних фрагментів. Для арок і склепінь ракурсні спотворення особливо критичні, оскільки змінюють геометрію дуги та співвідношення опорних частин, що ускладнює стабільне виділення геометричних індикаторів класу. По-друге, суттєвим є вплив умов освітлення. Пряме сонячне світло, контрове освітлення або штучне підсвічування змінюють розподіл яскравості та контрасту, формують різкі тіні й засвіти, через що частина текстурних ознак може бути втрачена або спотворена. За таких умов зростає ризик того, що модель фіксуватиметься на нестійких фотометричних патернах, які не є семантично значущими для класу.

По-третє, у реальних зображеннях часто наявні часткові перекриття (оклюзії) та фонові домішки: гілки дерев, дроти, елементи міського середовища, люди або транспорт можуть перекривати фрагменти архітектурного елемента. Оклюзії призводять до неповноти спостережуваних ознак і зменшують інформативність зображення, що особливо проблематично для елементів, де клас визначається сукупністю деталей (наприклад, капітель із характерним орнаментом або багатокomпонентне обрамлення). По-четверте, характерною є міжкласова схожість, коли різні категорії мають подібні візуальні риси або спільні морфологічні компоненти. Наприклад, окремі типи карнизів можуть бути візуально близькими до фризів через схожий профіль та горизонтальну організацію, а частина обрамлень може містити аркові мотиви, що наближує їх до класу арок у випадку кадрування фрагмента. Така схожість створює умови для систематичних помилок, коли модель плутає класи не випадково, а за наявності спільного візуального “ядра”.

По-п'яте, якість фотоматеріалів є неоднорідною: зустрічаються зображення з низькою роздільною здатністю, стисненням із втратами, розмиттям, шумом, артефактами масштабування або некоректним балансом білого. Для наборів даних із фіксованим малим розміром зображень (наприклад, 64×64) додатково актуалізується проблема втрати дрібних деталей, які можуть бути ключовими для відмінності класів. У таких умовах модель може демонструвати

високу чутливість до якості вхідного сигналу, а отже, потребує процедур підвищення стійкості – зокрема нормалізації, аугментацій та використання перенесення навчання (transfer learning), що дозволяє використати узагальнені візуальні ознаки, сформовані на великих корпусах зображень.

З огляду на наведені особливості, архітектурні елементи як об'єкт цифрової обробки потребують формалізованого підходу до підготовки даних і побудови класифікаційної моделі: визначення стабільних класових індикаторів, забезпечення репрезентативності вибірки за умовами зйомки, зменшення впливу фотометричних і геометричних варіацій та аналізу типових помилок із метою подальшого вдосконалення пайплайна. Загальну логіку задачі та чинники, що впливають на якість розпізнавання, узагальнено на рисунку 1.1.

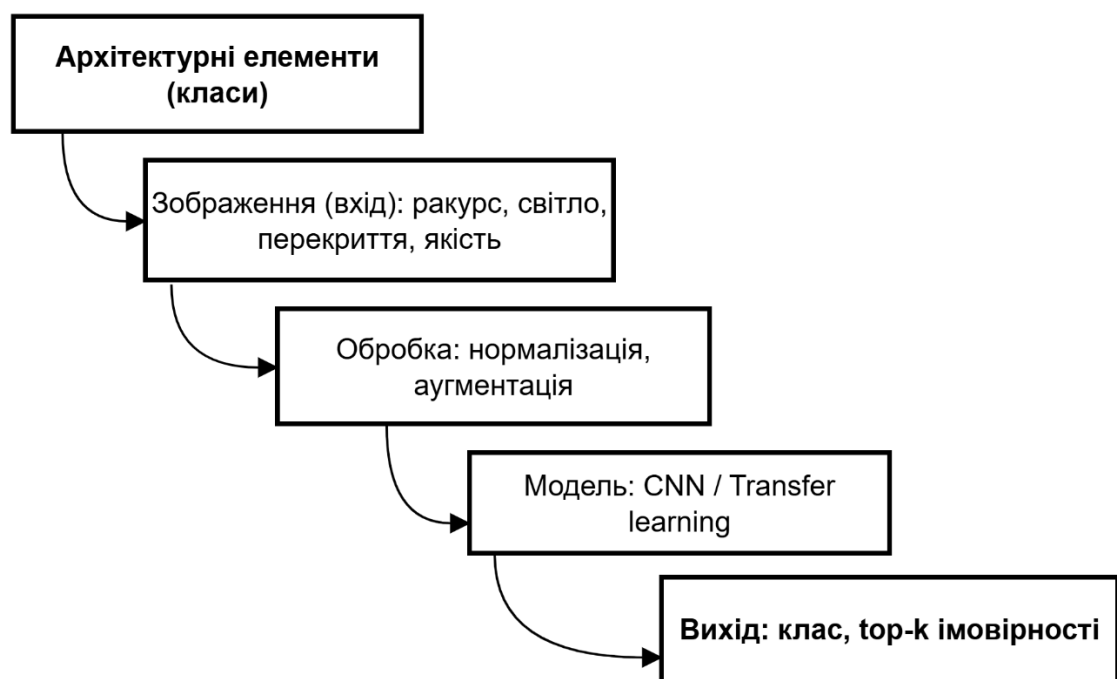


Рисунок 1.1 – Узагальнена схема цифрової класифікації архітектурних елементів і факторів варіативності зображень

1.2 Аналіз набору даних

Для реалізації програмного модуля класифікації архітектурних елементів обрано набір даних Architectural Heritage Elements Image64 Dataset, який

використано у відповідному Kaggle notebook. Набір даних є доцільним для задачі багатокласової класифікації, оскільки містить 10 категорій архітектурних елементів та уніфікований формат зображень 64×64 пікселі, а загальний обсяг становить близько 10 тис. зображень. Така конфігурація забезпечує компроміс між достатньою репрезентативністю вибірки та обчислювальною доступністю експериментів, що є важливим для відтворюваного навчання моделей у межах кваліфікаційної роботи.

Структурно дані організовані за принципом клас-орієнтованої розмітки: кожне зображення належить рівно одному класу, який відповідає конкретному типу елемента. До множини класів входять, зокрема: `vault`, `stained_glass`, `apse`, `dome(inner)`, `dome(outer)`, `column`, `bell_tower`, `flying_buttress`, `gargoyle`, `altar`. Наведений перелік охоплює як фасадні, так і інтер'єрні компоненти сакральної та історичної архітектури, що підвищує варіативність візуальних патернів і наближає умови навчання моделі до практичних сценаріїв застосування.

Візуальна різноманітність вибірки проілюстрована на рисунку 1.2, де наведено приклади зображень з навчального підмножини. На прикладах чітко простежуються типові джерела внутрішньокласової варіативності: відмінності ракурсу та масштабу (наприклад, для `bell_tower` і `column`), суттєва залежність від умов освітлення та контрасту (особливо для `stained_glass` і `vault`), а також вплив фону та композиції кадру (наприклад, `flying_buttress` і `apse`, де елемент часто займає лише частину зображення). Додатково, окремі класи характеризуються міжкласовою схожістю на рівні геометричних ознак: `dome(inner)` та `dome(outer)` можуть мати подібні криволінійні контури, тоді як декоративні елементи на кшталт `gargoyle` у ряді випадків ускладнюються фактурністю матеріалу та частковими перекриттями, що підвищує ризик помилок класифікації.

Оскільки розмір зображень у наборі даних є фіксованим (64×64), критичним стає питання збереження діагностичних ознак класу при зменшеній роздільній здатності: дрібні деталі різьблення, контури арок чи елементи орнаменту можуть частково втрачатися внаслідок ресайзінгу та стиснення. У зв'язку з цим під час підготовки даних доцільно застосовувати нормалізацію

значень пікселів, стандартизований підхід до перетворення розміру (якщо використовується transfer learning з іншими вхідними розмірами), а також аугментації, спрямовані на підвищення стійкості моделі до зміни умов зйомки (невеликі повороти, зсуви, масштабування, варіації яскравості/контрасту). Такі перетворення мають бути контрольованими, щоб не руйнувати семантику елемента та не формувати штучних патернів, які відсутні у реальних даних.

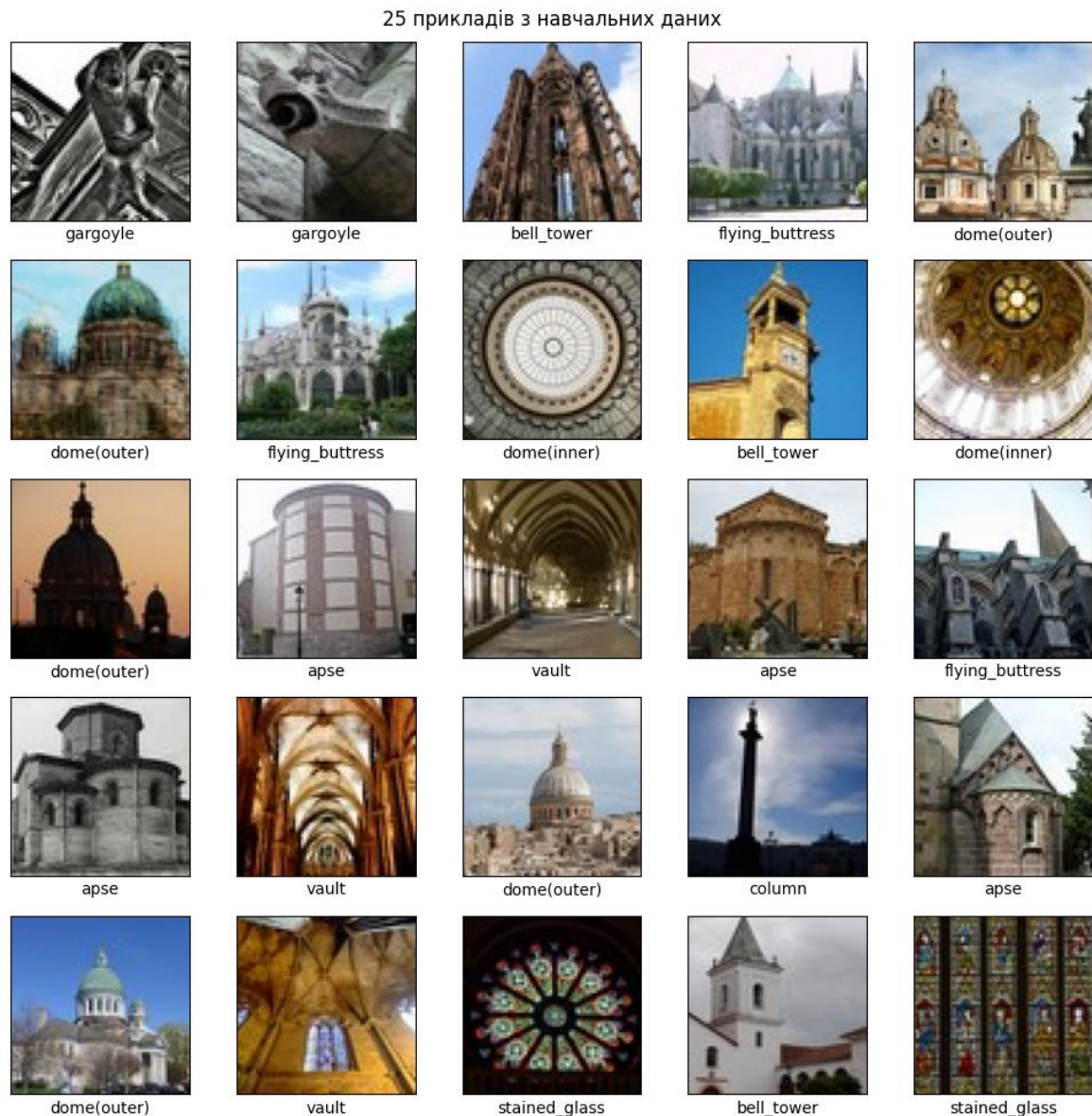


Рисунок 1.2 – Приклади зображень з навчальної вибірки (25 зразків) для класів архітектурних елементів

Окремої уваги потребує аналіз розподілу зображень за класами. Навіть за загального обсягу близько 10 тис. прикладів можливий дисбаланс, що

призводить до зміщення моделі у бік домінантних категорій. Тому в роботі доцільно: (1) подати кількісну характеристику класів у вигляді таблиці (таблиця 1.1), (2) використовувати стратифікований поділ на навчальну, валідаційну та тестову вибірки, (3) оцінювати результати не лише за accuracy, а й за macro-F1 та матрицею помилок, що дозволяє коректніше відобразити якість розпізнавання для кожного класу.

Таблиця 1.1 – Характеристики набору даних Architectural Heritage Elements Image64 Dataset (класи, кількість зображень, частка у вибірці)

Параметр	Значення
Кількість зображень (навчальна вибірка)	10 130
Кількість зображень (тестова вибірка)	1 404
Загальна кількість зображень	11 534
Кількість класів	10
Перелік класів	altar; apse; bell_tower; column; dome(inner); dome(outer); flying_buttress; gargoyle; stained_glass; vault

Таким чином, обраний набір даних забезпечує необхідну основу для навчання моделі класифікації архітектурних елементів, водночас вимагаючи формалізованих процедур передобробки та контролю дисбалансу. У наступному підрозділі буде розглянуто методи класифікації зображень і обґрунтовано вибір архітектури моделі для реалізації програмного модуля.

1.3 Огляд методів класифікації зображень

Класифікація зображень у задачах комп'ютерного зору історично розвивалася від ручного конструювання ознак до автоматичного навчання багаторівневих представлень у глибоких нейронних мережах. У класичній постановці обробка зображення зводиться до виділення дескрипторів (контурних, текстурних, локальних) і подальшої побудови класифікатора (SVM,

логістична регресія, випадковий ліс тощо), що приймає рішення в просторі ознак [1–6]. Перевагою таких підходів є відносна інтерпретованість та помірні обчислювальні витрати, однак їхня якість суттєво залежить від вибору й стабільності ознак щодо ракурсів, освітлення та шумів, що є критичним саме для архітектурних елементів.

До найпоширеніших класичних дескрипторів належать SIFT/SURF, HOG, локальні бінарні шаблони (LBP), а також підходи на основі “візуального словника” (Bag-of-Visual-Words) [2–6]. У межах таких рішень типова схема включає нормалізацію, виділення ключових точок або градієнтних орієнтацій, агрегацію ознак і навчання лінійного/ядрового класифікатора [4–7]. Для невеликих датасетів це може бути практичним, але при зростанні внутрішньокласової варіативності (різні масштаби елемента, часткові перекриття, складний фон) ручні ознаки часто не забезпечують достатньої узагальнювальної здатності [6–8].

Переломним етапом у класифікації зображень стали згорткові нейронні мережі (CNN), які навчають ієрархію ознак безпосередньо з даних: від простих фільтрів (краї, текстури) до складних композиційних патернів (частини об’єктів і їхні конфігурації) [9–11]. Порівняно з класичними підходами CNN краще адаптуються до реальних викривлень зображень, оскільки інваріантність до зсувів/масштабів частково закладена у згортках і пулінгу, а регуляризація (dropout, batch normalization) підвищує стійкість моделі [12–14]. Водночас CNN потребують більших обсягів даних і ретельної настройки процесу навчання (оптимізатор, швидкість навчання, рання зупинка), аби уникати перенавчання [13–16].

У практичних системах широко застосовують transfer learning – перенесення знань із попередньо навченої моделі (зазвичай на ImageNet) до цільової задачі з обмеженими даними [17–19]. Для задачі класифікації архітектурних елементів це особливо релевантно, оскільки набір даних має фіксовані класи й відносно помірний обсяг, а також високу варіативність умов зйомки; тому попередньо вивчені універсальні візуальні ознаки дають змогу

підвищити якість і стабільність прогнозування [18–20]. Типовими стратегіями є: (1) використання backbone як “екстрактора ознак” із навчанням лише класифікаційної “голови”; (2) fine-tuning – часткове або повне донавчання верхніх шарів із малою швидкістю навчання [18–21]. Паралельно застосовують контрольовані аугментації, що імітують реальні спотворення (повороти, зсуви, зміни яскравості/контрасту), зменшуючи чутливість моделі до умов зйомки [22–23].

Серед типових архітектур CNN, які застосовують у переносі навчання, доцільно виокремити VGG, ResNet та EfficientNet як репрезентативні класи моделей із різним компромісом “якість–ресурси”. Архітектура VGG відзначається простотою (послідовні 3×3 згортки) і зручністю як базова точка порівняння, але є параметрично “важкою” [24]. ResNet вводить залишкові з’єднання, що стабілізують градієнти та дозволяють ефективно навчати глибші мережі; на практиці це часто дає кращу узагальнюваність при зіставних або менших ризиках перенавчання [25]. EfficientNet пропонує масштабування глибини/ширини/роздільної здатності за принципом compound scaling і нерідко демонструє високу якість при помірних обчислювальних витратах, що важливо для модульних прикладних рішень [26]. Для мобільних або ресурсно обмежених сценаріїв додатково використовують MobileNet-подібні сімейства, що оптимізують обчислення за рахунок depthwise separable convolutions [27].

Вибір підходу в межах цієї роботи обґрунтовується вимогами (таблиця 1.2): (а) забезпечити прийнятну якість на 10-класовій класифікації; (б) зберегти відтворюваність та керованість експерименту; (в) реалізувати інференс у межах програмного модуля. Тому як базову лінію доцільно розглядати CNN (власну або просту архітектуру), а для підвищення якості – transfer learning із однією з типових моделей (ResNet/EfficientNet) з подальшим fine-tuning, що узгоджується з сучасною практикою побудови прикладних класифікаторів зображень [18–21, 25–27]. Оцінювання якості має враховувати можливий дисбаланс класів, тому поряд із accuracy слід застосовувати macro-F1 та матрицю помилок, що забезпечує більш коректне порівняння альтернатив [28–30].

Таблиця 1.2 – Порівняння підходів класифікації зображень (переваги/обмеження/умови застосування)

Підхід	Переваги	Обмеження	Умови доцільного застосування
Ручні ознаки + класичний класифікатор (HOG/ LBP/ SIFT + SVM/ RF) [2–7]	Менші вимоги до ресурсів; часткова інтерпретованість; швидке навчання на малих даних	Чутливість до ракурсів/освітлення; складність узагальнення; залежність від підбору ознак	Невеликі датасети, контрольовані умови зйомки, потреба у швидкому прототипі
CNN “з нуля” [9–16]	Автоматичне виділення ознак; висока якість на складних даних; гнучкість архітектур	Потреба в даних; ризик перенавчання; вимоги до налаштування навчання	Середні/великі датасети або агресивна аугментація; наявність достатніх ресурсів
Transfer learning (VGG/ ResNet/ EfficientNet) + fine-tuning [18–21, 24–26]	Вища якість на обмежених даних; швидша збіжність; стабільніші ознаки	Потреба у коректній стратегії донавчання; залежність від вхідного розміру/попереднього домену	Обмежений обсяг даних, потреба в кращій узагальнюваності та практичній придатності модуля

Таким чином, для задачі 10-класової класифікації архітектурних елементів найбільш обґрунтованим є використання згорткових нейронних мереж, а для підвищення якості на обмеженому наборі даних — підхід перенесення навчання з подальшим fine-tuning типових архітектур (ResNet/EfficientNet). Оцінювання результатів доцільно здійснювати не лише за accuracy, а й за macro-F1 та матрицею помилок, що забезпечує коректніший аналіз якості для всіх класів за можливого дисбалансу.

1.4 Постановка задачі

Актуальність розроблення програмного модуля класифікації архітектурних елементів зумовлена поєднанням прикладових потреб цифровізації культурної спадщини та методичних викликів автоматизованого аналізу зображень у реальних умовах. Як показано в підрозділі 1.1, архітектурні елементи є морфологічно виокремлюваними складовими об’єктів архітектури, що

виконують конструктивні, функціональні або декоративні ролі та можуть слугувати семантичними «одиницями опису» під час інвентаризації і каталогізації пам'яток. Саме рівень елементів (колона, купол, апсида, аркбутан, вітраж тощо) є практично значущим для пошуку подібних фрагментів у цифрових архівах, підтримки реставраційних робіт, створення довідкових і навчальних систем. Водночас ручна атрибуція таких фрагментів експертами не масштабується на великі обсяги фотоматеріалів, є трудомісткою та залежить від суб'єктивних інтерпретацій, що ускладнює забезпечення узгодженості описів у колекціях.

Додаткову актуальність формує складність самої задачі розпізнавання архітектурних елементів у візуальних даних. Відповідно до підрозділу 1.1, ці об'єкти характеризуються поєднанням локальних і глобальних ознак: з одного боку, діагностичними є дрібні деталі профілю, різьблення, фактури матеріалу; з іншого – просторові конфігурації, пропорції та геометричні співвідношення. Така багаторівнева природа ознак вимагає від моделі одночасно інваріантності до змін масштабу, орієнтації та часткових спотворень і, водночас, чутливості до малих морфологічних відмінностей між близькими класами. Ситуацію ускладнюють типові для польових знімків чинники: перспективні спотворення та ракурсна варіативність, неоднорідне освітлення із тінями та засвітами, оклюзії (дерева, дроти, люди, транспорт), фонові домішки й композиційні зсуви. Окремою проблемою є міжкласова схожість, коли різні елементи мають спільне «візуальне ядро» (наприклад, криволінійні контури або повторювані декоративні мотиви), що створює умови для систематичних плутанин, а не випадкових помилок. Таким чином, без спеціалізованого підходу до моделювання та контролю якості отримані результати можуть бути нестійкими до зміни умов зйомки, а отже – малопридатними для практичної експлуатації.

Як обґрунтовано в підрозділі 1.2, актуальність задачі підсилюється характеристиками обраного набору даних: зображення мають фіксовану невелику роздільну здатність (64×64), а класи охоплюють різнорідні архітектурні елементи, для яких дрібні деталі часто є визначальними. У таких умовах зростає

ризик втрати діагностичних ознак через ресайзинг, стиснення та шум, а якість моделі суттєво залежить від формалізованої передобробки, контрольованих аугментацій і коректної організації вибірок (стратифікований поділ, аналіз розподілу за класами). Крім того, навіть за загального обсягу близько 10 тис. зображень можливий дисбаланс класів, що робить критично важливим застосування метрик, чутливих до якості по кожному класу (масо-усереднення, матриця плутанини), а не лише загальної асигасу. Отже, актуальним є не тільки навчання моделі, а й побудова відтворюваного експериментального контуру «дані–модель–оцінювання–аналіз помилок», який забезпечує методично коректну перевірку результатів.

Підрозділ 1.3 додатково підтверджує актуальність вибору сучасних методів глибинного навчання. Класичні підходи на основі ручних дескрипторів (SIFT/SURF, HOG, LBP) та традиційних класифікаторів можуть бути обчислювально доступними, проте вони є чутливими до ракурсів, освітлення і шумів, а також мають обмежену здатність узагальнювати складні візуальні патерни, характерні для архітектурних елементів. Натомість згорткові нейронні мережі формують ієрархію ознак безпосередньо з даних, що підвищує стійкість до типових викривлень, а перенесення навчання дає можливість ефективно використати попередньо вивчені універсальні візуальні представлення для задач із помірним обсягом доменних даних. Водночас практичні рішення повинні враховувати компроміс «якість–ресурси», забезпечувати відтворюваність і бути придатними до інтеграції, тобто реалізовувати не лише модель, а повний програмний модуль з інференсом, top-k результатами та засобами інтерпретації.

Отже, актуальність роботи визначається тим, що класифікація архітектурних елементів є методично складною та прикладово значущою задачею цифрової спадщини, яка потребує: (1) стійких моделей до варіативності реальних зображень; (2) формалізованої підготовки даних з урахуванням низької роздільної здатності та можливого дисбалансу класів; (3) відтворюваного програмного модуля, придатного для практичного використання і подальшого розширення. Саме тому розроблення та експериментальна оцінка модуля на

основі CNN і transfer learning із порівнянням сценаріїв навчання та аналізом помилок є своєчасним і обґрунтованим напрямом дослідження.

Мета роботи – розробити та експериментально оцінити програмний модуль автоматизованої класифікації архітектурних елементів об’єктів культурної спадщини за зображеннями із застосуванням згорткових нейронних мереж і перенесення навчання, а також забезпечити інтерпретованість результатів за допомогою Grad-CAM.

Для досягнення мети поставлено такі завдання:

1. Оцінити сучасний стан розв’язання задачі класифікації архітектурних елементів у комп’ютерному зорі та обґрунтувати вибір підходів глибинного навчання.
2. Проаналізувати набір даних архітектурних елементів, визначити склад класів і вимоги до передобробки зображень для навчання та інференсу.
3. Реалізувати два сценарії моделювання: базову CNN та модель із перенесенням навчання (VGG16) з порівнянням режимів «заморожена база» і тонкого налаштування (fine-tuning).
4. Провести експериментальне оцінювання якості моделей за метриками accuracy, precision, recall, F1-score та виконати аналіз помилок за матрицею плутанини.
5. Розробити інтерфейс користувача для інференсу з підтримкою Top-k прогнозів та реалізувати механізм пояснюваності на основі Grad-CAM для обраної моделі.
6. Узагальнити результати, визначити практичні сценарії застосування модуля та сформулювати рекомендації щодо його використання у задачах цифрової каталогізації.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

2.1 Архітектура програмного модуля

Архітектура модуля побудована як послідовність взаємопов'язаних підсистем, кожна з яких відповідає за окремий етап обробки даних та експлуатації моделі. На рисунку 2.1 подано узагальнену архітектуру та потоки даних, що відображають перехід від сирих зображень і міток до прогнозів у прикладному інтерфейсі. Початковою ланкою є підсистема завантаження даних (Data ingestion), яка забезпечує читання набору даних та формування підвбірок (train/val/test) із дотриманням відтворюваності розбиття. Далі дані передаються до підсистеми передобробки та аугментації (Preprocess + Augment), що реалізує нормалізацію, приведення до потрібного формату та контрольовані перетворення зображень для підвищення стійкості моделі до зміни умов зйомки.

Наступним компонентом є підсистема навчання моделі (Training), в якій здійснюється ініціалізація архітектури (власна CNN або попередньо навчений backbone) та оптимізація параметрів на навчальній вибірці. Результати навчання передаються до підсистеми оцінювання та звітності (Evaluation + Reports), яка формує кількісні показники якості та діагностичні матеріали (метрики, матрицю помилок, криві навчання). Окремо, за результатами навчання і валідації здійснюється збереження моделі у сховище артефактів (Model storage), що дозволяє відокремити етапи тренування від етапу експлуатації й гарантувати повторюваність інференсу за фіксованими вагами та конфігураціями.

Кінцевим етапом у архітектурі є інференс і керування артефактами (Inference + Artifacts) та інтерфейс користувача (UI/API). Інференс забезпечує завантаження моделі, обробку нового зображення за тим самим правилом передобробки та отримання прогнозу. Інтерфейс користувача надає доступ до результатів (клас, top-k, ймовірності) та, за потреби, до сформованих звітів і метрик, що особливо важливо для демонстрації коректності та якості модуля. Таким чином, архітектура на рисунку 2.1 забезпечує модульність, розділення

функцій і можливість розширення рішення (наприклад, заміна backbone або розширення системи звітності) без порушення загальної логіки потоків даних.

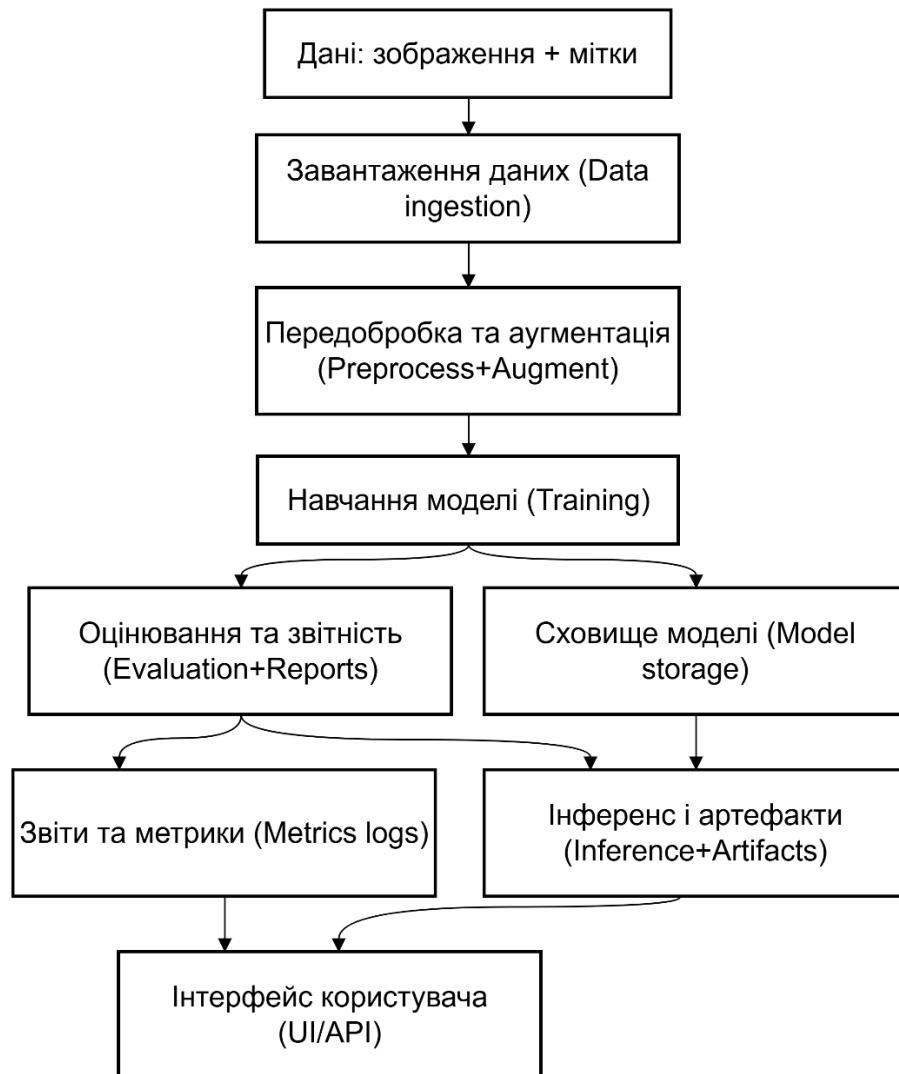


Рисунок 2.1 – Архітектура модулів та потоки даних

2.2 Алгоритм навчання моделі

Алгоритмічне забезпечення модуля охоплює процедури підготовки даних, навчання, оцінювання та подальшого інференсу. Узагальнений алгоритм представлено на рисунку 2.2, де відображено ключові рішення щодо ініціалізації моделі та контролю навчання. На першому етапі виконується підготовка даних, що включає формування підвбірок (train/val/test) та застосування стандартизованих перетворень, які повинні бути однаковими для навчання і для

інференсу. Це знижує ризик зміщення розподілу (train–serve skew) і забезпечує коректність прогнозування в прикладному режимі.

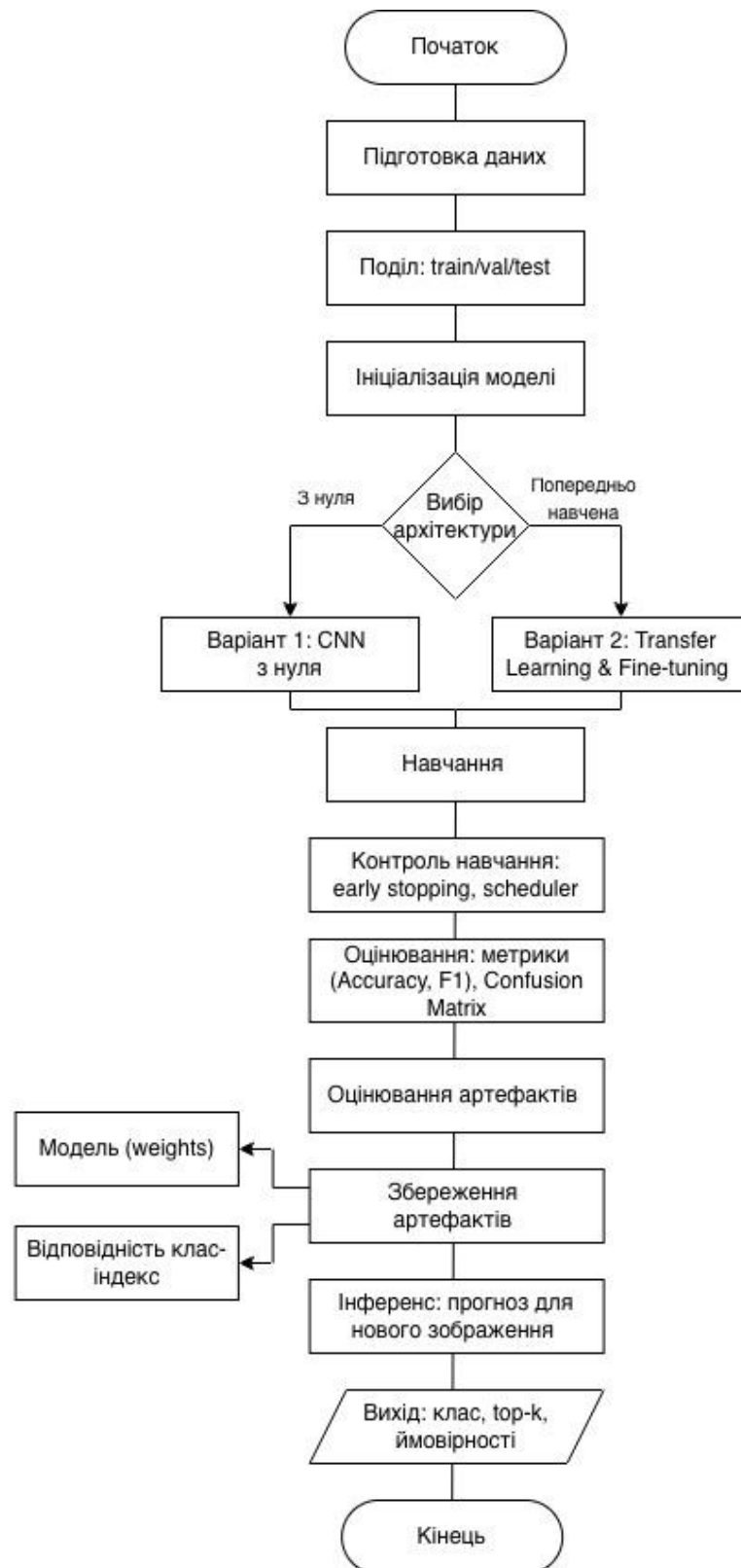


Рисунок 2.2 – Блок-схема пайплайна навчання та інференсу

Далі відбувається ініціалізація моделі, для якої передбачено два базових сценарії: навчання CNN з нуля або використання попередньо навченої базової мережі (pretrained backbone). Використання transfer learning є доцільним у ситуаціях, коли обсяг даних є обмеженим або спостерігається висока варіативність зображень: попередньо навчена модель містить універсальні низько- та середньорівневі ознаки (контури, текстури, базові форми), що сприяють швидшій збіжності та кращій узагальнюваності. У межах transfer learning типовою є стратегія початкового навчання класифікаційної “голови” при заморожених базових шарах, після чого застосовується fine-tuning — часткове розморожування верхніх шарів backbone та донавчання з меншою швидкістю навчання. Саме такий підхід (зображений на рисунку 2.2) дозволяє адаптувати узагальнені ознаки до домену архітектурних елементів, не втрачаючи стабільності представлень.

Етап навчання супроводжується механізмами контролю збіжності та запобігання перенавчанню: використовується early stopping (зупинка за відсутності покращення на валідації) та планувальник швидкості навчання (scheduler), що коригує крок оптимізації у процесі тренування. Після завершення навчання виконують оцінювання якості. Базовою метрикою виступає асигасу, однак для багатокласової задачі за потенційного дисбалансу класів більш інформативною є macro-F1, оскільки вона усереднює внесок кожного класу рівномірно і не дозволяє домінантним класам “маскувати” низьку якість на рідкісних категоріях. Додатково формується матриця помилок (confusion matrix), яка дає змогу ідентифікувати систематичні плутанини між класами та локалізувати слабкі місця моделі.

Завершальним кроком є збереження артефактів: ваг моделі та відповідності “клас–індекс” (label map). Це критично для відтворюваності інференсу, оскільки прогнозований індекс має однозначно інтерпретуватися як конкретний клас архітектурного елемента. Після збереження артефактів модуль переходить у режим інференсу, де для нового зображення обчислюються ймовірності класів і

повертається прогноз у вигляді класу та top-k результатів, як показано на рисунку 2.2.

2.3 Інформаційне забезпечення

Інформаційне забезпечення модуля визначає склад вхідних і вихідних даних, а також структуру артефактів експериментів, необхідних для збереження результатів і подальшої перевірки. На рисунку 2.3 узагальнено інтерфейс даних (I/O) у контексті прогнозування: на вході подаються RGB-зображення у форматі 64×64 (або масштабовані до необхідного розміру відповідно до вимог обраного backbone) та, для задач навчання/оцінювання, мітка класу. Вихідними даними є прогнозований клас, вектор ймовірностей, топ-k класів за спаданням ймовірності та протокол оцінювання/тестування. Такий набір виходів забезпечує не лише функціональність класифікації, а й можливість прозорого аналізу якості та діагностики помилок.

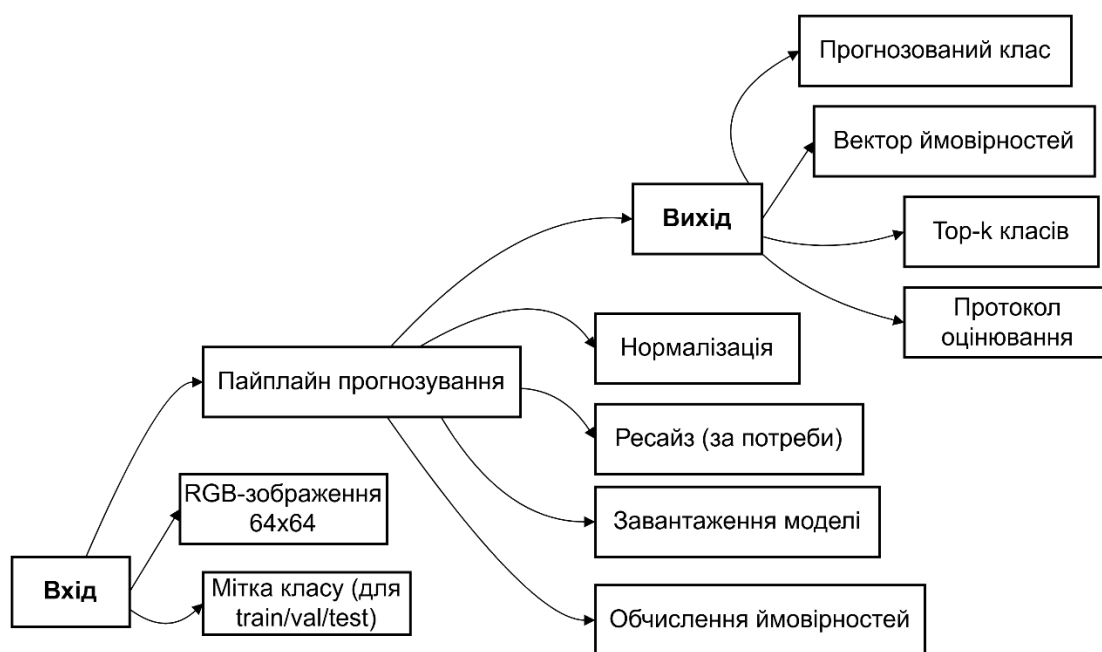


Рисунок 2.3 – Вхідні та вихідні дані (I/O схема)

Для забезпечення відтворюваності експериментів і коректного аудиту результатів необхідно формалізувати основні сутності та їхні зв'язки. На рисунку

2.4 подано концептуальну інфологічну модель експерименту, яка включає сутності Dataset, Image, Label, ExperimentRun, Metrics та ModelArtifact. Зв'язок Image→Label визначає приналежність кожного зображення до конкретного класу; ExperimentRun→Metrics фіксує метрики, отримані в результаті певного запуску навчання/оцінювання; ExperimentRun→ModelArtifact пов'язує конкретний експеримент із збереженими вагами та допоміжними артефактами. Додатково Dataset→ExperimentRun відображає факт використання певного набору даних у відповідному запуску, що дозволяє уникнути неоднозначностей при порівнянні різних експериментів.

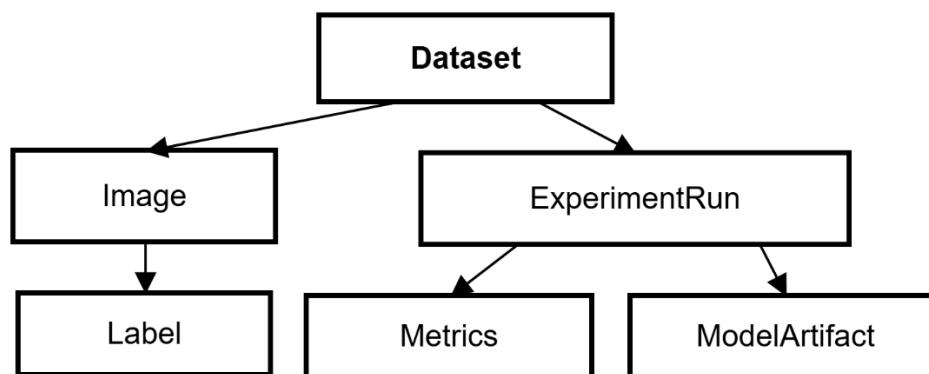


Рисунок 2.4 – Концептуальна інфологічна модель експерименту (сутності та зв'язки)

У межах бакалаврської роботи доцільним є варіант організації зберігання артефактів без використання бази даних, з чітко визначеною файловою структурою. На рисунку 2.5 наведено рекомендовану структуру каталогів, яка включає дані (train/val/test), збережені моделі (best_model, checkpoints, label_map), логи навчання та оцінювання, звіти (таблиця метрик, матриця помилок, криві навчання) та конфігурації запуску. Така організація є достатньою для повного відтворення експериментів, оскільки забезпечує збереження як параметрів моделі, так і контексту їх отримання (налаштування, журнали, результати).

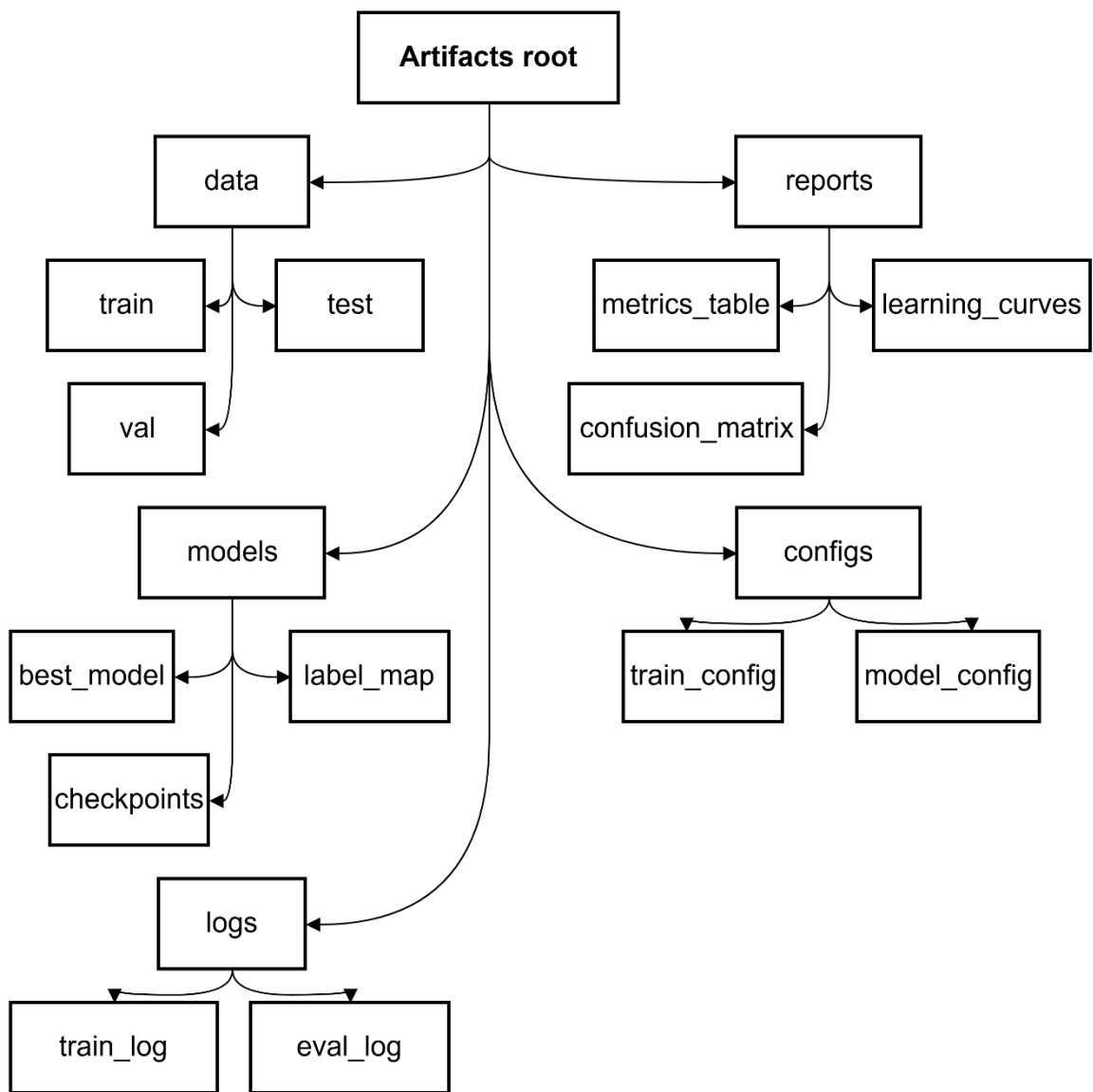


Рисунок 2.5 – Організація зберігання артефактів експериментів (Варіант А, без БД)

Отже, запропоноване інформаційне забезпечення формує цілісний контур “дані → навчання → оцінювання → артефакти → інференс”, що є необхідним для практичної реалізації програмного модуля та для методично коректного представлення результатів у кваліфікаційній роботі. У наступному розділі буде описано програмно-технологічну реалізацію модулів, інтерфейс користувача та результати експериментальної перевірки якості.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Реалізація моделей та навчальних сценаріїв

Програмний модуль реалізовано мовою Python із використанням TensorFlow/Keras. Навчання організовано через генератори батчів із нормалізацією піксельних значень та аугментаціями, що дозволяє підвищити варіативність навчальних прикладів і знизити ризик переобучення. Моделювання виконано у трьох основних сценаріях:

- базова CNN,
- перенесення навчання VGG16 із замороженою базою,
- тонке налаштування VGG16 (fine-tuning) з малою швидкістю навчання.

Базову CNN побудовано як послідовну згорткову архітектуру з трьома згортковими блоками та класифікаційною частиною (Flatten–Dense–Softmax), із регуляризацією на основі Dropout та Batch Normalization. Динаміку навчання базової CNN відображено на рисунку 3.1: найкраща точність на валідації становить 81,45%, при цьому криві точності/втрат демонструють збіжність із помірними коливаннями, що характерно для навчання з аугментацією.

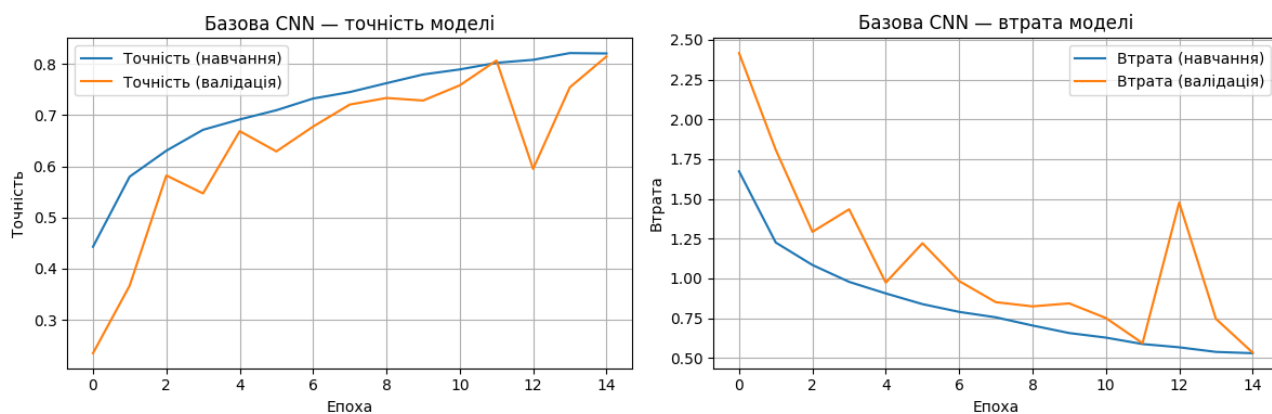


Рисунок 3.1 – Базова CNN: криві точності та втрат на навчанні і валідації

Для оцінки впливу перенесення навчання реалізовано модель на основі VGG16 у режимі замороженої згорткової бази. Для входу 64×64 динаміка навчання наведена на рисунку 3.2, де найкраща точність на валідації становить 68,10%; для входу 128×128 аналогічний режим показано на рисунку 3.3 із

найкращою валідаційною точністю 61,52%. Отримані результати свідчать, що без доменної адаптації (fine-tuning) перенесені ознаки VGG16 не забезпечують достатньої узгодженості з цільовою задачею.

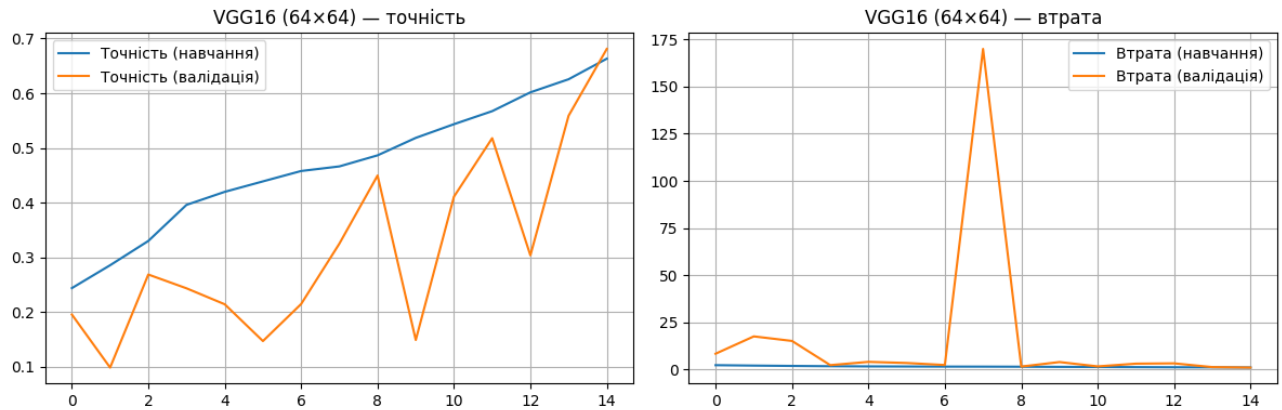


Рисунок 3.2 – VGG16 (64×64, заморожена база): криві точності та втрат;
найкраща точність на валідації 68,10%

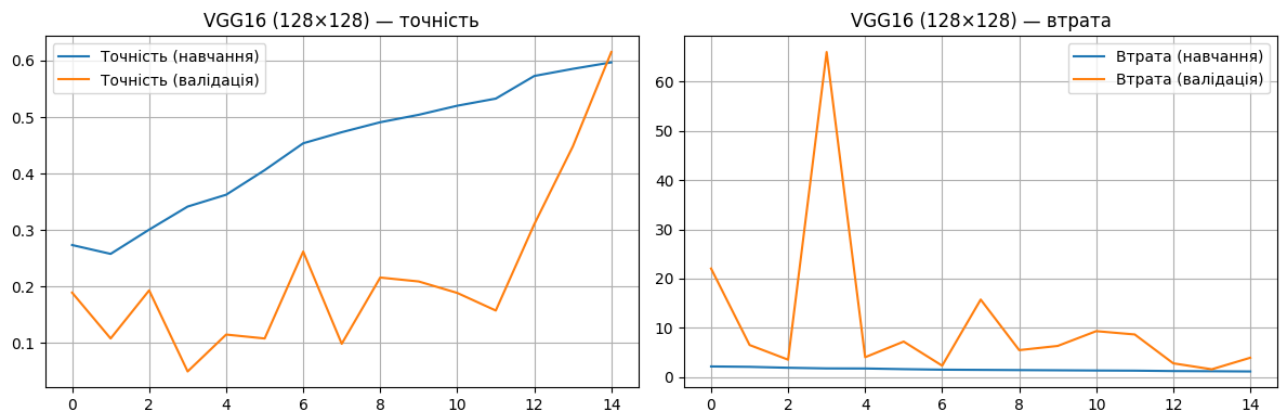


Рисунок 3.3 – VGG16 (128×128, заморожена база): криві точності та втрат;
найкраща точність на валідації 61,52%

Фінальний сценарій – тонке налаштування VGG16 для входу 128×128 із частковим розморожуванням верхніх шарів і малою швидкістю навчання – забезпечив суттєве підвищення якості. Як показано на рисунку 3.4, найкраща точність на валідації становить 91,54%, а криві навчання демонструють стабільну збіжність без різких “вибухів” втрат.

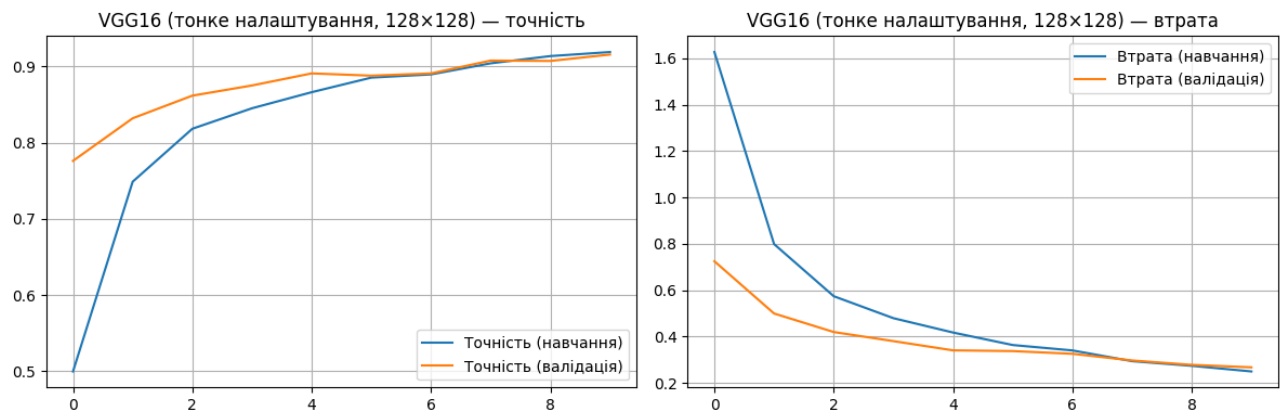


Рисунок 3.4 – VGG16 (128×128, тонке налаштування): криві точності та втрат

3.2 Оцінювання якості, аналіз помилок та порівняння моделей

Оцінювання виконано на тестовому наборі за метриками ассурасу, precision, recall, F1-score, а також із використанням матриці плутанини (у абсолютному та нормалізованому вигляді). Для базової CNN отримано точність на тесті 68,09% та втрату 1,0320, що відображено у звіті класифікації на рисунку 3.5. При цьому macro-F1 становить 0,68, що підтверджує суттєву неоднорідність якості за класами.

Втрата на тесті: 1.0320
Точність на тесті: 0.6809 (68.09%)
Виконується прогнозування для тестового набору...

Звіт класифікації:

	precision	recall	f1-score	support
Вівтар	0.83	0.66	0.74	140
Апсида	0.18	0.96	0.30	50
Дзвіниця	0.79	0.57	0.66	170
Колона	0.91	0.56	0.70	210
Купол (внутрішній)	0.92	0.86	0.89	69
Купол (зовнішній)	0.80	0.65	0.72	142
Аркбутан	0.34	0.54	0.42	70
Гаргуля	0.96	0.54	0.69	240
Вітраж	0.74	0.97	0.84	150
Склепіння	0.94	0.83	0.89	163
assurasy			0.68	1404
macro avg	0.74	0.71	0.68	1404
weighted avg	0.82	0.68	0.72	1404

Рисунок 3.5 – Базова CNN: звіт класифікації та підсумкові метрики на тесті (assurasy 68,09%)

Структуру помилок базової CNN наведено на рисунку 3.6 (матриця плутанини в абсолютних значеннях) та рисунку 3.7 (нормалізована матриця). Нормалізована форма дозволяє коректно оцінювати частки правильних/хибних рішень у кожному класі незалежно від їхньої представленості.

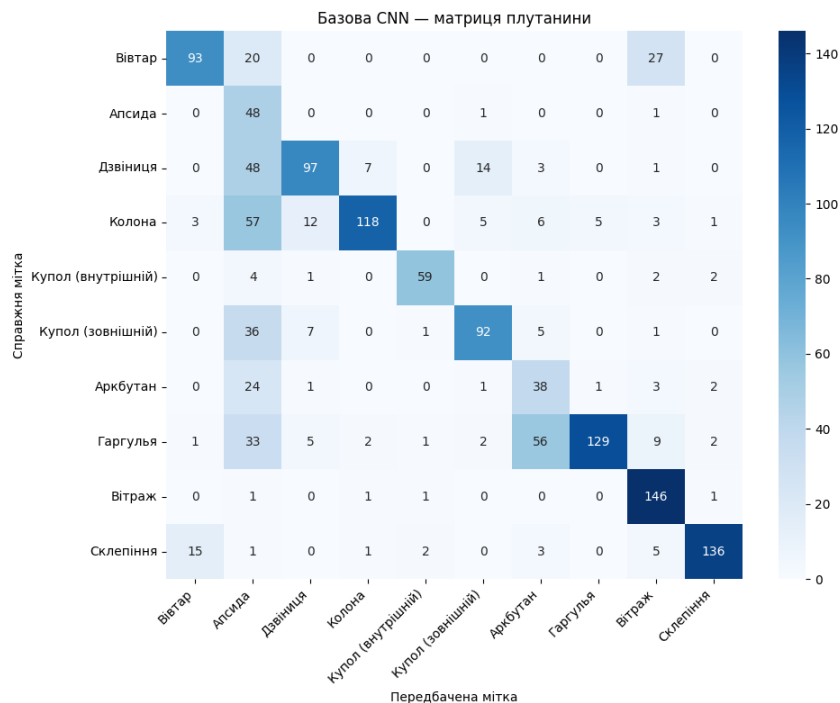


Рисунок 3.6 – Базова CNN: матриця плутанини (абсолютні значення)

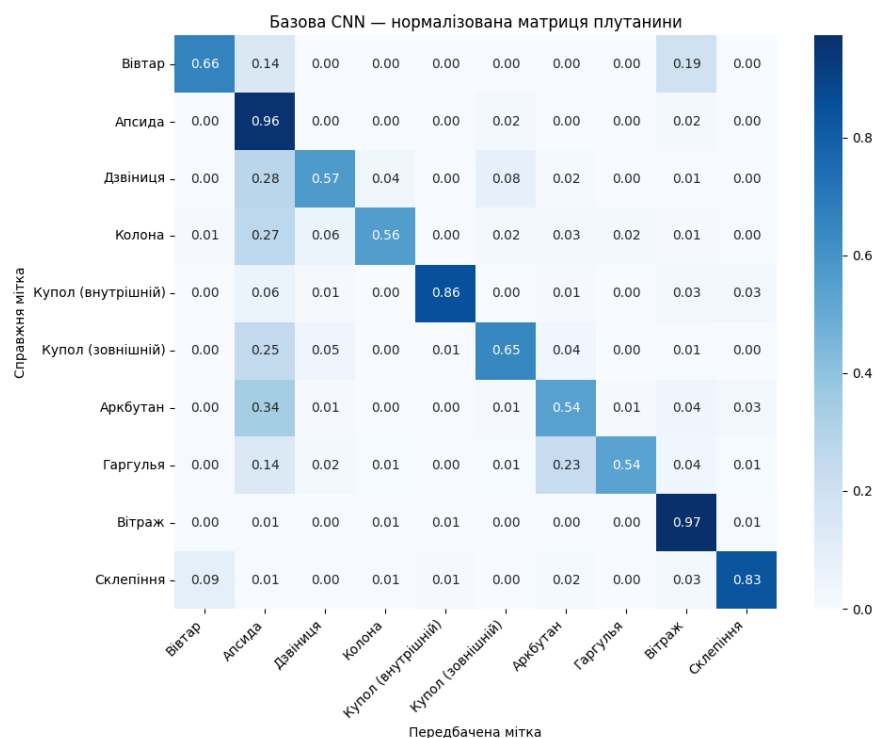


Рисунок 3.7 – Базова CNN: нормалізована матриця плутанини

Додатково виконано візуальну перевірку прогнозів на прикладах, що дозволяє інтерпретувати частину помилок як наслідок низької роздільної здатності, складного фону або подібності геометричних елементів. Приклади правильних і неправильних прогнозів наведено на рисунку 3.8.



Рисунок 3.8 – Базова CNN: приклади правильних та неправильних прогнозів

Для VGG16 із тонким налаштуванням (128×128) отримано суттєво кращі результати: точність на тесті 91,17%, втрата 0,2793. Звіт класифікації із покласовими метриками наведено на рисунку 3.9, а нормалізовану матрицю плутанини – на рисунку 3.10. Macro-усереднені значення становлять precision 0,89, recall 0,91, F1 0,90, що свідчить про збалансовану якість розпізнавання та підвищення стійкості моделі порівняно з базовою CNN.

Порівняння моделей за найкращою точністю на валідації наведено на рисунку 3.11. Встановлено, що тонке налаштування VGG16 забезпечує підвищення точності з 81,45% (базова CNN) до 91,54%, тобто на +10,09 відсоткових пунктів. Це є підставою для вибору VGG16 fine-tune як фінальної моделі програмного модуля.

Звіт класифікації (VGG16 fine-tune):

	precision	recall	f1-score	support
Вівтар	0.80	0.96	0.87	140
Апсида	0.64	0.78	0.70	50
Дзвіниця	0.95	0.87	0.91	170
Колона	0.94	0.90	0.92	210
Купол (внутрішній)	0.94	0.91	0.93	69
Купол (зовнішній)	0.92	0.98	0.95	142
Аркбутан	0.80	0.90	0.85	70
Гаргуля	0.98	0.87	0.92	240
Вітраж	0.99	0.95	0.97	150
Склепіння	0.93	0.94	0.94	163
accuracy			0.91	1404
macro avg	0.89	0.91	0.90	1404
weighted avg	0.92	0.91	0.91	1404

Рисунок 3.9 – VGG16 fine-tune (128×128): метрики на тесті (loss 0,2793; accuracy 91,17%) і звіт класифікації

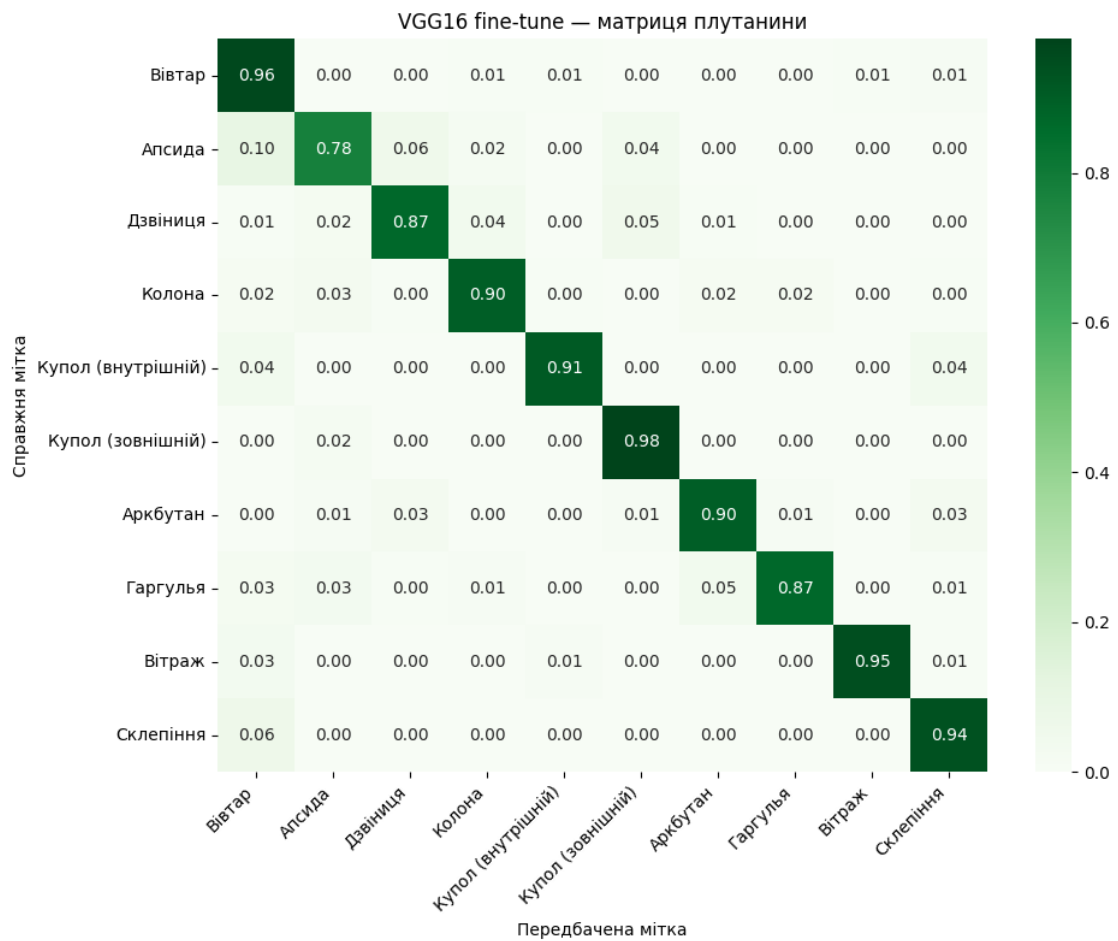
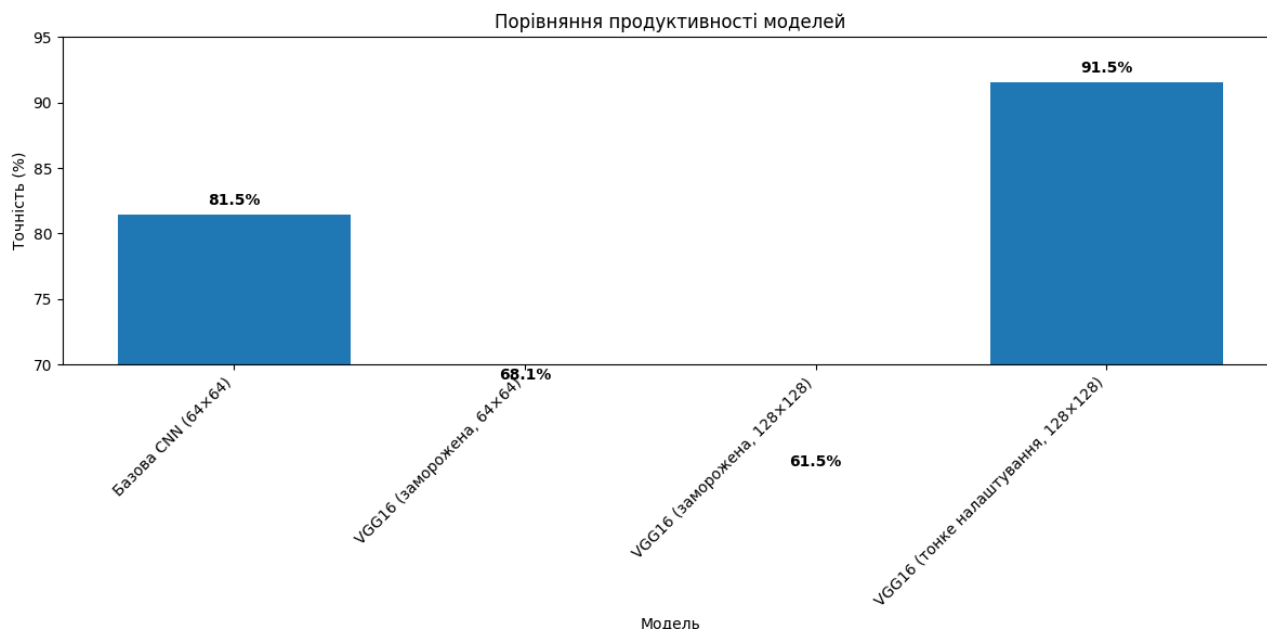


Рисунок 3.10 – VGG16 fine-tune (128×128): нормалізована матриця плутанини

ПОРІВНЯННЯ ПРОДУКТИВНОСТІ МОДЕЛЕЙ

=====

	Точність на валідації	Точність (%)
Базова CNN (64×64)	0.814540	81.45
VGG16 (заморожена, 64×64)	0.681009	68.10
VGG16 (заморожена, 128×128)	0.615232	61.52
VGG16 (тонке налаштування, 128×128)	0.915430	91.54



ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ:

Початково (базова CNN): 81.45%

Фінальна модель (fine-tune): 91.54%

Загальне покращення: +10.09 в.п.

Рисунок 3.11 – Порівняння продуктивності моделей та величина покращення (+10,09 в.п.)

3.3 Інтерфейс користувача та пояснюваність Grad-CAM

Для забезпечення практичної апробації розробленого програмного модуля реалізовано інтерактивний інтерфейс користувача на основі бібліотеки `ipywidgets`. Інтерфейс дозволяє виконувати повний цикл інференсу: завантаження або вибір тестового зображення, вибір моделі класифікації, налаштування параметра Top-k та візуалізацію результатів прогнозування.

Загальний вигляд інтерфейсу наведено на рисунку 3.12. Інтерфейс містить такі функціональні елементи:

- перемикач джерела зображення (завантаження/зразок);
- спадний список вибору моделі (базова CNN 64×64 або VGG16 fine-tune 128×128);
- повзунок параметра Top-k;
- опцію відображення Grad-CAM (для VGG16);
- кнопку запуску прогнозування.

★ Класифікація елементів архітектурної спадщини — інтерфейс

Джерело:

Завантаження Зразок

Обрати зо... **Перевір...**

Зразок: (у папці samples немає файлів) ▼

Модель: Базова CNN (64×64) ▼

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)

Рисунок 3.12 – Загальний вигляд інтерфейсу програмного модуля класифікації

Після виконання інференсу формується трикомпонентна візуалізація: вхідне зображення, діаграма Top-k ймовірностей та результат класифікації (з накладанням Grad-CAM або без нього залежно від обраної моделі). Приклад роботи модуля для моделі VGG16 з тонким налаштуванням наведено на рисунку 3.13.

У наведеному прикладі модель формує високий рівень впевненості для класу «Купол (зовнішній)» (ймовірність $\approx 0,996$), а теплова карта Grad-CAM локалізує ключові області у верхній частині купольної конструкції, що підтверджує використання семантично релевантних ознак.

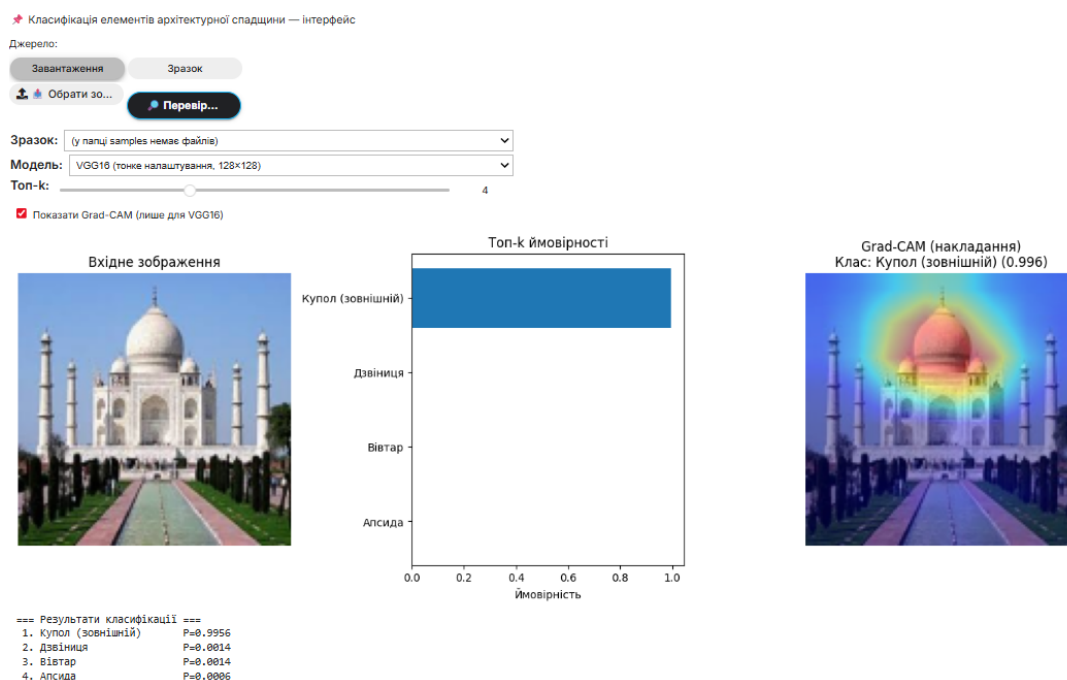


Рисунок 3.13 – Приклад інференсу в інтерфейсі: вхідне зображення, Top-k ймовірності та накладання Grad-CAM (модель VGG16 fine-tune)

Для порівняння поведінки моделей виконано тестування тих самих зображень за допомогою базової CNN. Встановлено, що в окремих випадках базова модель демонструє меншу впевненість та ширший розподіл імовірностей між конкуруючими класами. Детальні приклади роботи обох моделей, включно з різними архітектурними елементами (куполи, дзвіниці, апсиди), наведено в Додатку Б (рисунки Б.1–Б.8).

Аналіз Grad-CAM показав, що для коректних прогнозів модель VGG16 (fine-tune) концентрує увагу на морфологічно значущих фрагментах зображення (силует купола, вертикальна домінанта дзвіниці, центральна частина фасаду), тоді як при зниженні впевненості зона активації може частково зміщуватися на фонові елементи. Це свідчить про адекватність формування внутрішніх ознак та підвищує інтерпретованість отриманих результатів.

ВИСНОВКИ

У кваліфікаційній роботі досягнуто поставленої мети – розроблено та апробовано програмний модуль класифікації архітектурних елементів за зображеннями на основі моделей глибинного навчання, забезпечено порівняння альтернативних сценаріїв моделювання, кількісне оцінювання якості та інструменти прикладної експлуатації результатів у вигляді інтерфейсу й механізму пояснюваності.

1. Виконано аналіз підходів комп'ютерного зору до класифікації зображень і обґрунтовано доцільність використання згорткових нейронних мереж як базового інструментарію для розпізнавання архітектурних елементів. Показано, що класичні методи з ручними ознаками є менш стійкими до ракурсної та фотометричної варіативності, тоді як CNN забезпечують ієрархічне виділення ознак і кращу узагальнювальну здатність у задачах із високою внутрішньокласовою мінливістю. Обґрунтовано використання перенесення навчання як практично ефективної стратегії підвищення якості за помірного обсягу доменних даних.

2. Проаналізовано набір даних архітектурних елементів, визначено склад класів та окреслено ключові чинники, що впливають на складність розпізнавання: низька роздільна здатність зображень, варіативність ракурсів і освітлення, фонові домішки, оклюзії та міжкласова схожість. Встановлено вимоги до передобробки для навчання й інференсу (нормалізація, уніфікація формату, узгоджені перетворення, контрольована аугментація) та підкреслено необхідність однаковості правил обробки для уникнення розходження режимів «навчання–експлуатація».

3. Реалізовано два основні сценарії: базову CNN як еталонну модель порівняння та модель на основі VGG16 у підході перенесення навчання. Для VGG16 виконано порівняння режиму з «замороженою» згортковою базою і режиму тонкого налаштування (fine-tuning). За результатами експериментів встановлено, що fine-tuning забезпечує суттєво кращу адаптацію перенесених

ознак до домену архітектурних елементів порівняно з використанням фіксованої бази, що підтверджує доцільність доменної адаптації у практичних сценаріях.

4. Проведено оцінювання якості моделей за метриками accuracy, precision, recall і F1-score, а також виконано аналіз структури помилок за матрицею плутанини. Показано, що базова CNN забезпечує прийнятний рівень точності, проте демонструє неоднорідність якості за класами та підвищену чутливість до складних випадків (низька деталізація, домішки фону, подібність геометричних форм). Для моделі VGG16 з тонким налаштуванням зафіксовано суттєве підвищення узагальнюваності та більш збалансовані значення метрик, що підтверджує ефективність обраної архітектурно-алгоритмічної стратегії.

5. Розроблено інтерфейс користувача для інференсу, який підтримує вибір джерела зображення, вибір моделі, налаштування параметра Top-k та візуалізацію розподілу ймовірностей. Для моделі VGG16 реалізовано механізм пояснюваності на основі Grad-CAM, що забезпечує локалізацію ділянок зображення, які найбільше впливають на рішення моделі. Це підвищує прозорість прогнозування, дає можливість виявляти випадки з фокусом на фонових артефактах і є практично корисним для демонстрації коректності роботи модуля в задачах цифрової спадщини.

6. Узагальнено результати дослідження та визначено практичні сценарії застосування модуля: підтримка цифрової каталогізації та інвентаризації архівів зображень, допоміжна атрибуція фрагментів пам'яток у реставраційних задачах, формування навчально-довідкових систем, швидкий пошук подібних архітектурних фрагментів у фотоколекціях. Рекомендовано використовувати VGG16 з тонким налаштуванням як основну модель модуля, застосовувати Grad-CAM для контролю семантичної обґрунтованості рішень, а також підтримувати уніфікований пайплайн передобробки та контрольований набір аугментацій для підвищення стійкості при перенесенні на нові колекції.

Загалом результати роботи підтверджують, що поєднання глибинного навчання, перенесення навчання з тонким налаштуванням і модульної програмної реалізації з інтерфейсом та інтерпретацією забезпечує практично

придатне рішення для автоматизованої класифікації архітектурних елементів у задачах цифрової каталогізації культурної спадщини.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Lowe D. G. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision. 2004. Vol. 60, No. 2. P. 91–110. DOI: 10.1023/B:VISI.0000029664.99615.94
2. Bay H., Tuytelaars T., Van Gool L. SURF: Speeded Up Robust Features // European Conference on Computer Vision. 2006. P. 404–417. DOI: 10.1007/11744023_32
3. Dalal N., Triggs B. Histograms of Oriented Gradients for Human Detection // IEEE Conference on Computer Vision and Pattern Recognition. 2005. P. 886–893.
4. Ojala T., Pietikäinen M., Harwood D. A Comparative Study of Texture Measures with Classification Based on Feature Distributions // Pattern Recognition. 1996. Vol. 29, No. 1. P. 51–59. DOI: 10.1016/0031-3203(95)00067-4
5. Csurka G. et al. Visual Categorization with Bags of Keypoints // ECCV Workshop on Statistical Learning in Computer Vision. 2004. P. 1–22. *(DOI зазвичай відсутній)*
6. Cortes C., Vapnik V. Support-Vector Networks // Machine Learning. 1995. Vol. 20. P. 273–297. DOI: 10.1007/BF00994018
7. Breiman L. Random Forests // Machine Learning. 2001. Vol. 45. P. 5–32. DOI: 10.1023/A:1010933404324
8. Bishop C. M. Pattern Recognition and Machine Learning. New York : Springer, 2006. 738 p. ISBN: 978-0387310732
9. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition // Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. DOI: 10.1109/5.726791
10. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems. 2012. Vol. 25. P. 1097–1105.

11. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition // International Conference on Learning Representations. 2015. arXiv:1409.1556
12. Srivastava N. et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. 2014. Vol. 15. P. 1929–1958. (DOI зазвичай відсутній у JMLR)
13. Ioffe S., Szegedy C. Batch Normalization: Accelerating Deep Network Training // International Conference on Machine Learning. 2015. P. 448–456. arXiv:1502.03167
14. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 775 p. ISBN: 978-0262035613
15. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations. 2015. arXiv:1412.6980
16. Prechelt L. Early Stopping — But When? // Neural Networks: Tricks of the Trade. Berlin : Springer, 1998. P. 55–69. DOI: 10.1007/3-540-49430-8_3
17. Deng J. et al. ImageNet: A Large-Scale Hierarchical Image Database // IEEE Conference on Computer Vision and Pattern Recognition. 2009. P. 248–255. DOI: 10.1109/CVPR.2009.5206848
18. Pan S. J., Yang Q. A Survey on Transfer Learning // IEEE Transactions on Knowledge and Data Engineering. 2010. Vol. 22, No. 10. P. 1345–1359. DOI: 10.1109/TKDE.2009.191
19. Yosinski J. et al. How Transferable Are Features in Deep Neural Networks? // Advances in Neural Information Processing Systems. 2014. Vol. 27. arXiv:1411.1792
20. Kornblith S., Shlens J., Le Q. V. Do Better ImageNet Models Transfer Better? // IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. P. 2661–2671. DOI: 10.1109/CVPR.2019.00277
21. Howard J., Ruder S. Universal Language Model Fine-tuning for Text Classification // ACL. 2018. P. 328–339. DOI: 10.18653/v1/P18-1031

22. Shorten C., Khoshgoftaar T. M. A Survey on Image Data Augmentation for Deep Learning // Journal of Big Data. 2019. Vol. 6, No. 60. DOI: 10.1186/s40537-019-0197-0
23. Cubuk E. D. et al. AutoAugment: Learning Augmentation Strategies from Data // IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. P. 113–123. DOI: 10.1109/CVPR.2019.00020
24. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 770–778. DOI: 10.1109/CVPR.2016.90
25. Szegedy C. et al. Going Deeper with Convolutions // IEEE Conference on Computer Vision and Pattern Recognition. 2015. P. 1–9. DOI: 10.1109/CVPR.2015.7298594
26. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // International Conference on Machine Learning. 2019. P. 6105–6114. arXiv:1905.11946
27. Howard A. G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // arXiv preprint. 2017. arXiv:1704.04861
28. Powers D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. 2011. Vol. 2, No. 1. P. 37–63.
29. Sokolova M., Lapalme G. A Systematic Analysis of Performance Measures for Classification Tasks // Information Processing & Management. 2009. Vol. 45. P. 427–437. DOI: 10.1016/j.ipm.2009.03.002
30. Pedregosa F. et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. 2011. Vol. 12. P. 2825–2830.
31. Мелень В., Патраманська А., Пилипів Л. Комплексне застосування методів штучного інтелекту для розпізнавання та генерації об'єктів цифрової культурної спадщини. Наука сьогодення: від досліджень до стратегічних рішень: матеріали X Міжнар. студент. наук. конф. (Одеса, 27 лют. 2026 р.) / ГО «Молодіжна наукова ліга». Вінниця : ТОВ «УКРЛОГОС Груп», 2026. С. 90–92.

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.

Додаток А

Програмний код

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import cv2

# 1) Порівняння моделей (використовує вже отримані значення: baseline_acc, vgg16_64_acc, vgg16_128_acc,
finetune_acc)
results = {
    "Базова CNN (64×64)": baseline_acc,
    "VGG16 (заморожена, 64×64)": vgg16_64_acc,
    "VGG16 (заморожена, 128×128)": vgg16_128_acc,
    "VGG16 (тонке налаштування, 128×128)": finetune_acc
}

res_df = pd.DataFrame.from_dict(results, orient="index", columns=["Точність на валідації"])
res_df["Точність (%)"] = (res_df["Точність на валідації"] * 100).round(2)

print("ПІДСУМОК РЕЗУЛЬТАТІВ")
print(res_df)

plt.figure(figsize=(10, 4))
bars = plt.bar(res_df.index, res_df["Точність (%)"].values)
plt.title("Порівняння точності моделей (валідація)")
plt.ylabel("Точність (%)")
plt.xticks(rotation=35, ha="right")
for b, v in zip(bars, res_df["Точність (%)"].values):
    plt.text(b.get_x() + b.get_width()/2, b.get_height() + 0.3, f"{v:.1f}%", ha="center", va="bottom")
plt.tight_layout()
plt.show()

print(f"\nПокращення (fine-tune – базова CNN): +{(finetune_acc-baseline_acc)*100:.2f} в.п.")

# 2) Grad-CAM для одного зображення (використовує навчену модель vgg16_finetune)
def gradcam_one(img_path, model, class_names_ua, last_conv_layer="block5_conv3", size=(128,128)):
    img = tf.keras.preprocessing.image.load_img(img_path, target_size=size)
    x = tf.keras.preprocessing.image.img_to_array(img)
    x = np.expand_dims(x, axis=0) / 255.0

    grad_model = tf.keras.models.Model([model.inputs],
                                         [model.get_layer(last_conv_layer).output, model.output])

    with tf.GradientTape() as tape:
        conv_out, preds = grad_model(x)
        cls = tf.argmax(preds[0])
        loss = preds[:, cls]

    grads = tape.gradient(loss, conv_out)
    w = tf.reduce_mean(grads, axis=(0,1,2))
    cam = tf.squeeze(conv_out[0] @ w[...], tf.newaxis)
    cam = np.maximum(cam, 0) / (np.max(cam) + 1e-8)

    img_cv = cv2.cvtColor(cv2.imread(img_path), cv2.COLOR_BGR2RGB)
    img_cv = cv2.resize(img_cv, size)
    cam = cv2.resize(cam, size)
    heat = cv2.applyColorMap(np.uint8(255*cam), cv2.COLORMAP_JET)
```

```

overlay = cv2.addWeighted(img_cv, 0.6, heat, 0.4, 0)

cls_i = int(cls.numpy())
prob = float(preds[0][cls_i].numpy())

plt.figure(figsize=(10,3))
plt.subplot(1,3,1); plt.imshow(img_cv); plt.title("Оригінал"); plt.axis("off")
plt.subplot(1,3,2); plt.imshow(heat); plt.title("Grad-CAM"); plt.axis("off")
plt.subplot(1,3,3); plt.imshow(overlay); plt.title("Накладання"); plt.axis("off")
plt.suptitle(f"Передбачення: {class_names_ua[cls_i]} ({prob:.3f})", fontweight="bold")
plt.tight_layout()
plt.show()

# gradcam_one("/kaggle/input/.../test/altar/xxx.jpg", vgg16_finetune, class_names_tr)

```

Додаток Б

Тестування

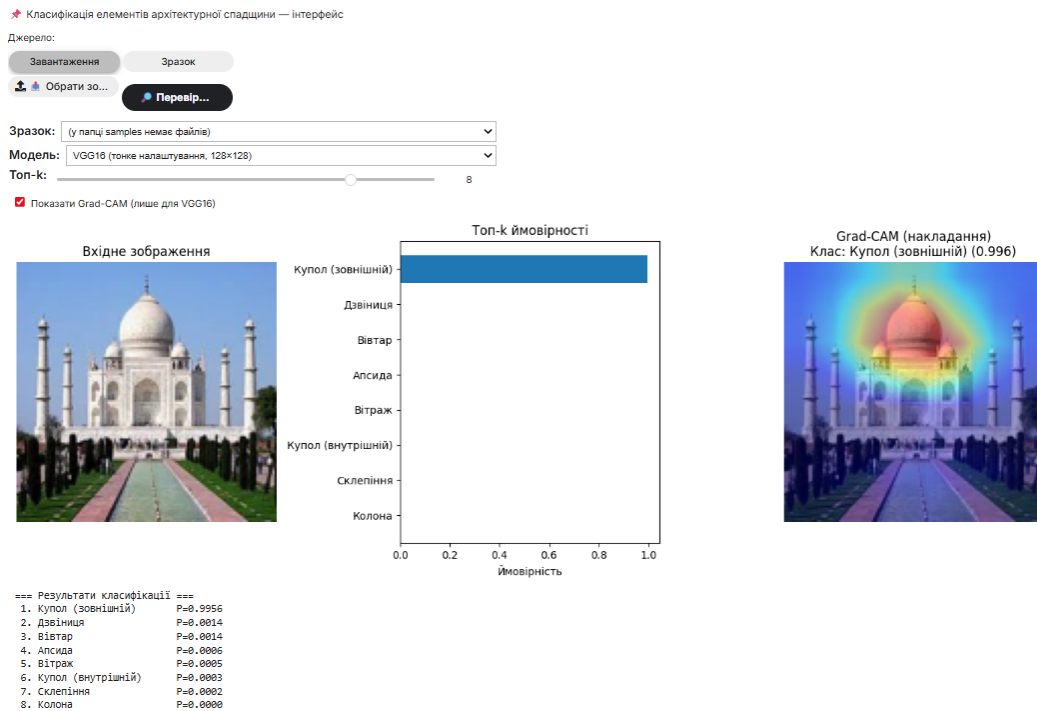


Рисунок Б.1 — Приклад роботи інтерфейсу: VGG16 (тонке налаштування, 128×128), прогноз класу «Купол (зовнішній)» із відображенням Топ-k та накладанням Grad-CAM (зразок із купольною домінантою).

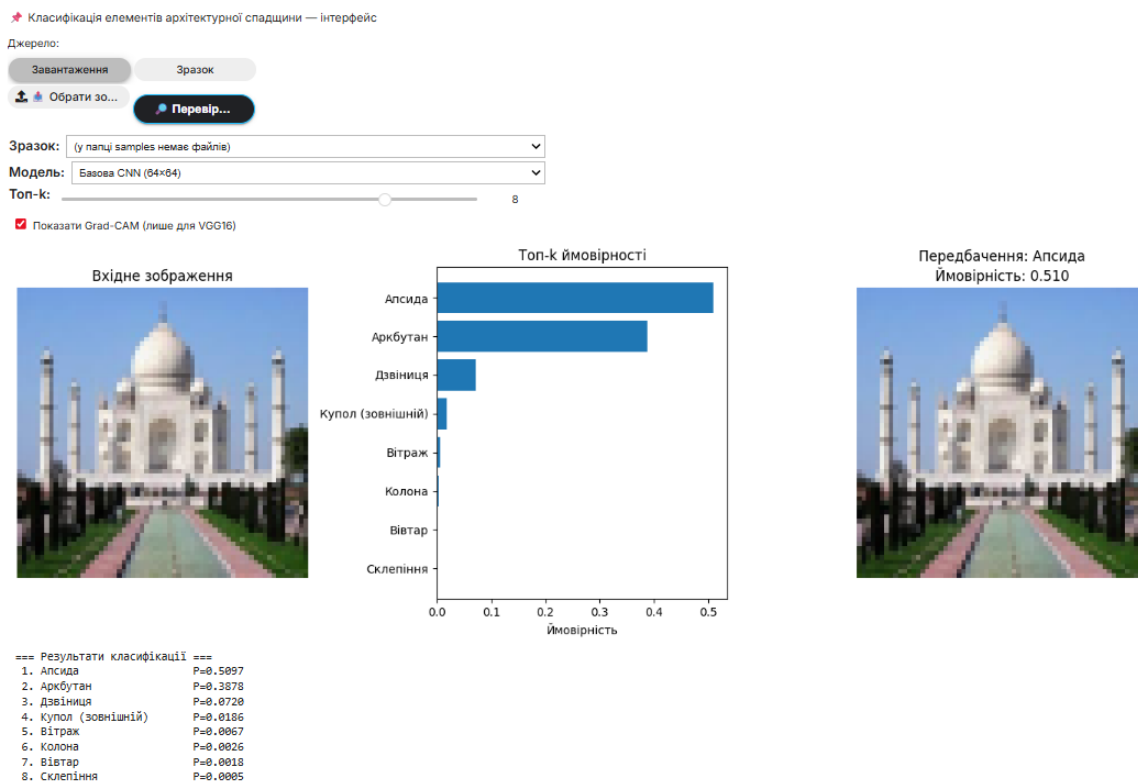


Рисунок Б.2 — Приклад роботи інтерфейсу: базова CNN (64×64), Топ-k ймовірності та прогноз для того самого зображення, що ілюструє відмінність ранжування класів порівняно з VGG16.

Джерело:

Завантаження Зразок

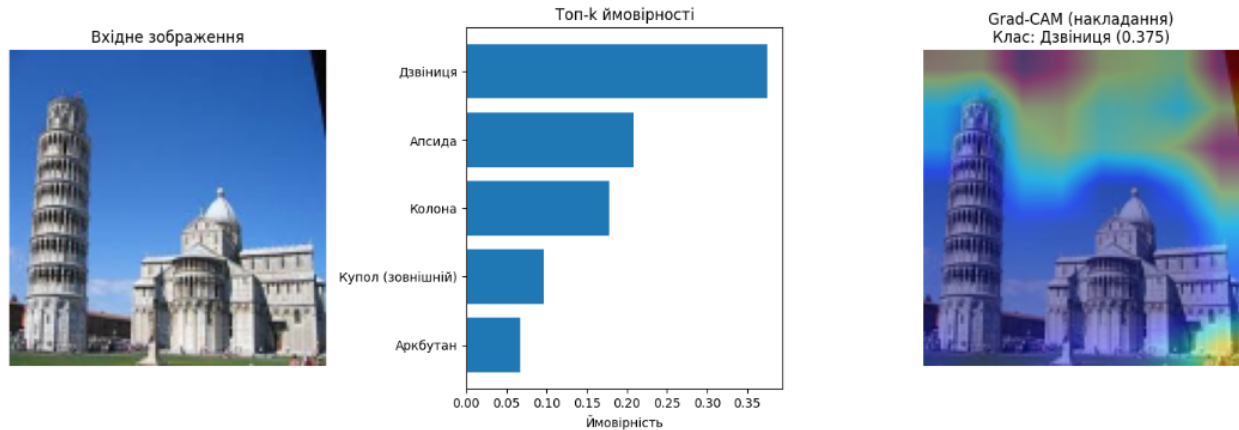
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: VGG16 (тонке налаштування, 128×128)

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Дзвіниця	P=0.3751
2. Апсида	P=0.2084
3. Колона	P=0.1782
4. Купол (зовнішній)	P=0.0962
5. Аркбутан	P=0.0675

Рисунок Б.3 – Приклад роботи інтерфейсу: VGG16 (тонке налаштування, 128×128), прогноз класу «Дзвіниця» з накладанням Grad-CAM та діаграмою Топ-k (сцена з вертикальною домінантою).

Джерело:

Завантаження Зразок

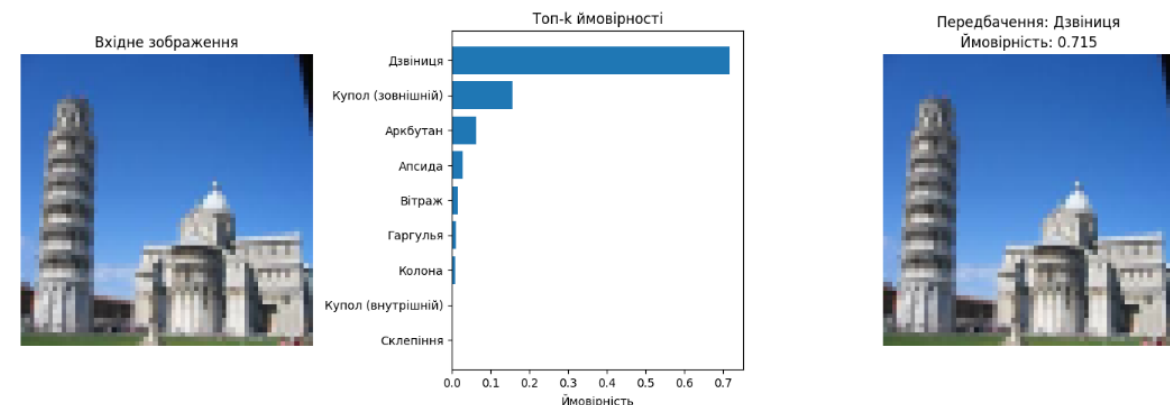
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: Базова CNN (64×64)

Топ-k: 9

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Дзвіниця	P=0.7155
2. Купол (зовнішній)	P=0.1573
3. Аркбутан	P=0.0623
4. Апсида	P=0.0290
5. Вітраж	P=0.0161
6. Гаргуля	P=0.0100
7. Колона	P=0.0087
8. Купол (внутрішній)	P=0.0007
9. Склепіння	P=0.0003

Рисунок Б.4 – Приклад роботи інтерфейсу: базова CNN (64×64), Топ-k ймовірності та прогноз для сцени з вертикальною домінантою (порівняння з Рисунок Б.3).

Джерело:

Завантаження Зразок

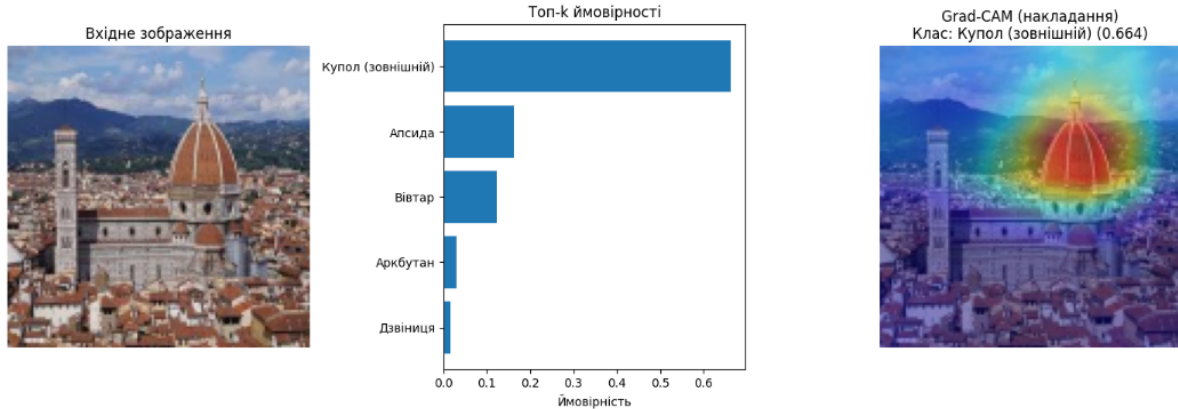
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: VGG16 (тонке налаштування, 128×128)

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Купол (зовнішній)	R=0.6637
2. Апсида	R=0.1634
3. Віттар	R=0.1235
4. Аркбутан	R=0.0303
5. Дзвіниця	R=0.0159

Рисунок Б.5 – Приклад роботи інтерфейсу: VGG16 (тонке налаштування, 128×128), прогноз класу «Купол (зовнішній)» для панорамного зображення; інтерпретація через Grad-CAM.

Джерело:

Завантаження Зразок

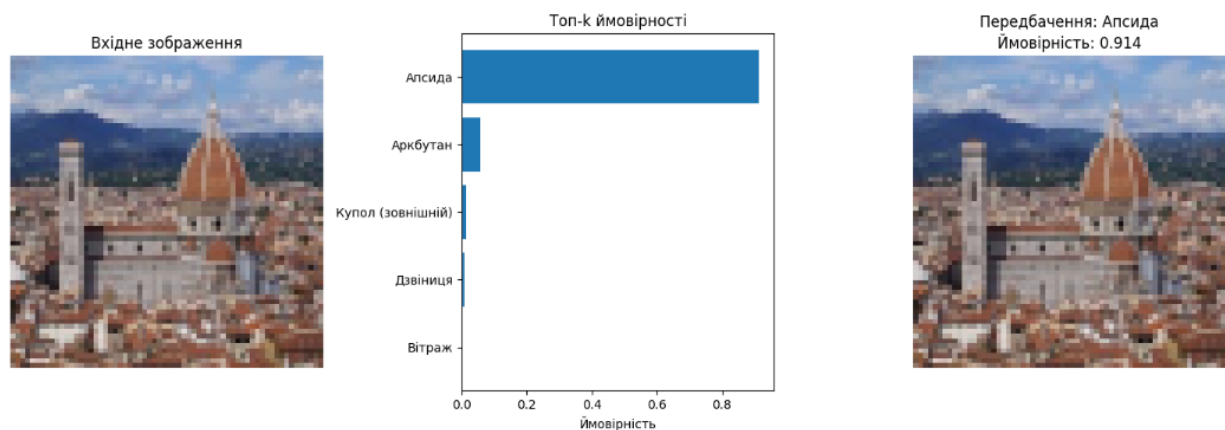
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: Базова CNN (64×64)

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Апсида	R=0.9139
2. Аркбутан	R=0.0566
3. Купол (зовнішній)	R=0.0148
4. Дзвіниця	R=0.0088
5. Вітраж	R=0.0040

Рисунок Б.6 – Приклад роботи інтерфейсу: базова CNN (64×64), Топ-k ймовірності та прогноз для панорамного зображення з куполом (порівняння з Рисунком Б.5).

Джерело:

Завантаження Зразок

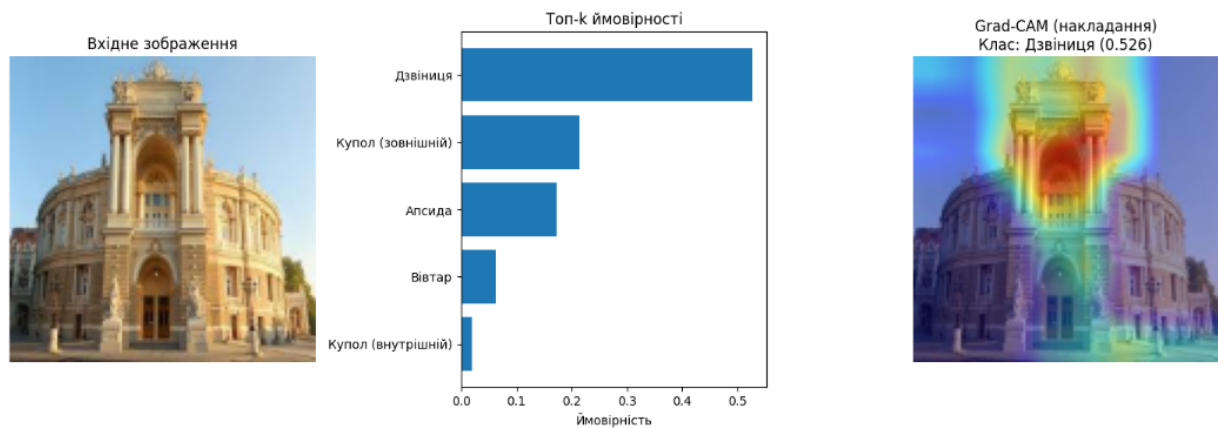
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: VGG16 (тонке налаштування, 128×128)

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Дзвіниця	P=0.5264
2. Купол (зовнішній)	P=0.2145
3. Апсида	P=0.1720
4. Вівтар	P=0.0620
5. Купол (внутрішній)	P=0.0192

Рисунок Б.7 – Приклад роботи інтерфейсу: VGG16 (тонке налаштування, 128×128), прогноз класу «Дзвіниця» з накладанням Grad-CAM для фасадної сцени зі складним фоном.

Джерело:

Завантаження Зразок

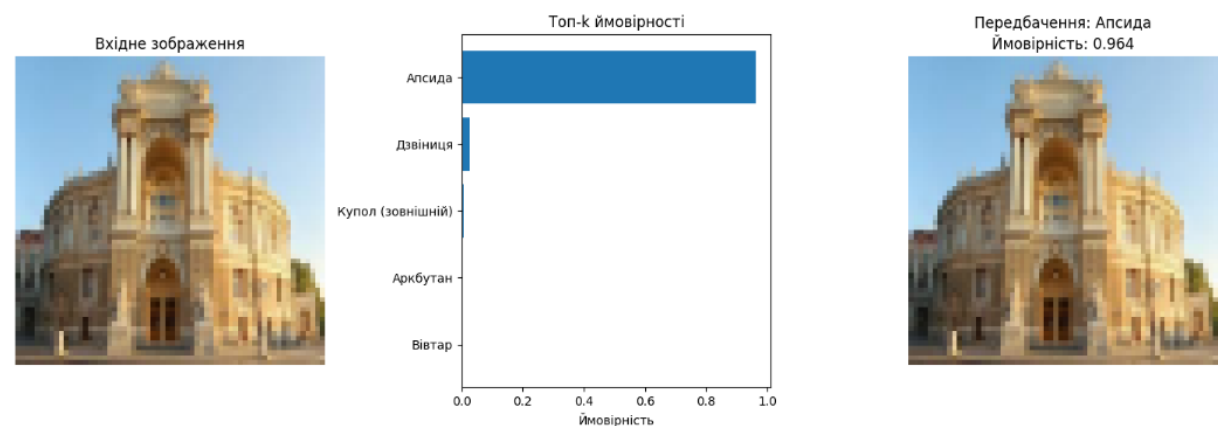
Обрати зо... Перевір...

Зразок: (у папці samples немає файлів)

Модель: Базова CNN (64×64)

Топ-k: 5

☒ Показати Grad-CAM (лише для VGG16)



=== Результати класифікації ===

1. Апсида	P=0.9638
2. Дзвіниця	P=0.0261
3. Купол (зовнішній)	P=0.0048
4. Аркбутан	P=0.0042
5. Вівтар	P=0.0007

Рисунок Б.8 – Приклад роботи інтерфейсу: базова CNN (64×64), Топ-k ймовірності та прогноз для фасадної сцени зі складним фоном (порівняння з Рисунком Б.7).

Додаток В
Апробація отриманих результатів

МАТЕРІАЛИ Х МІЖНАРОДНОЇ
СТУДЕНТСЬКОЇ НАУКОВОЇ
КОНФЕРЕНЦІЇ

НАУКА СЬОГОДЕННЯ:
ВІД ДОСЛІДЖЕНЬ ДО
СТРАТЕГІЧНИХ РІШЕНЬ



М. ОДЕСА, УКРАЇНА

**27 ЛЮТОГО
2026 РІК**

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).

Матеріали конференції знаходяться у відкритому доступі на умовах ліцензії Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).

Н 44

Наука сьогодення: від досліджень до стратегічних рішень:
матеріали Х 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 Х 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

© Колектив учасників конференції, 2026
© ГО «Молодіжна наукова ліга», 2026
© ТОВ «УКРЛОГОС Груп», 2026

СЕКЦІЯ 14. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

ЕКСПЕРТНІ ЗНАННЯ І ТОЧНІСТЬ ПРОГНОЗУВАННЯ: ПОРІВНЯННЯ PROPHET ТА LSTM	
Виговський Б.А.	86
КОМПЛЕКСНЕ ЗАСТОСУВАННЯ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗПІЗНАВАННЯ ТА ГЕНЕРАЦІЇ ОБ'ЄКТІВ ЦИФРОВОЇ КУЛЬТУРНОЇ СПАДЩИНИ	
Мелень В., Патраманська А., Пилипів Л., Науковий керівник: Лип'яніна-Гончаренко Х.	90
ПОРІВНЯЛЬНИЙ АНАЛІЗ ХМАРНИХ РІШЕНЬ ДЛЯ ВЕБ-СЕРВІСУ З ОРЕНДИ ТА КУПІВЛІ НЕРУХОМОСТІ	
Скульба С.О., Науковий керівник: Цвіркун О.А.	93

СЕКЦІЯ 15. ТРАНСПОРТ ТА ТРАНСПОРТНІ ТЕХНОЛОГІЇ

ВПРОВАДЖЕННЯ МЕТАНОЛУ ЯК НИЗЬКОВУГЛЕЦЕВОГО ПАЛИВА — ЕФЕКТИВНИЙ ШЛЯХ ДО ДЕКАРБОНІЗАЦІЇ	
Гуменюк А.В., Науковий керівник: Левківська А.Л.	95

СЕКЦІЯ 16. ФІЛОЛОГІЯ ТА ЖУРНАЛІСТИКА

ПОЛІТИЧНА МЕТАФОРА В СОЦІОЛІНГВІСТИЧНОМУ АСПЕКТІ (НА МАТЕРІАЛІ ЗМІ)	
Чутро В.С., Науковий керівник: Конопелькіна О.О.	97
ПРО ВАЖЛИВІСТЬ СИСТЕМАТИЗАЦІЇ ТЕРМІНОСИСТЕМИ БЕЗПЕКОВИХ УТОД УКРАЇНИ	
Муранова В.В., Науковий керівник: Коцюк Л.М.	99

СЕКЦІЯ 17. ПЕДАГОГІКА ТА ОСВІТА

CHALLENGES OF ONLINE EDUCATION IN GEORGIA	
Mchedlishvili N., Sadagashvili M., Scientific supervisor: Khorguashvili T.	102
АСПЕКТИ РЕАЛІЗАЦІЇ ІНТЕГРОВАНОГО НАВЧАННЯ В ПОЧАТКОВІЙ ШКОЛІ	
Цехмістер Б.В., Науковий керівник: Поберецька В.В.	105
ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ НА УРОКАХ ІНФОРМАТИКИ В ПОЧАТКОВІЙ ШКОЛІ	
Даценко М.С., Науковий керівник: Рибалко О.О.	108
МЕТОД ПРОБЛЕМНОГО НАВЧАННЯ ЯК ЗАСІБ АКТИВІЗАЦІЇ ПІЗНАВАЛЬНОЇ ДІЯЛЬНОСТІ УЧНІВ У ПРОЦЕСІ ВИВЧЕННЯ НІМЕЦЬКОЇ МОВИ	
Каменська Н.В.	110
ОСОБЛИВОСТІ ЛОГОПЕДИЧНОЇ РОБОТИ З ДІТЬМИ З КОМПЛЕКСНИМИ ПОРУШЕННЯМИ ПСИХОФІЗИЧНОГО РОЗВИТКУ В УМОВАХ ІНКЛЮЗИВНОЇ ОСВІТИ	
Комисарик Є.О., Науковий керівник: Савінова Н.В.	113

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 (Ассигасу), 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 входними 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.