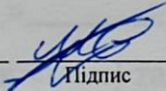
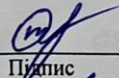
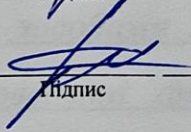


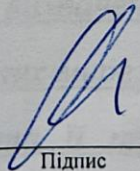
Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук

## КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

Галузь знань 12 – Інформаційні технології  
Шифр і назва галузі знань  
Спеціальність 122 – Комп'ютерні науки  
Шифр і назва спеціальності  
Освітня програма Комп'ютерні науки  
Назва освітньої програми

Виконав: студент групи КНс-22-1  Олександр МУШТИН  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ  
Керівник: асистент каф. КН  Олена ТИЩЕНКО  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ  
Нормоконтроль: к.т.н., доц. каф. КН  Руслан БАГРІЙ  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:  
Зав. кафедри КН, д.т.н., професор  Олександр БАРМАК  
Підпис Ім'я, ПРІЗВИЩЕ

18 06 2025 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
Факультет інформаційних технологій  
Кафедра комп'ютерних наук  
Освітній ступінь бакалавр  
Галузь знань 12 – Інформаційні технології  
Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ  
Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор Олександр БАРМАК  
« 10 » 02 2025 року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж»

2. Завдання видано студенту Олександру Муштину  
(Ім'я, прізвище)

3. Керівник роботи асистент кафедри КН Олена Тищенко  
(посада, ім'я, прізвище)

4. Затверджено наказом університету від « 07 » 02 2025 р. № 23

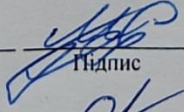
5. Дата видачі завдання студенту: « 10 » 02 2025 р.

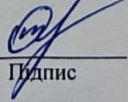
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи – підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.  
Завдання дослідження полягає у: проведенні аналізу предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням; створенні методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж; проектуванні інформаційної структури інтелектуальної системи та її програмній реалізації на основі запропонованого методу; дослідженні ефективності роботи створеного методу з використанням розробленої інтелектуальної системи.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником, складання календарного графіка виконання	січень 2025	Виконано
2	Ознайомлення з предметною областю, формулювання мети і задач дослідження, визначення об'єкта та предмета дослідження	лютий 2025	Виконано
3	Проектування та розроблення методу вирішення завдання, загальної архітектури програмного забезпечення, інтерфейсу користувача, вибір засобів реалізації програмного забезпечення	березень 2025	Виконано
4	Створення та тестування програмного забезпечення, дослідження ефективності, висновки з виконаної роботи	квітень 2025	Виконано
5	Написання пояснювальної записки, урахування зауважень керівника, оформлення згідно з вимогами	травень 2025	Виконано
6	Розробка презентаційних матеріалів та попередній захист кваліфікаційної роботи	травень 2025	Виконано
7	Отримання відгуку керівника, рецензії, перевірка на плагіат, нормоконтроль	червень 2025	Виконано
8	Підготовка до захисту та захист кваліфікаційної роботи	червень 2025	Виконано

Виконавець: студент групи КНС-22-1  Олександр МУШТИН  
Група виконавця Підпис Ім'я, ПРІЗВИЩЕ

Керівник: асистент каф. КН  Олена ТИЩЕНКО  
Науковий ступінь, посада Підпис Ім'я, ПРІЗВИЩЕ

## Анотація

Тема кваліфікаційної роботи бакалавра: «Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж»

Виконавець кваліфікаційної роботи бакалавра: студент групи КНс-22-1 Олександр Муштин

Керівник кваліфікаційної роботи бакалавра: асистент кафедри КН Олена Тищенко

Кваліфікаційна робота бакалавра містить:

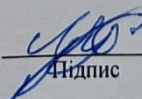
Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
60	14	2	46	6

Метою кваліфікаційної роботи бакалавра є підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж. Для розробки інтелектуальної системи використано мову програмування Python, мікрофреймворк Flask, а також бібліотеки tensorflow, numpy для виконання багатокласової класифікації з допомогою нейромережевої моделі MobileNetV2 навченою за методом fine-tuning. Розроблена система призначена для автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

Напрямами практичного використання розробленої інтелектуальної системи визначено надання висновків щодо визначеного архітектурного стилю за зображенням.

Ключові слова: архітектура, стилі, розпізнавання, багатокласова класифікація, MobileNetV2, fine-tuning.

Виконавець: студент групи КНс-22-1  
Група виконавця

  
Підпис

Олександр МУШТИН  
Імя, ПРІЗВИЩЕ

## Зміст

Перелік скорочень .....	4
Вступ.....	5
Розділ 1 Характеристика предметної області визначення архітектурних стилів за зображенням .....	7
1.1 Аналіз інформаційних моделей визначення архітектурних стилів за зображенням .....	7
1.2 Огляд відомих підходів до аналізу зображень.....	11
1.3 Аналіз наукових досліджень з визначення архітектурних стилів за зображенням .....	15
1.4 Мета, задачі та вимоги до реалізації інтелектуальної системи .....	18
Розділ 2 Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.....	19
2.1 Загальний підхід до визначення архітектурних стилів за зображенням.....	19
2.2 Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж .....	20
2.3 Отримання нейромережевої моделі для визначення архітектурних стилів за зображенням .....	22
2.4 Метрики для оцінювання ефективності нейромережевої моделі .....	23
2.5 Архітектура нейромережевої моделі для визначення архітектурних стилів за зображенням.....	24
2.6 Формування датасету для навчання нейромережевої моделі .....	27
2.7 Проєктування інтелектуальної системи визначення архітектурних стилів.	29
2.8 Висновки до розділу 2 .....	32
Розділ 3 Програмна реалізація інтелектуальної системи та експериментальне дослідження методу .....	34
3.1 Вибір засобів розробки інтелектуальної системи визначення архітектурних стилів .....	34

3.2 Структура та функціональне призначення програмних складових інтелектуальної системи визначення архітектурних стилів .....	35
3.3 Особливості програмної реалізації інтелектуальної системи визначення архітектурних стилів.....	38
3.4 Експериментальне дослідження .....	45
3.5 Висновки до розділу 3 .....	52
Загальні висновки.....	53
Перелік посилань.....	55
Додатки	

**Перелік скорочень**

<b>Скорочення, термін, позначення</b>	<b>Пояснення</b>
БД	База даних
ІТ	Інформаційні технології
КН	Комп'ютерні науки
НМ	Нейронна мережа
ІІ	Штучний інтелект
CNN	Згорткові нейроні мережі
МН	Машинне навчання
VGG16	Visual Geometry Group 16
FCOS	Fully Convolutional One-Stage Object Detection
CLAHE	Contrast Limited Adaptive Histogram Equalization
GPT-4	Generative Pre-trained Transformer 4
YOLO	You Only Live Once
R-CNN	Region-based Convolutional Neural Networks

## Вступ

Метою кваліфікаційної роботи бакалавра є підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж, для чого розроблено метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж, а також відповідну інтелектуальну систему визначення архітектурних стилів.

**Актуальність.** У сучасну епоху стрімкого розвитку цифрових технологій автоматичне розпізнавання візуального контенту стає дедалі важливішим у різних галузях – від культурної спадщини до урбаністики та комерційного дизайну. Визначення архітектурних стилів за зображенням є складним завданням через велику різноманітність стилістичних рис, комбінацій елементів і вплив часу на зовнішній вигляд будівель. Традиційні підходи, що базуються на ручній класифікації або аналізі окремих ознак, виявляються недостатньо ефективними у випадках великого обсягу даних або складних змішаних стилів.

Глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), довели свою високу ефективність у завданнях комп'ютерного зору, таких як класифікація об'єктів, розпізнавання образів та сегментація сцен. Їхня здатність автоматично виявляти приховані закономірності у великих масивах зображень робить їх ідеальним інструментом для розв'язання задачі визначення архітектурного стилю.

Таким чином, запропонований метод надає нові можливості для інтеграції сучасних технологій у сферу збереження архітектурної спадщини та розвитку смарт-міст.

**Об'єкт дослідження** – процес визначення архітектурних стилів за зображенням.

**Предмет дослідження** – методи та технології глибоких нейронних мереж для визначення архітектурних стилів за зображенням.

**Мета кваліфікаційної роботи бакалавра** – підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

**Завдання кваліфікаційної роботи бакалавра** – провести аналіз предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням; створити метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж; спроектувати інформаційну структуру інтелектуальної системи та виконати програмну реалізацію на основі запропонованого методу; дослідити ефективність створеного методу з використанням розробленої інтелектуальної системи.

## **Розділ 1 Характеристика предметної області визначення архітектурних стилів за зображенням**

### **1.1 Аналіз інформаційних моделей визначення архітектурних стилів за зображенням**

Людство з давніх часів створювало численні архітектурні споруди, які слугували не лише утилітарними об'єктами для задоволення базових потреб, але й виявом художнього бачення, культурної самобутності та технологічного прогресу тієї чи іншої епохи [1]. Багато з цих архітектурних шедеврів збереглися до нашого часу, і вони є об'єктами активного дослідження серед науковців різних галузей – архітектури, історії, культурології, мистецтвознавства та інформаційних технологій [2].

Вивчення архітектурних споруд має кілька важливих цілей. Воно сприяє збереженню культурної спадщини, що є основою національної ідентичності та самосвідомості суспільства [3]. Дослідження історичних об'єктів дозволяє реконструювати події минулого, зрозуміти особливості соціального устрою, рівень розвитку технологій, мистецькі пріоритети та світогляд людей різних епох [4].

Вивчення архітектурних стилів дає змогу простежити еволюцію формотворення, матеріалів та конструктивних рішень у будівництві [5]. Аналіз архітектурних об'єктів дозволяє виявити закономірності розвитку архітектури як мистецтва та науки, а також зрозуміти механізми запозичення та трансформації стилістичних елементів між різними регіонами і культурами [6].

Систематизація знань про архітектурні стилі є надзвичайно важливою для сучасної архітектурної практики [7]. Вона дозволяє фахівцям застосовувати історичний досвід для проектування нових об'єктів у контексті збереження історико-архітектурного середовища, що особливо актуально у процесах реставрації, реконструкції та інтеграції новобудов у старовинну міську структуру [8].

Крім того, вивчення архітектурної спадщини має значення і для туристичної індустрії, популяризації національної культури на міжнародному рівні, розвитку освітніх програм у сфері мистецтва та архітектури [9].

Архітектурний стиль – це сукупність характерних ознак і принципів формотворення, що визначають вигляд і конструктивні особливості архітектурних споруд певного історичного періоду, регіону або художнього напрямку [10]. Стиль в архітектурі охоплює як зовнішній вигляд будівлі – її форми, пропорції, декоративне оформлення – так і внутрішню логіку композиції, конструктивні рішення та використання матеріалів [11].

Кожен архітектурний стиль відображає специфічні естетичні ідеали, технічні можливості та культурно-соціальні умови свого часу. Стильова належність будівлі визначається за певними ознаками: типами конструкцій, характером об'ємно-просторових рішень, особливостями фасадів, орнаментами, використанням світла й простору [12].

Архітектурні стилі визначаються набором характеристик, що впливають на форми, структуру і конструкцію будівель. Одним із основних критеріїв для класифікації стилів є використання конкретних конструктивних рішень і матеріалів, що відповідають певним технічним і естетичним вимогам епохи [13].

Античний стиль характеризується використанням ордерних систем – основних конструктивних одиниць, таких як колони, що забезпечували як підтримку, так і естетичну виразність будівель. У цей період широко застосовувалися прості геометричні форми та строгі пропорції, що мали символічне значення, підкреслюючи гармонію між людиною та природою. Основною архітектурною формою були храми, амфітеатри та монументальні споруди, що використовували кам'яні та мармурові матеріали для досягнення довговічності [14].

Готичний стиль виник в Середньовіччі, коли були розроблені нові конструктивні методи, що дозволяли будувати високі, світлі споруди з великими вітражними вікнами. Стрільчасті арки, ребристі склепіння та контрфорси дозволяли зменшити вагу конструкцій, збільшити висоту і площу вікон, що

робило архітектуру більш відкритою та легшою. Ці технічні рішення сприяли появі вертикальних структур, що підкреслювали духовне піднесення [15].

Ренесанс, що розвинувся на основі класичних принципів античності, відновив інтерес до симетрії, геометрії і математичних пропорцій у побудові споруд. Основною рисою цього стилю стало прагнення до гармонії і балансу в пропорціях будівель, що виражалося у використанні куполів, арок і колон. Будівлі ставали більш розміреними та просторими, з акцентом на функціональність простору і естетичну витонченість [16].

Бароко, який з'явився в 17 столітті, відрізнявся розкішшю, динамічністю форм і драматизмом. Характерними рисами були великі площі, складні об'ємні композиції, використання криволінійних форм і багатих декоративних елементів. Барокові споруди зазвичай містили монументальні фасади, величезні куполи і детально оформлені інтер'єри, що сприяло створенню ефекту грандіозності та емоційного впливу на глядача [17].

Розрізнити архітектурні стилі часто складно, особливо коли мова йде про стилі, що мають подібні характеристики або розвивалися один з іншого, як це, наприклад, стосується Ренесансу та Бароко, або Готики та Ренесансу. Основним підходом до розрізнення таких стилів є аналіз ключових конструктивних і декоративних елементів, що визначають архітектуру кожного стилю, а також оцінка взаємодії цих елементів на рівні загальної композиції [18].

Для розрізнення архітектурних стилів також важливо враховувати загальні форми споруди та її кольорову гаму. Одні стилі тяжіють до симетрії, простих геометричних форм і стриманих тонів, як-от класицизм, тоді як інші — до складних, асиметричних композицій з вигнутими лініями та яскравими кольорами, наприклад, модерн. Колірні рішення можуть відображати не лише естетичні вподобання епохи, а й матеріали, характерні для певного регіону чи часу. Наприклад, ренесанс часто асоціюється з теплою палітрою природного каменю, а конструктивізм — з контрастом білого, сірого й чорного.

Варто також зважати на масштаб будівлі, її пропорції, спосіб організації простору навколо – відкриті площі, внутрішні двори, аркади чи галереї можуть бути ознаками окремих стилів. Крім того, важливу роль відіграє ритм розміщення вікон, дверей і декоративних елементів: регулярний чи хаотичний, витончений або масивний.

У той час як Ренесанс, хоча й бере за основу класичні античні принципи, має більш збалансовані та симетричні форми, без такої ж вираженої вертикальності. Ренесансні будівлі, як правило, мають чітко виражену симетрію і пропорційні елементи, де відсутня надмірна декоративність, характерна для готичної архітектури [19].

Бароко і Ренесанс, хоча й обидва стилі розвивалися в Європі на межі середньовіччя та нового часу, мають значні відмінності. Бароко відзначається складними, динамічними формами та екстравагантністю, де кожен елемент композиції здається «рухомим», що надає будівлі динаміки. Ренесанс, навпаки, акцентує увагу на спокійних, врівноважених формах і симетрії. Тут також варто звертати увагу на декоративні елементи: барокова архітектура використовує вигнуті лінії, великі куполи та складні фасади, у той час як у Ренесансі домінують прості, чітко окреслені форми, геометричні композиції [20, 21].

Розробка програмних засобів для автоматизованого визначення архітектурних стилів дозволить досягти Цілей сталого розвитку ПРООН. Ціль №11 «Сталий розвиток міст і громад» буде досягнута через автоматизацію процесу збереження культурної спадщини, що сприятиме відновленню та реставрації важливих архітектурних об'єктів. Ціль №9 «Індустріалізація, інновації та інфраструктура» буде досягнута через сприяння розвитку нових технологій у сфері архітектури, що підвищать ефективність проектування та будівництва, сприяючи економічному зростанню і створенню сталих інфраструктур [24].

Отже, актуальними залишаються питання автоматизованого визначення архітектурного стилю за зображенням шляхом виділення ключових візуальних

ознак з метою збереження культурної спадщини, а також стимуляції інновацій та сталого розвитку інфраструктури.

## 1.2 Огляд відомих підходів до аналізу зображень

Враховуючи складність і варіативність завдання визначення архітектурних стилів за зображеннями, інформаційні технології здатні значно полегшити цей процес [22]. Використання методів комп'ютерного зору дозволяє автоматизувати процеси аналізу зображень будівель, виділяючи характерні ознаки стилю, що не завжди очевидні для людського ока. Завдяки використанню великих датасетів архітектурних зображень та навчання моделей на таких зображеннях, є можливість створити алгоритми, які здатні точно класифікувати архітектурні стилі на основі візуальних характеристик, навіть якщо стилі дуже схожі між собою [23]. Це дозволить забезпечити більш точну і швидку ідентифікацію архітектурних об'єктів, значно спростивши процес їх аналізу та систематизації.

Завдання автоматичної класифікації архітектурних стилів за зображеннями привертає значну увагу дослідників у галузі комп'ютерного зору, оскільки поєднує проблеми візуального розпізнавання з культурно-історичною семантикою. Для розв'язання цього завдання дедалі частіше застосовуються згорткові нейронні мережі, які довели свою ефективність у витяганні складних візуальних ознак з необроблених піксельних даних та їх подальшій інтерпретації в контексті класифікаційних задач [22]. Глибокі нейромережеві моделі не лише автоматично ієрархізують ознаки різного рівня абстракції, а й здатні адаптуватися до специфіки візуального матеріалу, що є особливо важливим у випадку стилістично різномірної архітектури.

Процес розпізнавання архітектурного стилю розгортається на кількох рівнях обробки вхідного зображення. На початкових шарах нейромережа виявляє прості візуальні патерни, такі як краї, контури, контрастні переходи або базові геометричні форми. Ці ознаки не мають самостійного стилістичного

значення, однак вони слугують основою для формування більш складних структур. На середньому рівні абстракції модель інтегрує локальні ознаки у фрагменти об'єктів, наприклад, частини фасадів, арочні прорізи, форми колон, характерні геометрії вікон або даху. Нарешті, на вищих шарах модель оперує глобальними уявленнями про об'єкт або сцену загалом, що дозволяє врахувати композиційні особливості, симетрію, масштабність і характерні деталі, які є визначальними для конкретного архітектурного стилю [24].

Класифікація архітектури значно відрізняється від задач загального розпізнавання об'єктів, оскільки стилістичні відмінності можуть бути тонкими, багатошаровими та культурно обумовленими. Наприклад, готична архітектура відрізняється стрілочастими арками та вертикальними пропорціями, у той час як ренесансна характеризується гармонією, симетрією та круглими формами. Мережа навчається ідентифікувати ці відмінності не за формальними описами, а за статистично стабільними візуальними патернами, які повторюються у зображеннях, віднесених до одного стилю. Таким чином, ознаки, що вивчаються моделлю, є не обов'язково лінгвістично означеними або інтуїтивно зрозумілими, однак вони репрезентують суттєві структури, що дозволяють здійснювати точну класифікацію [25].

Визначення архітектурних стилів за зображенням є складним завданням через велику варіативність форм та стилістичних ознак будівель. CNN є основним підходом до обробки зображень завдяки своїй здатності автоматично виділяти важливі ознаки без необхідності ручного визначення характеристик. Це дозволяє використовувати CNN для класифікації архітектурних стилів, де важливо враховувати не тільки загальні елементи, але й деталі, які визначають конструктивні особливості будівель [26].

Для задачі класифікації архітектурних стилів велику роль відіграють підходи на основі попередньо натренованих моделей – transfer learning. Використовуючи моделі, попередньо натреновані на великих наборах даних, таких як ImageNet, можна значно прискорити процес навчання, адаптуючи вже отримані знання до специфіки архітектурних зображень. Transfer learning

дозволяє значно зменшити вимоги до обчислювальних ресурсів та часу на навчання, що робить його оптимальним вибором для застосувань у сфері класифікації архітектурних стилів [27].

Популярні архітектури CNN, такі як AlexNet, VGG, ResNet і EfficientNet, є основними моделями, що використовуються для аналізу зображень у задачах класифікації.

AlexNet був одним із перших великих досягнень у сфері глибокого навчання для обробки зображень. Запропонований у 2012 році Алексом Кріжевським, він став основою для революції у комп'ютерному зорі завдяки своїй здатності працювати з великими наборами даних, використовуючи графічні процесори (GPU). Мережа складається з восьми шарів, з яких п'ять – це згорткові шари, а три – повнозв'язні. Однією з ключових інновацій AlexNet є використання ReLU (Rectified Linear Units) як активаційної функції, що значно прискорює навчання і дозволяє мережі досягати кращих результатів у порівнянні з традиційними активаційними функціями, такими як сигмоїда. AlexNet став основним рушієм для розвитку глибоких нейронних мереж і досі залишається важливою базовою архітектурою для багатьох задач класифікації [28].

Нейромережа VGG, розроблена Visual Geometry Group з Оксфордського університету, виділяється своєю простотою та ефективністю. Основна ідея VGG полягає в тому, що використання маленьких фільтрів (3x3) в згорткових шарах дозволяє будувати глибокі мережі без втрати якості та точності. Мережа VGG може мати різну глибину, від 16 до 19 шарів, що дає можливість використовувати її для складних завдань, де потрібна висока точність. Однією з переваг VGG є те, що вона добре справляється з великими наборами даних і дає гарні результати на широкому спектрі задач, включаючи класифікацію зображень, детекцію об'єктів та сегментацію. Простота структури VGG дозволяє ефективно адаптувати її до специфічних задач, що робить її популярною у дослідженнях і реальних застосуваннях [29]

EfficientNet є новою архітектурою, яка оптимізує глибину, ширину та розмір зображень для досягнення найкращих результатів за мінімальні

обчислювальні витрати. Використовуючи принципи *compound scaling*, EfficientNet досягає оптимального балансу між розміром мережі та точністю. Це дозволяє створювати моделі, які є набагато більш ефективними в порівнянні з попередніми архітектурами, такими як ResNet або VGG, при цьому значно знижуючи вимоги до ресурсів. EfficientNet використовує менш складні мережі, що робить її привабливою для завдань, де важливо обробляти великі обсяги даних, зокрема у випадках, коли доступні обмежені обчислювальні ресурси. Завдяки своїй ефективності та точності, EfficientNet є перспективною архітектурою для класифікації архітектурних стилів, де важливо досягти високої точності при обмежених ресурсах [28].

ResNet, або Residual Networks, є однією з найбільш значущих архітектур в області глибоких нейронних мереж. Головною особливістю цієї архітектури є використання резидуальних зв'язків (*skip connections*), які дозволяють передавати вихід попереднього шару без змін на наступний. Це вирішує проблему деградації, яка виникає при збільшенні глибини мережі, коли додавання нових шарів починає погіршувати результати замість їх покращення. ResNet дозволяє створювати дуже глибокі мережі (до кількох сотень шарів), що значно підвищує точність на складних завданнях, таких як класифікація зображень, де потрібно виявляти дрібні деталі. В результаті, ResNet став стандартом для багатьох задач обробки зображень і добре адаптується до різних варіантів класифікації архітектурних стилів [30].

MobileNetV2 є легкою та ефективною згортковою нейронною мережею, спеціально розробленою для роботи в умовах обмежених обчислювальних ресурсів. Вона була представлена дослідниками з Google у 2018 році як покращення попередньої версії MobileNet і ґрунтується на кількох архітектурних принципах, які забезпечують високу точність при зменшеній кількості параметрів та низькому обчислювальному навантаженні. Це робить MobileNetV2 особливо придатною для задач класифікації в реальному часі на мобільних пристроях або в хмарних середовищах з обмеженими ресурсами, таких як Google Colab. Ключовою інновацією MobileNetV2 є використання інвертованих

резидуальних блоків (inverted residual blocks) у поєднанні з лінійною вузькою проєкцією (linear bottleneck) [31]. На відміну від класичних глибоких мереж, де глибокі шари поступово розширюють простір ознак, MobileNetV2 реалізує зворотний підхід: кожен блок спочатку розширює вхідний простір за допомогою точкової згортки ( $1 \times 1$  convolution), далі застосовується глибинна згортка (depthwise convolution), яка виконується окремо для кожного каналу, після чого результат стискається назад у вузьке представлення за допомогою ще однієї точкової згортки. На завершальному етапі блоку застосовується або не застосовується резидуальне з'єднання, залежно від відповідності розмірностей вхідного і вихідного тензорів [32]. Загалом, MobileNetV2 поєднує низьку обчислювальну складність з високою точністю, що робить її придатною для класифікації зображень, де важливо досягти балансу між якістю розпізнавання і швидкістю обробки. Її архітектурна компактність також дозволяє працювати з порівняно малими наборами зображень, що є типовим для досліджень в архітектурній сфері, де дані можуть бути обмеженими або важкодоступними [33].

Отже, застосування наведених підходів для класифікації архітектурних стилів дозволяє не тільки покращити точність та швидкість аналізу, а й сприяє автоматизації процесів, що можуть значно полегшити роботу архітекторів, істориків архітектури та дослідників у цій галузі. Доцільним є використання transfer learning та нейромережевої архітектури MobileNetV2 для завдання визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

### **1.3 Аналіз наукових досліджень з визначення архітектурних стилів за зображенням**

Сучасний розвиток інформаційних технологій і штучного інтелекту відкриває нові можливості для аналізу та класифікації архітектурних об'єктів. Особливу роль у цьому процесі відіграють глибокі нейронні мережі, які

дозволяють автоматизувати завдання розпізнавання архітектурних стилів на основі зображень. У цьому розділі розглянуто наукові дослідження, що присвячені для класифікації архітектурних об'єктів.

Автори статті [34] розв'язували задачу класифікації архітектурних стилів будівель культурної спадщини Афін. Для цього вони використали глибоке навчання з використанням алгоритму YOLO, що застосовується для обробки зображень. Наведений підхід дозволив автоматично категоризувати дані з вуличних зображень. Завдяки цьому методу вдалося досягти ефективної організації та класифікації даних, що сприяє покращенню збереження культурної спадщини. Значення метрик для всіх класів: Precision 74,71%, Recall 74,44%, F1-метрика 74,20%.

Запропоновано метод класифікації архітектурних стилів, який поєднує CNN з канално-просторовою увагою авторами статті [35]. Спочатку виконується попередня обробка для вибору основних регіонів будівлі, після чого CNN витягує глибокі ознаки. Модуль канално-просторової уваги покращує текстурні ознаки та фокусується на просторових елементах архітектури. Класифікація виконується за допомогою Softmax. Тестування на наборах Architectural Style Dataset і AHE\_Dataset показало хороші результати, покращуючи точність класифікації завдяки врахуванню як текстурних, так і просторових характеристик будівель.

У статті [36] автори розробили zero-shot класифікатор віку будівель за зображеннями фасадів, використовуючи GPT-4 Vision без додаткового навчання. Вони застосували текстові підказки з логічними інструкціями для управління процесом класифікації. Як тестову область було обрано Лондон, для чого створили новий набір даних FI-London, що містить зображення фасадів та відповідні епохи побудови будівель. У результатах експерименту автори отримали загальну точність класифікації 39,69% і середню абсолютну помилку (MAE) 0,85 десятиліття. Найкращі показники спостерігалися для будівель, зведених у періоди 1920–1939 років ( $F1 = 56,25\%$ ) та 2000–2019 років ( $F1 = 66,67\%$ ). Для інших епох точність і повнота були значно нижчими, особливо для

дуже старих будівель (до 1700 року), де модель фактично не змогла правильно класифікувати об'єкти. Висока похибка для найстаріших будівель і складність розрізнення близьких за віком періодів (до двох десятиліть) підтверджують, що GPT-4 Vision краще справляється із загальною оцінкою епохи, ніж із точним визначенням року побудови.

Автори [37] розв'язували задачу автоматичної класифікації історичних китайських будівель типу «qilou», поєднуючи методи обробки зображень і глибокі нейронні мережі. Для цього вони застосували алгоритм Contrast Limited Adaptive Histogram Equalization (CLAHE) для покращення якості зображень і використали поліпшену версію Faster R-CNN з ResNet50 (Faster R-CNN-R) для розпізнавання будівель. Для навчання та тестування використовувався набір із 760 зображень qilou, знятих у Гуанчжоу. Запропонований підхід показав кращі результати у порівнянні з Faster R-CNN на Visual Geometry Group 16 (VGG16) і Fully Convolutional One-Stage Object Detection (FCOS): досягнута точність склала 80,12%.

Автори статті [38] досліджували використання штучного інтелекту в архітектурі, зосередившись на класифікації архітектурних стилів різних періодів за допомогою згорткових нейронних мереж. Для реалізації підходу було використано бібліотеку TensorFlow та мову програмування Python. Модель CNN навчалася на наборах даних із зображеннями фасадів будівель у готичному, модерному та деконструктивістському стилях. Результати показали, що CNN здатна ефективно розрізняти складні деталі архітектурних стилів, досягнувши точності класифікації 84,66%. Крім того, автори зазначили потенціал такого підходу для вивчення історичних процесів, естетичних змін і підтримки процесів реставрації архітектурних об'єктів.

Проаналізовані публікації демонструють високу ефективність використання глибокого навчання для класифікації архітектурних стилів за зображеннями фасадів. Застосування CNN та їхніх модифікацій дозволяє досягати високих показників точності навіть для складних і близьких між собою стилістичних епох. Такі рішення не тільки автоматизують процес аналізу

архітектури, а й відкривають нові перспективи для цифрової архівізації, збереження культурної спадщини та підтримки сталого розвитку у сфері містобудування.

Розглянуті публікації присвячені визначенню архітектурних стилів за зображеннями, попри їхню ефективність, мають ряд суттєвих обмежень. Моделі на основі You Only Live Once (YOLO) та Faster R-CNN орієнтовані переважно на локалізацію об'єктів, що знижує їхню здатність до глибокої стилістичної диференціації. Zero-shot класифікація за допомогою великих візуально-мовних моделей, таких як Generative Pre-trained Transformer-4 Vision (GPT-4), демонструє недостатній рівень точності, особливо для будівель із невираженими або змішаними стилістичними характеристиками. Також наведені методи у розглянутих роботах орієнтовані на малу кількість класів архітектурних стилів. З огляду на виявлені обмеження існуючих підходів, постає необхідність у розробці нового методу визначення архітектурних стилів за зображеннями.

#### **1.4 Мета, задачі та вимоги до реалізації інтелектуальної системи**

Метою кваліфікаційної роботи бакалавра є підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням;
- розробити метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж;
- здійснити проектування та програмну реалізацію спроектованої інтелектуальної системи на основі запропонованого методу;
- виконати експериментальне дослідження методу з використанням розробленої інтелектуальної системи визначення архітектурних стилів.

## Розділ 2 Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

### 2.1 Загальний підхід до визначення архітектурних стилів за зображенням

Зважаючи на обмеження існуючих рішень, що наведено у пункті 1.3, доцільно запропонувати загальний підхід роботи (рисунок 2.1), який спрямований на досягти поставленої мети – підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

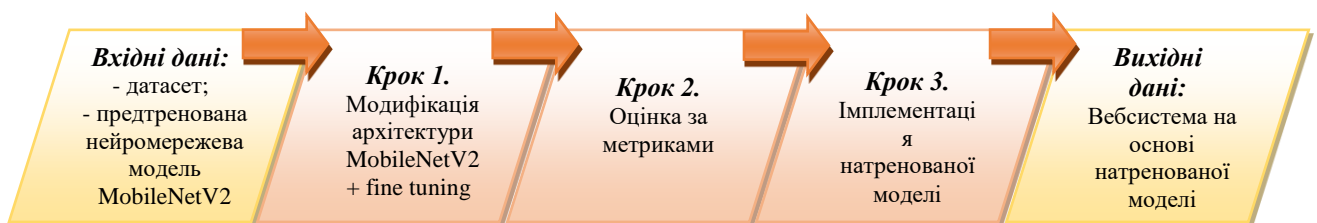


Рисунок 2.1 – Підхід до визначення архітектурних стилів за зображенням

Вхідними даними є датасет для навчання, що містить зображення архітектурних стилів та предтренована нейронна мережа модель MobileNetV2.

На першому кроці відбувається модифікація архітектури MobileNetV2 із подальшим донавчанням на датасеті, що містить зображення архітектурних стилів. Це дозволяє адаптувати модель під конкретну задачу, а саме визначення архітектурних стилів за зображенням. На другому кроці здійснюється оцінка моделі за визначеними метриками (п. 3.2), що дає змогу визначити її точність класифікації. Після цього виконується третій крок, який передбачає імплементацию натренованої моделі MobileNetV2, що здатна виконувати автоматичну класифікацію архітектурних стилів, у вебсистему.

Вихідними даними є вебсистема на основі натренованої моделі MobileNetV2, що спроможна досягти вищої точності, ніж відомі аналоги.

Отже, описаний підхід до роботи передбачає використання попередньо натренованої моделі MobileNetV2, адаптованої до задачі класифікації

архітектурних стилів шляхом її донавчання на відповідному датасеті. Послідовне виконання етапів модифікації архітектури та донавчання, оцінювання якості моделі за метриками та її інтеграції у вебсистему забезпечить розв'язання задачі автоматизованого визначення архітектурних стилів за зображеннями. Такий підхід дозволить виконати поставлені завдання та досягнути мети роботи.

## 2.2 Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

Глибокі нейронні мережі надають можливості для визначення архітектурних стилів за зображенням завдяки своїй здатності витягувати глибокі, багаторівневі ознаки з вхідних зображень. На рисунку 2.2 наведено схему методу визначення архітектурних стилів за зображенням, який базується на використанні згорткової нейронної мережі MobileNetV2 та спрямований на вирішення задачі багатокласової класифікації архітектурних стилів.

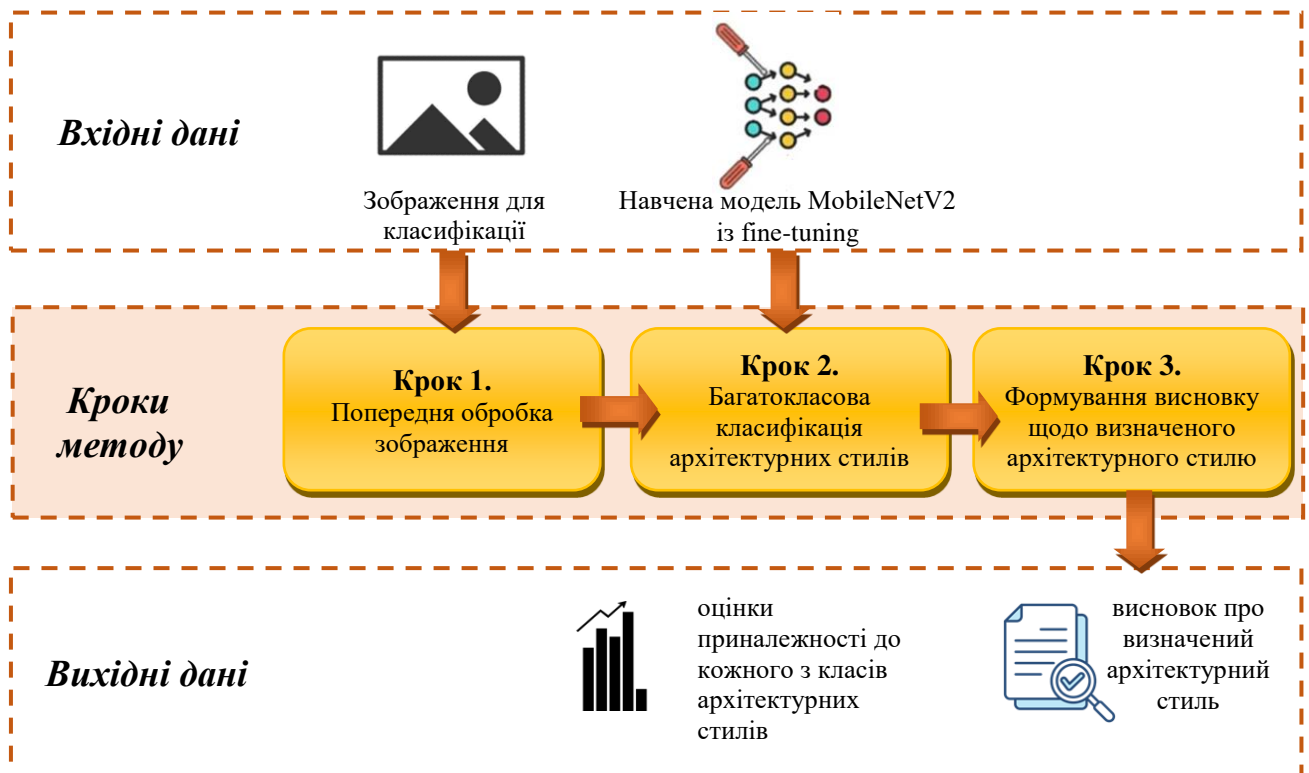


Рисунок 2.2 – Схема методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

Вхідними даними є зображення архітектурних об'єктів, а також попередньо навчена модель для автоматизованого аналізу вхідного зображення.

Перший крок методу передбачає попередню обробку зображення для аналізу. Попередня обробка полягає в тому, що кожне зображення, яке аналізується, зчитується, змінюється його розмір до заданого стандарту MobileNetV2, а потім значення пікселів перетворюються у формат, зручний для роботи нейронної мережі MobileNetV2. Зокрема, пікселі нормалізуються – тобто переводяться в числовий діапазон від  $-1$  до  $1$ , щоб мережа могла ефективно їх обробляти.

Другий крок реалізує багатокласову класифікацію, у якій згортова нейромережева модель MobileNetV2 із fine-tuning, попередньо навчена на великому наборі зображень ImageNet і донавчена під задачу класифікації архітектурних стилів, обробляє вхідне зображення. Зображення подається у нейронну мережу, де попередньо навчені шари витягують ієрархію ознак – від простих країв і форм до складніших композиційних структур, характерних для конкретних архітектурних стилів. Отримані ознаки передаються у нові класифікаційні шари, які були донавчені на вміст датасету, що містить зображення архітектурних стилів, і на виході модель генерує ймовірнісний розподіл між усіма доступними класами, визначаючи до якого архітектурного стилю належить вхідне зображення.

На третьому кроці формуються результати класифікації шляхом обробки вихідного шару моделі, який повертає ймовірнісні оцінки належності зображення до кожного з навчальних класів архітектурних стилів. Ці оцінки інтерпретуються як відсоткове представлення впевненості моделі, де сума всіх значень становить  $100\%$ . Найвища з цих оцінок вказує на той архітектурний стиль, до якого модель з найбільшою ймовірністю відносить вхідне зображення. Таким чином, надається не лише остаточний класифікаційний висновок, а й повна оцінка розподілу ймовірностей між усіма можливими класами, що дозволяє аналізувати ступінь впевненості моделі у своєму рішенні.

Вихідними даними методу є оцінки приналежності до кожного з класів архітектурних стилів та висновок про визначений архітектурний стиль.

Таким чином, метод забезпечує автоматизоване визначення архітектурного стилю зображення для аналізу з використанням моделі глибокого навчання.

### 2.3 Отримання нейромережевої моделі для визначення архітектурних стилів за зображенням

Отримання моделі глибокого навчання для багатокласової класифікації архітектурних стилів потребує дотримання кроків, що подані на рисунку 2.3. Схема на рисунку демонструє послідовність кроків від вибору архітектури моделі глибокого навчання до отримання готової моделі, здатної класифікувати архітектурні стилі за зображенням.



Рисунок 2.3 – Схема отримання моделі глибокого навчання для багатокласової класифікації архітектурних стилів

Вхідними даними є датасет архітектурних стилів. Далі відбувається вибір відповідної архітектури нейронної мережі. Після цього обрана нейромережева архітектура піддається процесу навчання на підготовленому наборі зображень. Навчена модель потім оцінюється за допомогою метрик (таких як Accuracy, Precision, Recall, F1-міра), що дозволяє проаналізувати її ефективність у класифікації архітектурних стилів. Якщо показники метрик є задовільними, то така модель використовується для класифікації архітектурних стилів. Якщо

метрики є незадовільними, то відбувається повторний вибір архітектури НМ або корекція моделі.

Отже, в пункті наведено кроки для отримання моделі глибокого навчання для багатокласової класифікації архітектурних стилів.

## 2.4 Метрики для оцінювання ефективності нейромережевої моделі

Оцінка ефективності моделей глибокого навчання є важливим для розуміння якості навченої моделі для класифікації архітектурних стилів. Для цього використовуються формалізовані метрики, що базуються на матриці сплутувань та дозволяють кількісно виміряти якість передбачень моделі як у межах окремих класів, так і для всієї вибірки в цілому.

Оцінювання якості моделей класифікації здійснюється на основі матриці помилок (confusion matrix), яка відображає порівняння між передбаченими та фактичними класами.

Матриця сплутувань – це табличне подання результатів класифікації, яке відображає зіставлення передбачених і фактичних міток класів [45]. У найпростішому випадку (бінарна класифікація) вона має вигляд:

- TP (True Positives) – кількість зразків, які правильно класифіковані як позитивні;
- TN (True Negatives) – кількість зразків, які правильно класифіковані як негативні;
- FP (False Positives) – кількість зразків, які помилково класифіковані як позитивні;
- FN (False Negatives) – кількість зразків, які помилково класифіковані як негативні.

У випадку багатокласової класифікації, матриця має розмір  $N \times N$ , де  $N$  – кількість класів, і кожна діагональна клітинка  $C_{ii}$  відображає правильно класифіковані приклади для класу  $i$ , тоді як позадіагональні – помилки класифікації.

Accuracy оцінює загальну правильність класифікації, тобто частку прикладів, що були правильно класифіковані.

Precision оцінює, яка частка об'єктів, передбачених як позитивні, дійсно є позитивними. Recall визначає, яку частку позитивних об'єктів модель змогла правильно виявити.

F1-міра – гармонійне середнє між Precision та Recall, яке узагальнює обидва показники в одному числі [46].

У багатокласовій класифікації також обчислюються усереднені:

- Macro F1 – середнє значення F1-міри по всіх класах (не зважаючи на розмір класів);

- Weighted F1 – середнє з урахуванням ваг (кількості прикладів) кожного класу.

Таким чином, метрики на основі матриці сплутувань, Accuracy, Precision, Recall, F1-міра дозволяють оцінити ефективність навченої моделі для класифікації архітектурних стилів за зображенням.

## **2.5 Архітектура нейромережевої моделі для визначення архітектурних стилів за зображенням**

У даному пункті наведено та описано архітектуру згорткової нейронної мережі MobileNetV2 (рисунок 2.4), яка використовується для багатокласової класифікації архітектурних стилів за зображенням на кроці 2 методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж, що наведений у п. 2.2.

Зображена на рисунку 2.4 архітектура моделі є модифікованою нейромережею на основі архітектури MobileNetV2, розширеною для задачі класифікації на 25 класів.

У наведеній архітектурі реалізовано transfer learning, тобто перенесення знань з однієї задачі на іншу. Цей підхід полягає в тому, що модель, попередньо навчена на великому загальному датасеті (у даному випадку ImageNet),

використовується як основа для розв'язання нової, часто вужчої задачі – класифікації зображень на 25 класів. Transfer learning тут реалізується через включення попередньо натренованої архітектури MobileNetV2 як екстрактора ознак. Також варто зазначити, що transfer learning в даному випадку реалізовано через стратегію часткового донавчання fine-tuning, що є розширенням базового transfer learning. Завдяки розморожуванню верхніх 30 шарів базової моделі (тобто найглибших шарів, які відповідають за більш специфічні абстракції), мережа здобуває здатність адаптувати високорівневі ознаки до особливостей нового датасету, а саме датасету, що містить зображення архітектурних стилів.

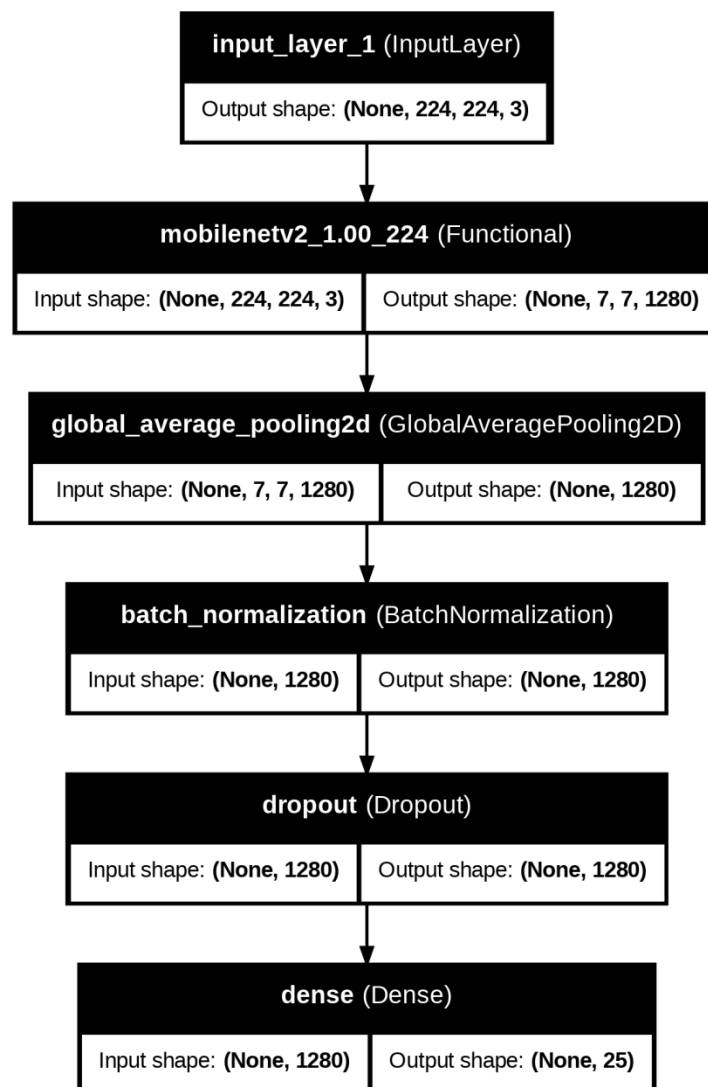


Рисунок 2.4 – Архітектура згорткової нейронної мережі MobileNetV2

Вхід до моделі здійснюється через об'єкт `tf.keras.Input` з формою тензора (224,224,3), де `IMG_SIZE` встановлено на 224. Далі використовується базова модель `MobileNetV2` без класифікаційної голови (`include_top=False`), а її ваги ініціалізуються з використанням попередньо навчених параметрів на `ImageNet`. Принципово важливим є те, що початкові шари цієї мережі заморожуються, тобто встановлюються як такі, що не тренуються. Конкретно, тренування дозволено лише для останніх 30 шарів мережі, починаючи з `fine_tune_at`, що забезпечує збереження загальних візуальних ознак, набутих під час попереднього навчання, при адаптації до нової задачі.

Після обробки зображення через `base_model` у режимі `training=False`, що виключає вплив динамічних компонентів як-от `Dropout` або `BatchNormalization` у фазі тренування, результати передаються на глобальний шар агрегації ознак – `GlobalAveragePooling2D`. Він, як уже згадувалося, виконує стискання просторових вимірів через середні значення для кожного каналу. Отриманий вектор ознак надалі нормалізується за допомогою шару `BatchNormalization`, що зменшує варіативність між мінібатчами.

Регуляризація моделі реалізована через шар `Dropout` із заданою ймовірністю виключення нейронів  $p=0.3$ , що дозволяє ефективно знижувати ризик перенавчання в умовах обмеженої кількості навчальних зразків. На завершальному етапі модель містить повнозв'язний шар `Dense`, який реалізує класифікацію за допомогою функції активації `softmax`. Кількість виходів цього шару динамічно визначається як довжина словника `label_map`, що відповідає 25 класам.

Модель компілюється з використанням оптимізатора `Adam` із малим коефіцієнтом навчання  $\alpha=3 \times 10^{-5}$ , що є типовим для етапу донавчання з частковою фіксацією ваг. Функцією втрат виступає `sparse_categorical_crossentropy`, яка використовується для багатокласової класифікації при поданні міток у вигляді цілих чисел.

У процесі навчання застосовуються три ключові механізми `callbacks`. Перший – `EarlyStopping`, який зупиняє тренування при відсутності покращення

функції втрат на валідаційному наборі впродовж п'яти епох. Другий – ReduceLROnPlateau, який автоматично зменшує швидкість навчання вдвічі при стабілізації втрат, що сприяє глибшій локальній оптимізації. Третій – ModelCheckpoint, який зберігає найкращу модель за критерієм валідаційної точності.

Таким чином, запропонована архітектура НМ демонструє реалізацію сучасного підходу до тонкого донавчання глибоких згорткових нейромереж. Вона поєднує переваги попередньо навчених моделей з адаптацією до конкретної предметної області, гарантуючи збалансовану продуктивність і стабільність навчання, а отже дозволить підвищити точність багатокласової класифікації архітектурних стилів за зображенням.

## **2.6 Формування датасету для навчання нейромережевої моделі**

Для навчання НМ обрано набір даних «Architectural Styles Dataset» [39], розміщений на платформі Kaggle користувачем dimitrux, містить 10 113 зображень, класифікованих за 25 архітектурними стилями. Цей датасет був сформований шляхом поєднання зображень, отриманих із Google Images, та зображень з іншого датасету, згаданого в описі, однак точне джерело не вказано. Приклади зображень з датасету наведено на рисунку 2.5.

Кожне зображення в датасеті асоційоване з одним із 25 архітектурних стилів, що дозволяє використовувати цей набір даних для задач багатокласової класифікації зображень. Завдяки різноманітності стилів та значному обсягу даних, цей датасет є цінним ресурсом для досліджень у галузі комп'ютерного зору, зокрема для навчання та оцінки моделей, здатних розпізнавати та класифікувати архітектурні стилі на основі візуальних характеристик.

Загальна кількість зображень датасету становить 10113, а кількість зображень у класах варіюється залежно від стилю, що зумовлює помітну диспропорцію між обсягами представлення окремих класів, що видно на рисунку 2.6.

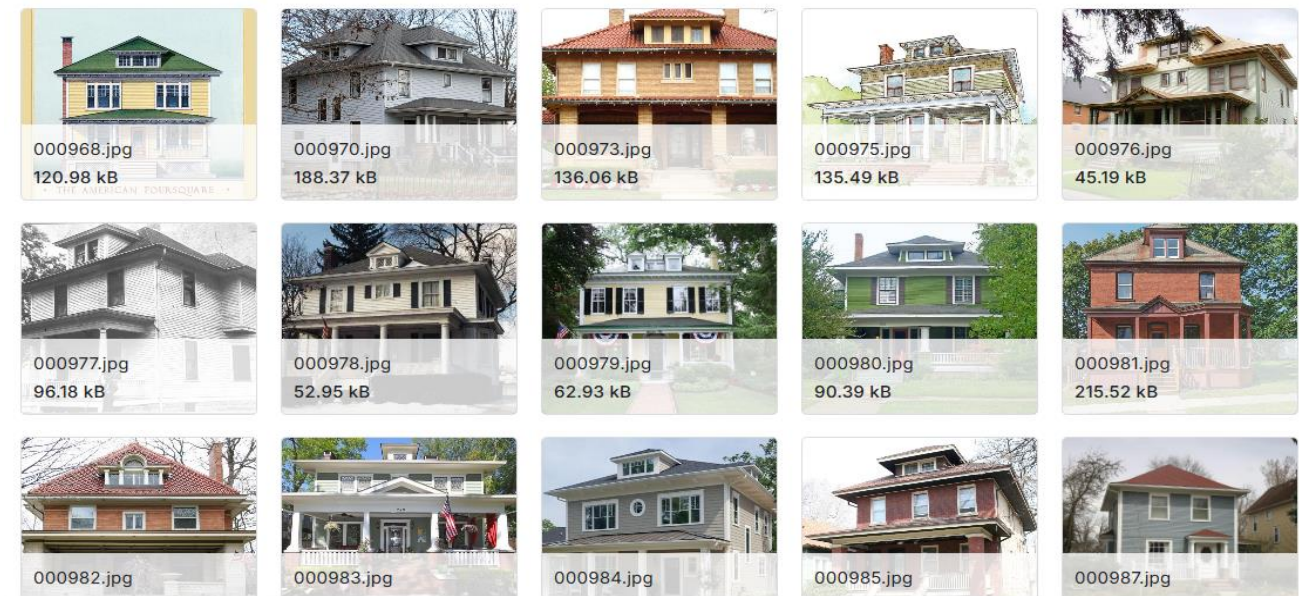


Рисунок 2.5 – Приклади зображень з класу American Foursquare architecture датасету

Це, у свою чергу, є важливим фактором при використанні цього датасету для задач класифікації за допомогою методів машинного навчання, оскільки потребує корекції дисбалансу класів.

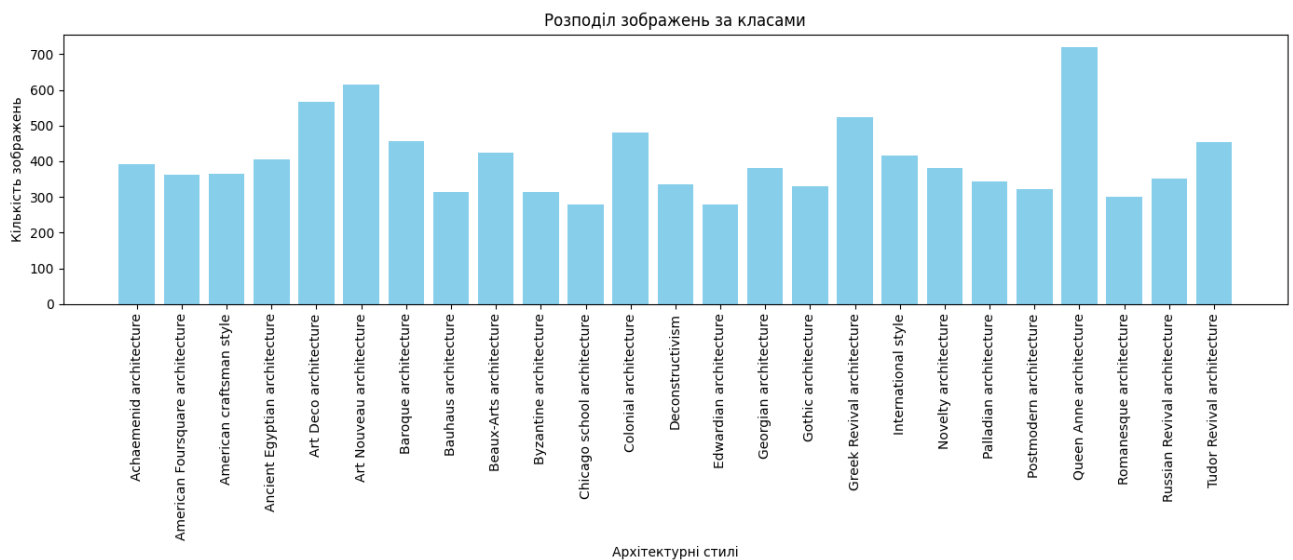


Рисунок 2.6 – Класи датасету та кількість зображень у них

Кількість зразків у датасеті була збалансована до 500 зразків у кожному класі. У процесі аугментації зразки зображень були штучно змінені для того, щоб зробити навчальні дані більш різноманітними. До кожного зображення

застосовували випадкові трансформації. Зображення віддзеркалювались по горизонталі, також змінювалась яскравість. Колірна насиченість і контраст теж випадково змінювались в невеликому діапазоні, що дозволить моделі навчатися незалежно від змін освітлення чи балансу білого. Крім того, модифікувався відтінок кольорів, тобто зображення набували кольорових зсувів. Було штучно збільшено розмір зображень з подальшим випадковим обрізанням назад до початкового формату. Це дозволило зміщувати композицію об'єкта в кадрі, щоб модель при тренуванні не була чутливою до точного положення будівлі.

Також датасет був попередньо оброблений, і за допомогою інструментів Python було сформовано навчальну та тестову вибірки. Таким чином загальна кількість зразків у датасеті після балансування зразків у ньому становить 12500, з них для навчання використовуватиметься 10625, а для тестування 1875 зображень.

Отже, сформовано збалансований датасет, що містить рівну кількість зразків для кожного з 25 класів архітектурних стилів. Кількість зразків у кожному класі була встановлена до 500, що забезпечило рівномірний розподіл даних. Датасет був попередньо оброблений, і для подальшого навчання нейронної мережі були створені навчальна вибірка та тестувальна вибірка.

## **2.7 Проектування інтелектуальної системи визначення архітектурних стилів**

У межах побудови інтелектуальної системи автоматизованого визначення архітектурних стилів важливою є модульна організація, що забезпечує розділення функціональних відповідальностей між підсистемами. Запропонована на рисунку 2.6 схема підсистем реалізує підхід до обробки даних, навчання нейромережевої моделі та взаємодії з користувачем.

Схема, подана на рисунку 2.7, ілюструє архітектуру інтелектуальної системи визначення архітектурного стилю за зображенням. Система складається з 5 взаємопов'язаних підсистем, кожна з яких реалізує окремі етапи повного

циклу обробки даних – від підготовки датасету до отримання класифікаційного результату з відповідним інтерфейсом користувача. Далі розглянуто детально кожен підсистему та її функціональне призначення.

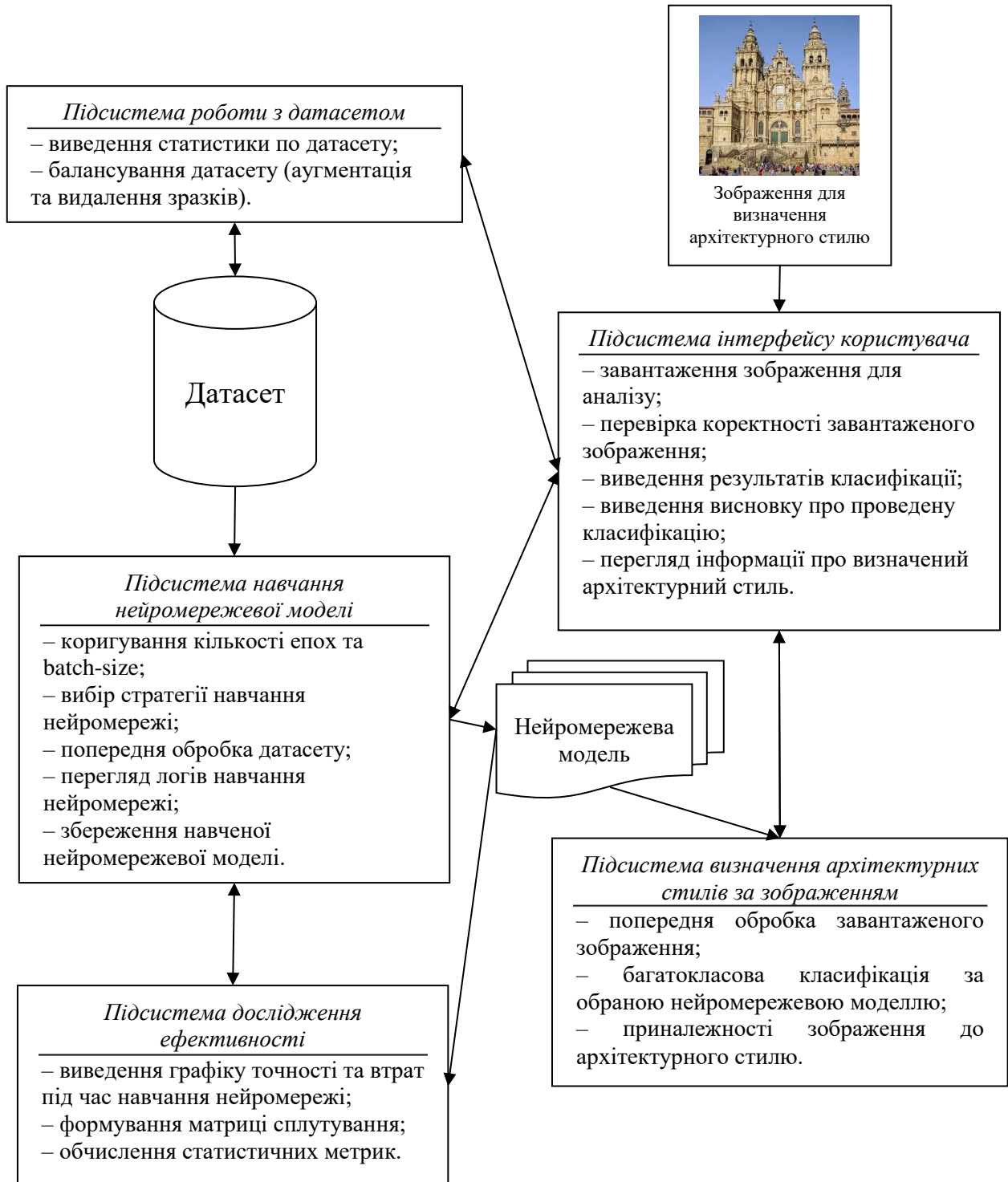


Рисунок 2.7 – Схема підсистем інтелектуальної системи визначення архітектурних стилів

Підсистема роботи з датасетом виконує базові операції з підготовки вхідних даних. Зокрема, вона забезпечує виведення статистичної інформації про датасет, що включає кількість класів, зображень у кожному класі, балансування класів тощо. Особливу увагу приділено процесам аугментації і видалення, що дозволить підвищити загальну якість навчального набору та збільшити його варіативність, зменшуючи ризик перенавчання нейромережевої моделі.

Підсистема навчання нейромережевої моделі відповідає за реалізацію навчального процесу нейромережі. Основними параметрами, що задаються користувачем, є кількість епох, batch size та стратегія навчання (fixed number of epochs або early stopping). У межах цієї підсистеми також виконується попередня обробка зображень, ініціалізація та запуск процесу донавчання (fine-tuning) попередньо натренованої моделі, а також збереження отриманих ваг після завершення тренування.

Підсистема дослідження ефективності призначена для аналізу результатів навчання. Вона забезпечує виведення графіків точності та втрат як для тренувального, так і для валідаційного наборів. Крім того, у межах цієї підсистеми здійснюється побудова матриці помилок, що дозволяє обчислити статистичні метрики: Precision, Recall, F1-міра.

Підсистема інтерфейсу користувача реалізує механізм взаємодії з кінцевим користувачем. Дає змогу завантажувати зображення для класифікації, перевіряти їхню коректність, отримувати результати класифікації, формулювати висновки та переглядати супровідну інформацію про визначений архітектурний стиль, виконує роль фронтенду всієї інтелектуальної системи.

Підсистема визначення архітектурних стилів за зображенням виконує основну задачу багатокласової класифікації. Зокрема, реалізує попередню обробку вхідного зображення, його передавання у вже навчену нейромережеву модель, отримання й обробку передбаченого класу, а також зіставлення результату з відповідним архітектурним стилем.

У спроектованій архітектурі інтелектуальної системи автоматизованого визначення архітектурних стилів, представленої на схемі, зображення, датасет та

нейромережева модель є не лише вхідними або проміжними елементами, а також є результатами роботи відповідних підсистем.

Датасет формується як результат дій підсистеми роботи з датасетом. У межах цієї підсистеми виконується огляд, а також балансування та очищення наявного датасету зображень. Таким чином, датасет є похідною структурою, він одночасно є вхідними даними для підсистеми навчання та вихідними даними з точки зору попередньої обробки.

Нейромережева модель є результатом роботи підсистеми навчання. На основі поданого датасету, вибраних параметрів навчання (кількість епох, batch-size, стратегія завершення навчання) та попередньо ініціалізованої архітектури (у даному випадку – MobileNetV2 із застосуванням fine-tuning), відбувається оптимізація ваг, яка приводить до формування придатної для класифікації моделі. Збережена модель далі використовується підсистемою визначення архітектурного стилю.

Таким чином, спроектована архітектура інтелектуальної системи автоматизованого визначення архітектурних стилів засвідчує доцільність її розподіленої побудови, в якій підсистеми інтегруються через обмін даними, що формуються як результат роботи кожного з етапів роботи інтелектуальної системи. Такий підхід сприятиме гнучкому налаштуванню системи під різні сценарії використання та забезпечує відтворюваність експериментів і достовірність отриманих результатів.

## **2.8 Висновки до розділу 2**

В розділі 2 наведено загальний підхід до визначення архітектурних стилів за зображенням, що базується на використанні глибоких нейронних мереж для розв'язання задачі багатокласової класифікації архітектурних стилів.

Запропоновано метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж, який складається з 3 кроків та дозволяє за

вхідним зображенням за допомогою нейромережевої моделі визначати архітектурний стиль на зображенні.

Наведено кроки для отримання нейромережевої моделі для багатокласової класифікації архітектурних стилів. У пункті наведено схему та описано нейромережеву архітектуру MobileNetV2, що використовуватиметься на кроці 2 запропонованого методу.

Описано датасет, що використовується для навчання та розглянуто кроки його попередньої обробки. Також у розділі спроектовано інтелектуальну систему визначення архітектурних стилів, а саме наведено схему підсистем та їх опис.

Для подальшого досягнення мети роботи виникає потреба програмно реалізувати інтелектуальну систему визначення архітектурних стилів, яка матиме такі функціональні основні можливості:

- попередня обробка датасету;
- навчання нейромережі для визначення архітектурних стилів за зображенням;
- збереження моделей та можливість завантаження існуючої моделі для багатокласової класифікації;
- визначення архітектурних стилів за зображенням;
- експериментальне дослідження навченої нейромережевої моделі.

Для розробки інтелектуальної системи визначення архітектурних стилів необхідно обрати засоби розробки, навести структуру та функціональне призначення, а також взаємозв'язок її програмних складових.

Для експериментального дослідження запропонованого методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж необхідно обчислити за допомогою матриці помилок показники Accuracy, Precision, Recall, F1-mіру.

## **Розділ 3 Програмна реалізація інтелектуальної системи та експериментальне дослідження методу**

### **3.1 Вибір засобів розробки інтелектуальної системи визначення архітектурних стилів**

У процесі програмної реалізації методу автоматичної класифікації архітектурних стилів за зображеннями активно використовуються сучасні бібліотеки для машинного навчання, візуалізації даних та попередньої обробки зображень. Серед них ключовими є `tensorflow.keras` [40], `matplotlib.pyplot` [41], `tensorflow.keras.preprocessing` [42] та `numpy` [43], кожна з яких відіграє важливу роль на різних етапах створення, навчання та аналізу нейронної мережі.

`tensorflow.keras` є високорівневою бібліотекою для побудови моделей глибокого навчання, інтегрованою в екосистему TensorFlow. Вона надає зручний інтерфейс для створення нейронних мереж як у послідовному вигляді (через Sequential API), так і з використанням функціонального підходу. У межах даної роботи `keras` використовується для розробки моделей згорткових нейронних мереж, включаючи попередньо навчені архітектури, такі як MobileNetV2, які дають змогу ефективно вирішувати задачу класифікації зображень архітектурних об'єктів на основі наявних прикладів.

Для візуального аналізу процесу навчання моделей застосовується бібліотека `matplotlib.pyplot`. Вона є складовою частиною пакету Matplotlib і надає засоби для побудови двовимірних графіків у стилі мови MATLAB. Зокрема, у роботі вона використовується для побудови графіків зміни точності та функції втрат на етапах тренування та валідації моделей. Такий підхід дозволяє здійснювати наглядову діагностику ефективності навчання та виявляти потенційні проблеми, пов'язані з перенавчанням або недостатнім узагальненням.

Одним з важливих аспектів підготовки зображень до подачі в нейронну мережу є їх попередня обробка, яка здійснюється за допомогою модулів `tensorflow.keras.preprocessing`. Зокрема, об'єкт `ImageDataGenerator` дозволяє проводити масштабування піксельних значень, а також реалізує засоби

аугментації зображень – обертання, зсув, зміну масштабу тощо. Це сприяє штучному збільшенню обсягу тренувального набору, покращує здатність моделі до узагальнення та дозволяє уникнути перенавчання на обмеженому наборі даних.

Бібліотека `numpy` використовується для перетворення зображень у тензори, нормалізації даних, а також для виконання базових математичних операцій над масивами. Висока швидкодія і широкі можливості цієї бібліотеки роблять її незамінною при обробці великих обсягів візуальної інформації.

Для реалізації ж вебзастосунку використано мікрофреймворк `Flask` та відповідну бібліотеку `Python` [44]. `Flask` – це легкий і гнучкий вебфреймворк для мови програмування `Python`, призначений для створення вебзастосунків і API. Він розроблений з акцентом на простоту використання та розширюваність, що дозволяє розробникам швидко запускати проекти будь-якої складності – від невеликих вебсайтів до масштабних сервісів. `Flask` базується на `WSGI` (`Web Server Gateway Interface`) і не нав'язує строгих структур чи обмежень, залишаючи розробнику свободу вибору архітектури й компонентів.

Таким чином, використання вищезгаданих бібліотек дозволяє створити цілісну інтелектуальну систему для побудови та дослідження глибокої нейронної мережі, здатної до розпізнавання архітектурних стилів за зображенням.

### **3.2 Структура та функціональне призначення програмних складових інтелектуальної системи визначення архітектурних стилів**

Згідно з спроектованою архітектурою інтелектуальної системи визначення архітектурних стилів у п. 2.7 було спроектовано програмний застосунок, що дозволяє провести апробацію запропонованого у роботі методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

Програмна структура інтелектуальної системи визначення архітектурних стилів охоплює кілька ключових компонентів, які забезпечують обробку даних,

навчання моделі, оцінку її ефективності та визначення архітектурних стилів на основі зображень. Нижче наведено діаграму класів (рисунок 3.1) та їх опис.

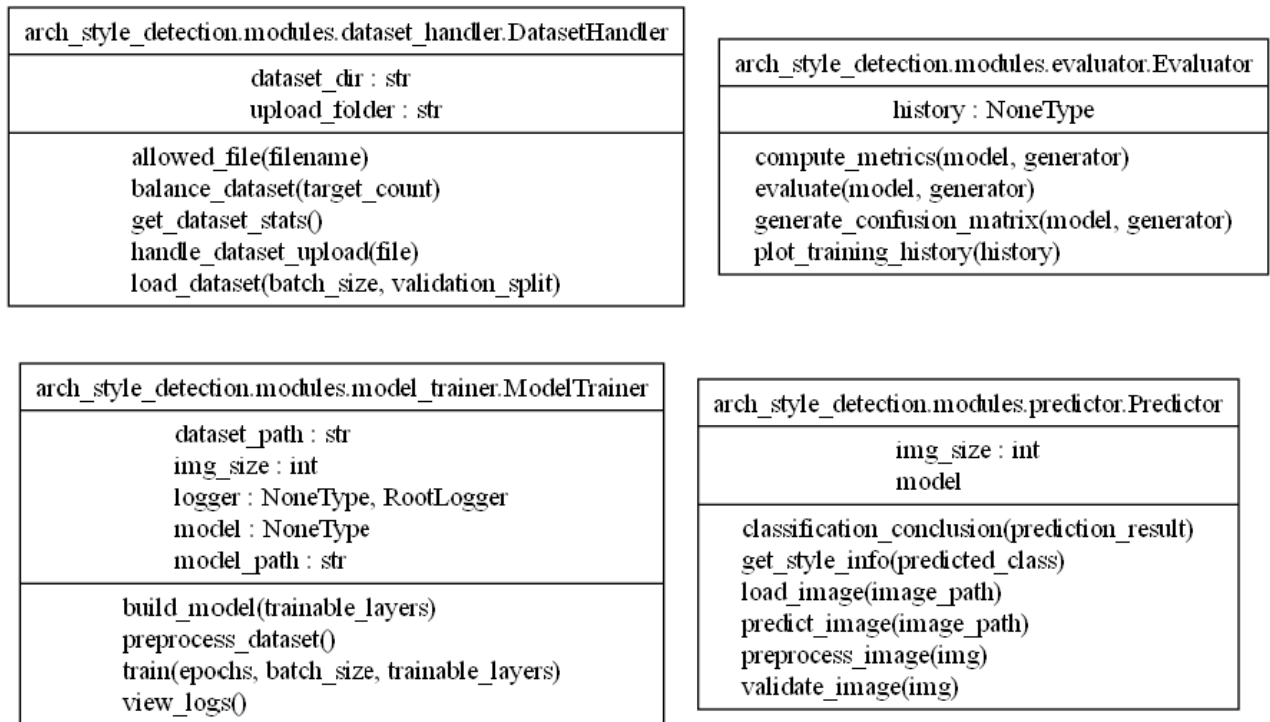


Рисунок 3.1 – Діаграма класів системи автоматизованого визначення архітектурних стилів

Клас DatasetHandle відповідає за валідацію вхідних файлів, визначаючи їхній формат, а також за обробку завантажених зображень, розпаковуючи їх у відповідну директорію. Він здійснює аналіз структури датасету, обчислюючи кількість зображень у кожному класі, загальну кількість зразків та кількість унікальних класів, що дозволяє оцінити розподіл даних. Крім того, DatasetHandler забезпечує балансування датасету, застосовуючи методи аугментації зображень через ImageDataGenerator для класів із недостатньою кількістю прикладів або видаляючи надлишкові зразки, якщо задано цільову кількість.

Клас ModelTrainer реалізує навчання нейронної мережі. Цей клас ініціалізує модель на основі архітектури MobileNetV2, адаптуючи її до задачі класифікації 25 архітектурних стилів. Процес навчання моделі дозволяє обирати

стратегію навчання: при значеннях параметра `trainable_layers`, рівному нулю, усі шари заморожуються для використання як екстрактор ознак, тоді як при додатних значеннях цього параметра активізується донавчання для вказаних шарів. Навчання відбувається з урахуванням заданої кількості епох, розміру батчу та стратегії, після чого модель зберігається у файлі, а логи процесу записуються для подальшого аналізу.

Клас `Evaluator` виконує оцінку ефективності моделі. Цей клас використовує генератори даних для тестування моделі, обчислюючи метрики, такі як `Accuracy`, `precision`, `recall` та `F1-score`, за допомогою `classification_report`. Він генерує матрицю сплутування, що відображає розподіл правильних і неправильних передбачень по класах, а також створює графіки тренувальної та валідаційної точності та втрат на основі історії навчання. Результати візуалізації зберігаються у вигляді зображень, що забезпечує можливість подальшого аналізу продуктивності моделі.

Клас `Predictor`, реалізує функціонал визначення архітектурних стилів. Цей клас завантажує вхідне зображення, перевіряє його коректність за критеріями, такими як мінімальний розмір та формат `RGB`, а потім обробляє його, змінюючи розмір до стандартного значення та нормалізуючи за специфікацією `MobileNetV2`. Використовуючи навчену модель, `Predictor` обчислює ймовірності належності зображення до кожного з 25 класів, визначаючи найімовірніший стиль та рівень впевненості. Крім того, він формулює висновок про найімовірніший клас, залежно від величини впевненості та надає опис відповідного стилю українською мовою, отримуючи його з попередньо визначеного словника. У разі помилок чи некоректних даних клас повертає відповідні повідомлення.

Отже, описані чотири класи функціонують як інтелектуальна система, де `DatasetHandler` готує дані для навчання, `ModelTrainer` формує та навчає модель, `Evaluator` аналізує її ефективність, а `Predictor` забезпечує кінцевий результат для користувача. Така архітектура сприяє модульності, дозволяючи легко

модифікувати чи розширювати окремі компоненти без порушення загальної структури проєкту.

### **3.3 Особливості програмної реалізації інтелектуальної системи визначення архітектурних стилів**

Інтелектуальна система для інтелектуальної системи визначення архітектурних стилів у вигляді вебзастосунку, що забезпечує зручний доступ до її функціоналу через браузер. Ця система, побудована на основі модульної архітектури з використанням фреймворку Flask, використовує методи глибокого навчання для обробки зображень, навчання моделей та оцінки їхньої ефективності, дозволяючи користувачам завантажувати зображення та отримувати детальні результати класифікації в реальному часі.

Файл `templates/index.html` є HTML-шаблоном для веб-інтерфейсу, виконаним із використанням Tailwind CSS для стилізації. Шаблон містить чотири вкладки, реалізовані через JavaScript-функцію `switchTab` для перемикання між ними. Перша вкладка «Визначення архітектурних стилів за зображенням» містить форму для завантаження зображення та відображення результатів передбачення, таких як стиль, впевненість, висновок, опис стилю, зображення та таблиця ймовірностей для всіх класів, із динамічним оновленням через JavaScript. Друга вкладка «Навчання нейромережевої моделі» включає форму для налаштування параметрів навчання, таких як кількість епох, розмір батчу та кількість шарів для `fine-tuning`, а також відображає графік точності та втрат після навчання. Третя вкладка «Робота з датасетом» дозволяє завантажувати ZIP-архів із датасетом, переглядати статистику через кнопку та балансувати датасет через форму із заданою цільовою кількістю зображень на клас, із асинхронним оновленням через JavaScript. Четверта вкладка «Дослідження ефективності» містить форму для оцінки моделі, відображає матрицю сплутування та звіт оцінки. JavaScript-код у шаблоні обробляє асинхронні запити до сервера для

завантаження файлів, передбачення, отримання статистики, балансування та оцінки, динамічно оновлюючи інтерфейс.

Додаткові елементи проєкту включають папку `static/`, яка містить підпапку `uploads/` для завантажених зображень, а також файли `training_history.png` та `confusion_matrix.png`, що генеруються під час оцінки. Папка `models/` використовується для збереження навченої моделі у файлі `mymodel.h5`. Файл `training.log` містить логи процесу навчання моделі. Папка `architectural_styles/` є місцем зберігання датасету, де кожен клас представлений окремою підпапкою із відповідними зображеннями.

Основним файлом програмного застосунку є `main.py`, який слугує точкою входу для Flask-застосунку. У ньому ініціалізуються основні компоненти системи та визначаються маршрути для веб-інтерфейсу. Спочатку створюється Flask-застосунок за допомогою `app = Flask(__name__)`. Далі ініціалізуються об'єкти для роботи з датасетом, моделлю, оцінкою та передбаченням: `dataset_handler` створюється через клас `DatasetHandler` із параметрами шляху до датасету та папки для завантажень, `model_trainer` через клас `ModelTrainer` із шляхом до датасету та моделі, `evaluator` через клас `Evaluator` для оцінки, а `predictor` через клас `Predictor` із шляхом до моделі.

Маршрути в `main.py` забезпечують взаємодію з користувачем: маршрут `/` відображає головну сторінку через шаблон `index.html`, маршрут `/predict` обробляє POST-запити для передбачення стилю зображення та повертає JSON із результатами, такими як передбачений стиль, впевненість, висновок, опис стилю, ймовірності всіх класів та шлях до зображення. Маршрут `/train` обробляє POST-запити для навчання моделі, приймаючи параметри, такі як кількість епох, розмір батчу та кількість шарів для `fine-tuning`, після чого повертає сторінку з повідомленням про успішне навчання та метриками. Маршрут `/upload_dataset` дозволяє завантажувати датасет у форматі ZIP, розпаковує його та повертає сторінку з підтвердженням або помилкою. Маршрут `/dataset_stats` повертає статистику датасету у форматі JSON, а маршрут `/balance_dataset` балансує датасет на основі заданої цільової кількості зображень на клас, повертаючи

оновлену статистику. Нарешті, маршрут /evaluate виконує оцінку моделі, генерує графіки та повертає сторінку з результатами оцінки.

Файл modules/predictor.py відповідає за передбачення архітектурного стилю через клас Predictor, відповідно сторінка вебсайту з реалізованим функціоналом наведено на рисунку 3.2.

## Інтелектуальна система визначення архітектурних стилів

1. Визначення архітектурних стилів за зображенням

2. Навчання нейромережевої моделі

3. Робота з датасетом

4. Дослідження ефективності

---

### 1. Розпізнавання архітектурного стилю

Завантажте фотографію

Вибрати файл 0a9e6988-7f18-11ef-be2f-eeffbe1de8a3.1220x600.jpeg

Визначити стиль

#### Результати класифікації

Найбільш ймовірний стиль: Baroque architecture  
 Ймовірність: 87.37%  
 Зображення класифіковано як стиль 'Baroque architecture' з впевненістю 87.37%.

Опис стилю: Європейський стиль 17–18 століть із драматичними, оздобленими дизайнами, вигнутими формами та величчю, наприклад, у Базилиці Святого Петра.




Рисунок 3.2 – Сторінка визначення архітектурного стилю за зображенням

У методі ініціалізації `__init__` завантажуються навчена модель із заданого шляху та встановлюється розмір зображень. Метод `load_image` завантажує

зображення через `Image.open`, конвертуючи його у формат RGB. Метод `validate_image` перевіряє коректність зображення, зокрема його наявність, розмір та формат RGB, повертаючи результат перевірки та повідомлення про помилку, якщо перевірка не пройдена. Метод `preprocess_image` обробляє зображення для передбачення: змінює розмір, конвертує у масив, застосовує нормалізацію через `preprocess_input` для `MobileNetV2` та додає розмірність батчу. Метод `predict_image` виконує передбачення стилю: завантажує зображення, перевіряє його, обробляє та передає моделі, після чого повертає словник із передбаченим стилем, впевненістю та ймовірностями для всіх класів. Метод `get_style_info` повертає опис стилю із словника `ARCHITECTURE_INFO`, використовуючи ключ, отриманий шляхом заміни пробілів на підкреслення у назві передбаченого класу; якщо опис відсутній, повертається повідомлення «Інформація про цей стиль відсутня». У файлі також визначені дві константи: `ARCHITECTURE_CLASSES` – список із 25 назв архітектурних стилів із пробілами, та `ARCHITECTURE_INFO` – словник із описами стилів українською мовою.

Файл `modules/model_trainer.py` відповідає за створення, навчання та збереження моделі через клас `ModelTrainer`, світлину екрану з даним функціоналом вебзастосунку показано на рисунку 3.3.

У методі ініціалізації `__init__` задаються шлях до датасету, шлях для збереження моделі, розмір зображень, а також налаштовується логування, яке записується у файл `training.log`. Метод `preprocess_dataset` готує датасет для навчання та валідації за допомогою `ImageDataGenerator`, застосовуючи аугментацію, таку як повороти, зміщення, масштабування та горизонтальне віддзеркалення, із розбиттям на тренувальну та валідаційну вибірки.

Метод `build_model` створює модель на основі `MobileNetV2` із можливістю вибору стратегії навчання: якщо параметр `trainable_layers` дорівнює 0, усі шари заморожуються для використання `feature extraction`, а якщо більше 0, останні `trainable_layers` шарів стають навчальними для `fine-tuning`; до базової моделі додаються шари `GlobalAveragePooling2D` та `Dense` для класифікації на 25 класів.

Метод `train` виконує навчання моделі із заданими параметрами, такими як кількість епох, розмір батчу та кількість навчальних шарів, після чого зберігає модель за вказаним шляхом. Метод `view_logs` виводить логи навчання, читаючи їх із файлу `training.log`.



Рисунок 3.3 – Сторінка з навчання нейромережевої моделі

Наступним важливим компонентом є файл `modules/dataset_handler.py`, який містить клас `DatasetHandler` для обробки датасету, вигляд сторінки показано на рисунку 3.3.

## Інтелектуальна система визначення архітектурних стилів

1. Визначення архітектурних стилів за зображенням

2. Навчання нейромережевої моделі

3. Робота з датасетом

4. Дослідження ефективності

### 3. Робота з датасетом

Завантажити датасет (ZIP)

Вибрати файл arch\_dataset.zip

Завантажити

Отримати статистику

Розподіл зображень за класами

Архітектурний стиль	Кількість зображень
Achitectural architecture	400
American Gothic architecture	380
American craftsman style	380
Art Deco architecture	550
Art Nouveau architecture	620
Baroque architecture	450
Beaux-Arts architecture	320
Byzantine architecture	300
Chicago school architecture	480
Colonial architecture	350
Decorativism	280
Edwardian architecture	380
Georgian architecture	350
Gothic architecture	520
Greek Revival architecture	420
International style	380
Neoclassical architecture	350
Neoclassical architecture	350
Polynesian architecture	350
Queen Anne architecture	700
Romanesque architecture	320
Russian Revival architecture	350
Tudor Revival architecture	450

Цільова кількість зображень на клас (залиште порожнім для середнього)

400

Балансувати

Рисунок 3.3 – Сторінка роботи з датасетом

У методі ініціалізації `__init__` задаються шляхи до папки датасету та завантажень, а також створюються необхідні директорії. Метод `allowed_file` перевіряє, чи має файл дозволене розширення, таке як `.png`, `.jpg` або `.jpeg`. Метод `handle_dataset_upload` обробляє завантаження файлу: якщо це зображення або ZIP-архів, він зберігається, а ZIP розпаковується в папку датасету. Метод `get_dataset_stats` обчислює статистику датасету, повертаючи словник із кількістю зображень у кожному класі, загальною кількістю зображень та кількістю класів. Метод `balance_dataset` балансує датасет, використовуючи аугментацію через `ImageDataGenerator` для класів із недостатньою кількістю зображень або видаляючи зайві зображення, якщо їх забагато, із можливістю задавати цільову кількість зображень на клас. Метод `load_dataset` завантажує датасет для навчання

та валідації через `image_dataset_from_directory`, розбиваючи його на тренувальну та валідаційну вибірки із заданим розміром батчу та співвідношенням розбиття.

Файл `modules/evaluator.py` містить клас `Evaluator` для оцінки моделі, світлина екрану сторінки для обчислення метрик подано на рисунку 3.4.



Рисунок 3.4 – Сторінка обчислення метрик навченої моделі

У методі ініціалізації `__init__` створюється порожня змінна `history` для зберігання історії навчання. Метод `evaluate` оцінює модель на основі генератора даних, повертаючи звіт класифікації через `classification_report` та матрицю

сплутування через `confusion_matrix`. Метод `plot_training_history` створює графіки точності та втрат для тренувальних і валідаційних даних, зберігаючи їх у файлі `static/training_history.png`. Метод `generate_confusion_matrix` генерує матрицю помилок, візуалізує її за допомогою `matplotlib` та зберігає як `static/confusion_matrix.png`. Метод `compute_metrics` обчислює метрики, такі як загальна точність, середні значення `precision`, `recall` та F1-міри, використовуючи `classification_report`, і повертає їх у структурованому форматі.

Таким чином, розроблений програмний застосунок забезпечує повний цикл роботи із класифікації архітектурних стилів: від обробки та балансування даних, створення та навчання моделі, її оцінки до передбачення стилів із детальними результатами, представленими через веб-інтерфейс. Кожен компонент чітко визначений і виконує свою роль, забезпечуючи модульність та можливість підтримки, адже відповідає попередньо спроектованій архітектурі.

### 3.4 Експериментальне дослідження

Згідно наведених у п. 2.4 метрик для оцінювання ефективності нейромережових моделей проведено дослідження розробленого методу. Було проведено експериментальне дослідження навчання моделі за різної кількості епох, `batch-size` та стратегій навчання, результати якого наведено у таблиці 3.1.

Таблиця 3.1 – Залежність точності класифікації від кількості епох, `batch-size` та стратегії навчання

Епохи	<code>batch-size</code>	Accuracy, %	Стратегія навчання
30	32	72	fixed number of epochs
50	16	65	fixed number of epochs
20	32	78	early stopping
26	16	80	early stopping

Згідно таблиці найгірший результат у точності класифікації (65%) спостерігається при використанні фіксованого числа епох (50) з малим розміром batch-size (16). Натомість використання більшого batch-size (32) при 30 епохах покращує точність до 72%, що свідчить про збільшення точності.

Суттєве зростання точності (до 78%) спостерігається при застосуванні стратегії навчання early stopping із розміром batch-size 32, що свідчить про те, що автоматичне припинення навчання у момент втрати прогресу на валідації дозволяє уникнути перенавчання та сприяє збереженню оптимальних параметрів моделі. Найвищу точність (80%) зафіксовано при поєднанні малого batch-size (16) з early stopping.

На рисунку 3.5 наведено графіки динаміки точності та втрат при навчанні моделі MobileNetV2, що показала кращий результат згідно таблиці 3.1.

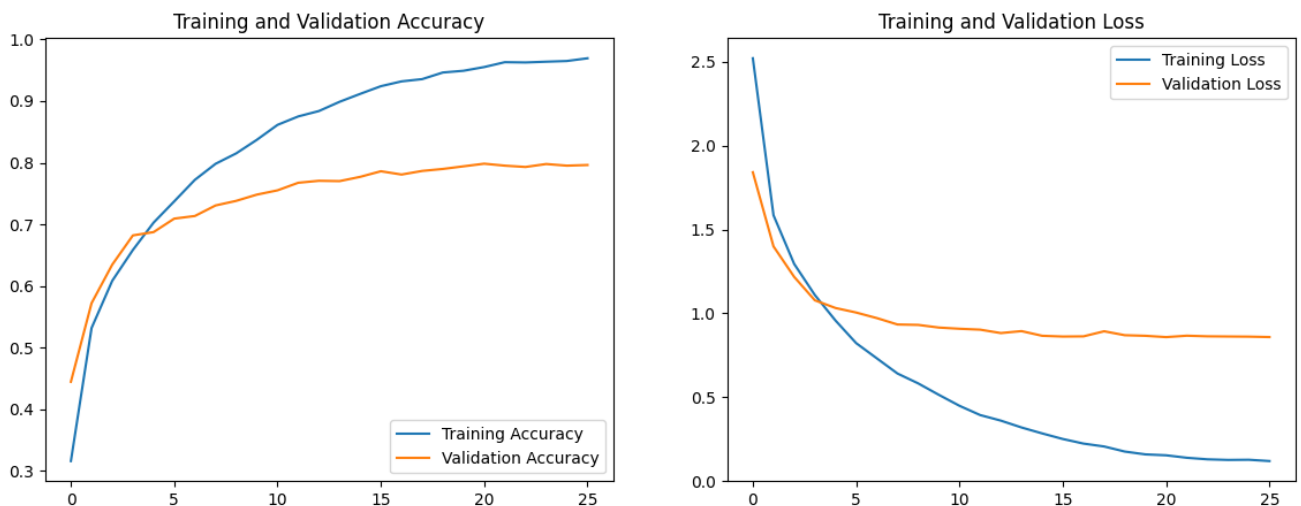


Рисунок 3.5 – Графіки точності та втрат під час навчання нейромережевої моделі MobileNetV2

Наведені графіки навчання нейронної мережі, побудованої на основі архітектури MobileNetV2, свідчать про стабільний і поступовий процес оптимізації моделі впродовж перших 22 епох навчання. Модель використовує попередньо натреновану базу MobileNetV2 (завантажену без верхнього класифікаційного шару) з додаванням шару глобального усереднення,

нормалізації, дропауту та щільного шару з 25 виходами, що відповідає кількості класів задачі класифікації.

У першій епосі модель демонструє низьку точність (accuracy  $\sim 21\%$ ) і високе значення функції втрат (loss  $\sim 3\%$ ), що цілком очікувано, враховуючи початкову ініціалізацію верхніх шарів. Проте вже з другої епохи відзначається стрімке зростання точності та зниження втрат як на тренувальному, так і на валідаційному наборах: наприклад, у 5-й епосі точність сягає 71%, а функція втрат знижується до 92%. Така динаміка вказує на ефективне навчання класифікаційного шару на новому датасеті при заморожених шарах базової моделі.

Подальші епохи демонструють помірне, але стабільне зростання точності: вже на 15-й епосі точність перевищує 91% на тренувальному наборі, а на валідаційному – 77%, при цьому втрати знижуються. Темп зростання показників зменшується, що узгоджується з плавною кривою навчання і свідчить про поступове навчання моделі на датасеті.

Далі на рисунку 3.6 наведено матрицю сплутувань для багатокласової класифікації архітектурних стилів, що була отримана при навчанні моделі при 26 епохах з розміром batch-size 16 та стратегією навчання early stopping.

Матриця сплутувань, отримана в результаті оцінювання навченої нейромережевої моделі для багатокласової класифікації зображень архітектурних стилів, дозволяє здійснити детальний аналіз помилок моделі та простежити характер плутанини між окремими класами.

Проведений аналіз результатів ґрунтується на 25 класах, кожен із яких представлений 75 зразками, що забезпечує збалансованість вибірки та дозволяє об'єктивно оцінити ефективність використаної нейромережевої моделі в умовах рівномірного розподілу.



Таблиця 3.2 – Показники Precision, Recall, F1-міра для багатокласової класифікації архітектурних стилів

Архітектурний стиль	Precision, %	Recall, %	F1-міра, %	Support
Achaemenid architecture	97	91	94	75
American Foursquare architecture	79	87	83	75
American craftsman style	81	75	78	75
Ancient Egyptian architecture	96	96	96	75
Art Deco architecture	71	77	74	75
Art Nouveau architecture	77	72	74	75
Baroque architecture	69	68	68	75
Bauhaus architecture	77	85	81	75
Beaux-Arts architecture	62	71	66	75
Byzantine architecture	83	85	84	75
Chicago school architecture	92	89	91	75
Colonial architecture	63	51	56	75
Deconstructivism	88	88	88	75
Edwardian architecture	75	80	77	75
Georgian architecture	76	71	73	75
Gothic architecture	96	87	91	75
Greek Revival architecture	75	72	73	75
International style	79	65	72	75
Novelty architecture	87	92	90	75
Palladian architecture	78	83	80	75
Postmodern architecture	81	80	81	75
Queen Anne architecture	69	68	68	75
Romanesque architecture	85	85	85	75
Russian Revival architecture	82	93	88	75
Tudor Revival architecture	80	85	83	75
<b>Accuracy</b>			<b>80</b>	<b>1875</b>
<b>Macro avg</b>	<b>80</b>	<b>80</b>	<b>80</b>	<b>1875</b>
<b>Weighted avg</b>	<b>80</b>	<b>80</b>	<b>80</b>	<b>1875</b>

Натомість спостерігаються певні регулярні патерни помилкових класифікацій, які вказують на стилістичну або візуальну подібність між окремими архітектурними стилями. Зокрема, навчена нейромережева модель часто плутає Art Nouveau architecture з Art Deco architecture (6 випадків) та навпаки, що є логічним з огляду на спільне естетичне походження обох напрямів

на межі XIX–XX століть. Аналогічно, значна кількість зразків Beaux-Arts architecture помилково віднесена до Baroque, Edwardian та Georgian architecture, що свідчить про труднощі в розмежуванні класицизованих стилів з подібною орнаментальністю, симетрією та композиційною структурою.

Крім того, Colonial architecture демонструє підвищену частоту сплутування з Queen Anne (6 випадків) та Georgian стилями (5 випадків), що може бути зумовлено історичною спорідненістю, зокрема, трансформацією британських архітектурних канонів у північноамериканському контексті. Ще одним прикладом є часте сплутування International style з Postmodern architecture (10 випадків), яке вказує на труднощі моделі в розпізнаванні функціоналістських елементів у контексті візуально схожої, але концептуально протилежної постмодерністської естетики. Приклади будівель у цих стилях наведено на рисунку 3.7.



а)



б)

Рисунок 3.7 – Приклади будівель архітектурних стилів:

а) International style; б) Postmodern architecture

Підсумовуючи, матриця плутанини засвідчує, що модель загалом досягає прийняттого рівня класифікації, однак існує системна проблема розрізнення візуально або стилістично близьких класів, як на рисунку 3.7.

Загальна точність моделі становить 80%, що є прийнятним результатом з огляду на велику кількість класів та високу міжкласову схожість у візуальному

представленні архітектурних стилів. Показники середніх значень (macro avg і weighted avg) перебувають на рівні 80% для всіх трьох метрик, що свідчить про відносно збалансовану ефективність моделі по всьому набору даних.

Найвищі значення F1-міри спостерігаються для класів Ancient Egyptian architecture (96%), Gothic architecture (91%), Chicago school architecture (91%), Novelty architecture (90%).

Водночас низькі значення F1-міри зафіксовано для Beaux-Arts architecture (66%), Colonial architecture (56%), Queen Anne architecture (68%), Baroque architecture (68%) та Art Deco architecture (74%). Зокрема, для Colonial architecture повнота становить лише 51%. Подібна ситуація є наслідком естетичної близькості до інших стилів, таких як Georgian, Edwardian або Queen Anne, що спричиняє сплутування (підтверджено матрицею сплутувань на рисунку 3.6).

Помітна також розбіжність між точністю та повнотою для окремих класів. Наприклад, American Foursquare architecture має високу повноту (87%) за відносно помірної точності (79%), що вказує на тенденцію моделі класифікувати багато зразків до цього класу, зокрема й хибно. Протилежна ситуація спостерігається для International style, де точність перевищує повноту (79% проти 65%), що свідчить про обережність моделі при класифікації класу.

Отримані результати багатокласової класифікації архітектурних стилів за зображенням перевершують відомі підходи, наприклад, [34, 36] на 6% та 14%, а також є майже на рівні з результатами, що отримані у [37, 38] у яких точність класифікації склала 80,12% та 84,66% (для 5-и та 3-х класів архітектурних стилів відповідно). Проте варто зазначити, що запропонований у роботі підхід показує точність 80% при класифікації 25 класів архітектурних стилів, в той час у наведених дослідженнях [37, 38] кількість архітектурних стилів є меншою.

Отже, результати експериментів підтвердили ефективність використання MobileNetV2 із fine-tuning: найвищу точність класифікації 80% отримано при конфігурації з розміром batch-size 16 та застосуванням стратегії early stopping, що забезпечила оптимальний баланс між перенавчанням і узагальненням. А запропонований у роботі метод визначення архітектурних стилів за зображенням

засобами глибоких нейронних мереж, що базується на використанні попередньо навченої моделі MobileNetV2 (до якої було застосовано метод навчання *fine-tuning*) дозволив досягти підвищення точності автоматизованого визначення архітектурних стилів за зображенням.

### 3.5 Висновки до розділу 3

У розділі виконано програмну реалізацію розробленої інтелектуальної системи визначення архітектурних стилів. Застосування сучасних програмних бібліотек забезпечило побудову функціональної архітектури веб-застосунку, в якому чітко визначені модулі *DatasetHandler*, *ModelTrainer*, *Evaluator* та *Predictor*, що разом формують гнучку та масштабовану інтелектуальну систему. Особливості реалізації яких було описано у відповідному пункті.

Наведено метрики, за якими здійснено оцінювання нейромережевої моделі – Accuracy, Precision, Recall, F1-міра, а також матрицю сплутувань, що дозволили здійснити оцінку ефективності навченої моделі при класифікації архітектурних стилів за зображенням.

Окрему увагу в розділі було приділено експериментальній частині, результати якої підтверджують доцільність використання попередньо навченої моделі MobileNetV2 з подальшим тонким налаштуванням. У процесі експериментального дослідження встановлено, що оптимальними умовами для навчання є *batch size* 16 та застосуванням стратегії *early stopping*, що дозволяє уникнути перенавчання моделі й забезпечити її здатність до узагальнення.

У підсумку, модель досягла точності класифікації 80%, що є високим показником з огляду на складність задачі й варіативність зображень архітектурних об'єктів. А, отже, запропонований у роботі метод дозволяє досягнути поставленої мети – підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

## Загальні висновки

Метою кваліфікаційної роботи бакалавра було підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж. Для досягнення поставленої мети виконано такі завдання:

- проведено аналіз предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням;
- розроблено метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж;
- здійснено проєктування та програмну реалізацію спроектованої інтелектуальної системи на основі запропонованого методу;
- виконано експериментальне дослідження ефективності роботи методу з використанням розробленої інтелектуальної системи визначення архітектурних стилів.

На основі розробленого методу виконано відповідну програмну реалізацію у вигляді інтелектуальної вебсистеми, яка використовує нейромережу MobileNetV2 із fine-tuning, точність класифікації якої 80%. Розроблена інтелектуальна система виконує наступні функції:

- попередня обробка датасету;
- навчання нейромережі для визначення архітектурних стилів за зображенням;
- збереження моделей та можливість завантаження існуючої моделі для багатокласової класифікації;
- визначення архітектурних стилів за зображенням;
- експериментальне дослідження навченої нейромережевої моделі.

Проведене дослідження демонструє, що використання глибоких згорткових нейронних мереж, зокрема MobileNetV2 із fine-tuning, є доцільним та ефективним для вирішення задачі класифікації архітектурних стилів за зображеннями, адже запропонований метод дозволив досягти точності

класифікації 80% для 25 архітектурних стилів, проте, отримані результати мікрометрик свідчать також про можливість подальшого підвищення точності моделі в класифікації для окремих класів.

Перспективи та ступінь впровадження розробленого методу пов'язані з можливістю його адаптації до задач цифрової культурної спадщини, аналітики в архітектурознавстві, освітніх проєктів та інтеграції для мобільної платформи з метою швидкого визначення архітектурного стилю.

Покращення методу можливе за рахунок використання більших та більш різноманітних датасетів для підвищення точності класифікації, а також впровадження механізмів пояснюваності для підвищення прозорості рішень нейромережевої моделі.

## Перелік посилань

1. The Beauty of Architectural Structures : Elegance beyond Aesthetics - Arch2O.com. *Arch2O.com*. URL: <https://www.arch2o.com/the-beauty-of-architectural-structures/> (date of access: 29.04.2025).
2. Brozovsky J., Labonnote N., Vigren O. Digital technologies in architecture, engineering, and construction. *Automation in Construction*. 2024. Vol. 158. P. 105212. URL: <https://doi.org/10.1016/j.autcon.2023.105212> (date of access: 29.04.2025).
3. A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends / P. Sun et al. *IEEE Internet of Things Journal*. 2024. P. 1. URL: <https://doi.org/10.1109/jiot.2024.3372518> (date of access: 29.04.2025).
4. Xie K., Zhang Y., Han W. Architectural Heritage Preservation for Rural Revitalization: Typical Case of Traditional Village Retrofitting in China. *Sustainability*. 2024. Vol. 16, no. 2. P. 681. URL: <https://doi.org/10.3390/su16020681> (date of access: 29.04.2025).
5. Revolutionizing cultural heritage preservation: an innovative IoT-based framework for protecting historical buildings / M. Casillo et al. *Evolutionary Intelligence*. 2024. URL: <https://doi.org/10.1007/s12065-024-00959-y> (date of access: 29.04.2025).
6. Malewczyk M., Taraszkiewicz A., Czyż P. Visual Perception of Regularity and the Composition Pattern Type of the Facade. *Buildings*. 2024. Vol. 14, no. 5. P. 1389. URL: <https://doi.org/10.3390/buildings14051389> (date of access: 29.04.2025).
7. Baper S. Y. Can Architectural Identity Be Measured?. *Buildings*. 2024. Vol. 14, no. 5. P. 1379. URL: <https://doi.org/10.3390/buildings14051379> (date of access: 29.04.2025).
8. A. Salingaros N. Architectural Knowledge: Lacking A Knowledge System, The Profession Rejects Healing Environments That Promote Health And Well-Being. *New Design Ideas*. 2024. Vol. 8, no. 2. P. 261–299. URL: <https://doi.org/10.62476/ndi82261> (date of access: 29.04.2025).

9. Faisca A. R., Gomes Januário P. An Analysis of Knowledge Production in Architectural Graphic Ideation Applied to Railway Architecture. *Graphic Horizons*. Cham, 2024. P. 128–137. URL: [https://doi.org/10.1007/978-3-031-57579-2\\_16](https://doi.org/10.1007/978-3-031-57579-2_16) (date of access: 29.04.2025).
10. Surogina O. Knowledge about transitions in architectural practices. 2024. URL: <http://resolver.tudelft.nl/uuid:fbd80029-7304-4e50-91ea-52938bb78083> (date of access: 29.04.2025).
11. Cao Z., Mustafa M., Mohd Isa M. H. The role of artistic quality in a heritage architectural style in modulating tourist interest and aesthetic pleasure: a case study of Hui-style architecture in the Hongcun Scenic Area, China. *Journal of Heritage Tourism*. 2024. P. 1–23. URL: <https://doi.org/10.1080/1743873x.2024.2378805> (date of access: 29.04.2025).
12. Wang J., Atipattayakul C., Sengna K. The Relationship Between Architectural Design Style and Architectural Culture. *Architecture Engineering and Science*. 2024. Vol. 5, no. 2. P. 100. URL: <https://doi.org/10.32629/aes.v5i2.2278> (date of access: 29.04.2025).
13. Zwead A. A. H. Modern Architectural Style and Its Impact on Tourism Activity (Najaf Governorate is A Model). *European Journal of Research Development and Sustainability*. 2023. Vol. 4, no. 10. P. 130–145.
14. Vukanović M., Martinović D., Jokić B. Cross-domain and intersectoral cooperation for developing arts and urban development case study: Architecture and fine arts. *Kultura*. 2024. No. 182-183. P. 245–268. URL: <https://doi.org/10.5937/kultura2483245v> (date of access: 29.04.2025).
15. Arouet F. M., Scott G. G. Architecture, Antiquarianism, and Styles.
16. Farooq N. Gothic whispers: architecture in the age of darkness. *Kashf Journal of Multidisciplinary Research*. 2024. Vol. 1, no. 05. P. 25–35. URL: <https://kjmr.com.pk/kjmr/article/view/94> (date of access: 29.04.2025).
17. Basriddin K. The Development Of Architecture And Fine Arts In The Renaissance Period In Europe. *Western European Journal of Linguistics and Education*. 2024. Vol. 2, no. 3. P. 94–99.

18. Recep M., Purde A., Muci M. K. Baroque Architecture And Its Contribution To European Cultural Identity. *International Scientific Journal Vision*. 2023. P. 79–94.
19. Architectural styles of curiosity in global Wikipedia mobile app readership / D. Zhou et al. *Science Advances*. 2024. Vol. 10, no. 43. URL: <https://doi.org/10.1126/sciadv.adn3268> (date of access: 29.04.2025).
20. Generative Deep Learning for Visual Animation in Landscapes Design / P. Ardhianto et al. *Scientific Programming*. 2023. Vol. 2023. P. 1–12. URL: <https://doi.org/10.1155/2023/9443704> (date of access: 29.04.2025).
21. Sadikhova S. Contrasting Splendor: A Comparative Study of Baroque and Classical Styles in Visual Arts. *Acta Globalis Humanitatis et Linguarum*. 2024. Vol. 1, no. 2. P. 144–153. URL: <https://doi.org/10.69760/aghel.01024076> (date of access: 29.04.2025).
22. Tandon M., Bano F. Comparative Critical Analysis of Modern Architectural Styles. *Indian J. Nat. Sci.* 2023. Vol. 14. P. 57095–57106
23. Using Artificial Intelligence to Generate Master-Quality Architectural Designs from Text Descriptions / J. Chen et al. *Buildings*. 2023. Vol. 13, no. 9. P. 2285. URL: <https://doi.org/10.3390/buildings13092285> (date of access: 29.04.2025).
24. Zhang Z., Fort J. M., Giménez Mateu L. Exploring the Potential of Artificial Intelligence as a Tool for Architectural Design: A Perception Study Using Gaudí's Works. *Buildings*. 2023. Vol. 13, no. 7. P. 1863. URL: <https://doi.org/10.3390/buildings13071863> (date of access: 29.04.2025).
25. Ibrahim V., Eltanbouly M. Architecture and Sustainability: Case Studies from Cairo's Downtown and Nubia-Aswan Towards Achieving the UN Sustainable Development Goals. *JES. Journal of Engineering Sciences*. 2024. P. 0. URL: <https://doi.org/10.21608/jesaun.2024.251635.1291> (date of access: 29.04.2025).
26. Archana R., Jeevaraj P. S. E. Deep learning models for digital image processing: a review. *Artificial Intelligence Review*. 2024. Vol. 57, no. 1. URL: <https://doi.org/10.1007/s10462-023-10631-z> (date of access: 29.04.2025).

27. Tummala M. Image Classification Using Convolutional Neural Networks. *International Journal of Scientific and Research Publications (IJSRP)*. 2019. Vol. 9, no. 8. P. p9261. URL: <https://doi.org/10.29322/ijsrp.9.08.2019.p9261> (date of access: 29.04.2025).
28. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations / Z. Zhao et al. *Expert Systems with Applications*. 2024. Vol. 242. P. 122807. URL: <https://doi.org/10.1016/j.eswa.2023.122807> (date of access: 29.04.2025).
29. Fayyad M. F., Mustakim. Application of AlexNet, EfficientNetV2B0, and VGG19 with Explainable AI for Cataract and Glaucoma Image Classification. *2024 International Electronics Symposium (IES)*, Denpasar, Indonesia, 6–8 August 2024. 2024. P. 406–412. URL: <https://doi.org/10.1109/ies63037.2024.10665856> (date of access: 29.04.2025).
30. Image classification based on improved VGG network / M. Xie et al. *5th International Conference on Computer Vision and Data Mining (ICCVDM 2024)*, Changchun, China, 19–21 July 2024 / ed. by X. Zhang, M. Yin. 2024. P. 2. URL: <https://doi.org/10.1117/12.3048051> (date of access: 29.04.2025).
31. Liu Y., Wang W., He W. A Study on an Improved ResNet Model for Image Classification. *2024 17th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Shanghai, China, 26–28 October 2024. 2024. P. 1–6. URL: <https://doi.org/10.1109/cisp-bmei64163.2024.10906260> (date of access: 29.04.2025).
32. Riyadi S., Abidin F. A., Audita N. Comparison of ResNet50V2 and MobileNetV2 Models in Building Architectural Style Classification. *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, Fez, Morocco, 8–10 May 2024. 2024. P. 1–8. URL: <https://doi.org/10.1109/iscv60512.2024.10620099> (date of access: 25.05.2025).
33. Advanced Image Classification on Intel Datasets Using Optimized EfficientNet and MobileNetV2 / S. Vats et al. *2024 IEEE 9th International Conference*

for *Convergence in Technology (I2CT)*, Pune, India, 5–7 April 2024. 2024. URL: <https://doi.org/10.1109/i2ct61223.2024.10543649> (date of access: 29.04.2025).

34. Siountri K., Anagnostopoulos C.-N. The Classification of Cultural Heritage Buildings in Athens Using Deep Learning Techniques. *Heritage*. 2023. Vol. 6, no. 4. P. 3673–3705. URL: <https://doi.org/10.3390/heritage6040195> (date of access: 29.04.2025).

35. Architectural style classification based on CNN and channel–spatial attention / B. Wang et al. *Signal, Image and Video Processing*. 2022. URL: <https://doi.org/10.1007/s11760-022-02208-0> (date of access: 29.04.2025).

36. Zeng, Z., Goo, J. M., Wang, X., Chi, B., Wang, M., & Boehm, J. (2024). Zero-Shot Building Age Classification from Facade Image Using GPT-4. arXiv preprint arXiv:2404.09921.

37. Classification of the qilou (arcade building) using a robust image processing framework based on the Faster R-CNN with ResNet50 / M. H. Li et al. *Journal of Asian Architecture and Building Engineering*. 2023. URL: <https://doi.org/10.1080/13467581.2023.2238038> (date of access: 29.04.2025).

38. Cantemir E., Kandemir O. Use of artificial neural networks in architecture: determining the architectural style of a building with a convolutional neural networks. *Neural Computing and Applications*. 2024. URL: <https://doi.org/10.1007/s00521-023-09395-y> (date of access: 29.04.2025).

39. Kaggle: Architectural Styles Dataset | Kaggle. URL: <https://www.kaggle.com/datasets/dumitru/architectural-styles-dataset> (date of access: 28.05.2025).

40. Keras: The high-level API for TensorFlow | TensorFlow Core. *TensorFlow*. URL: <https://www.tensorflow.org/guide/keras> (date of access: 27.05.2025).

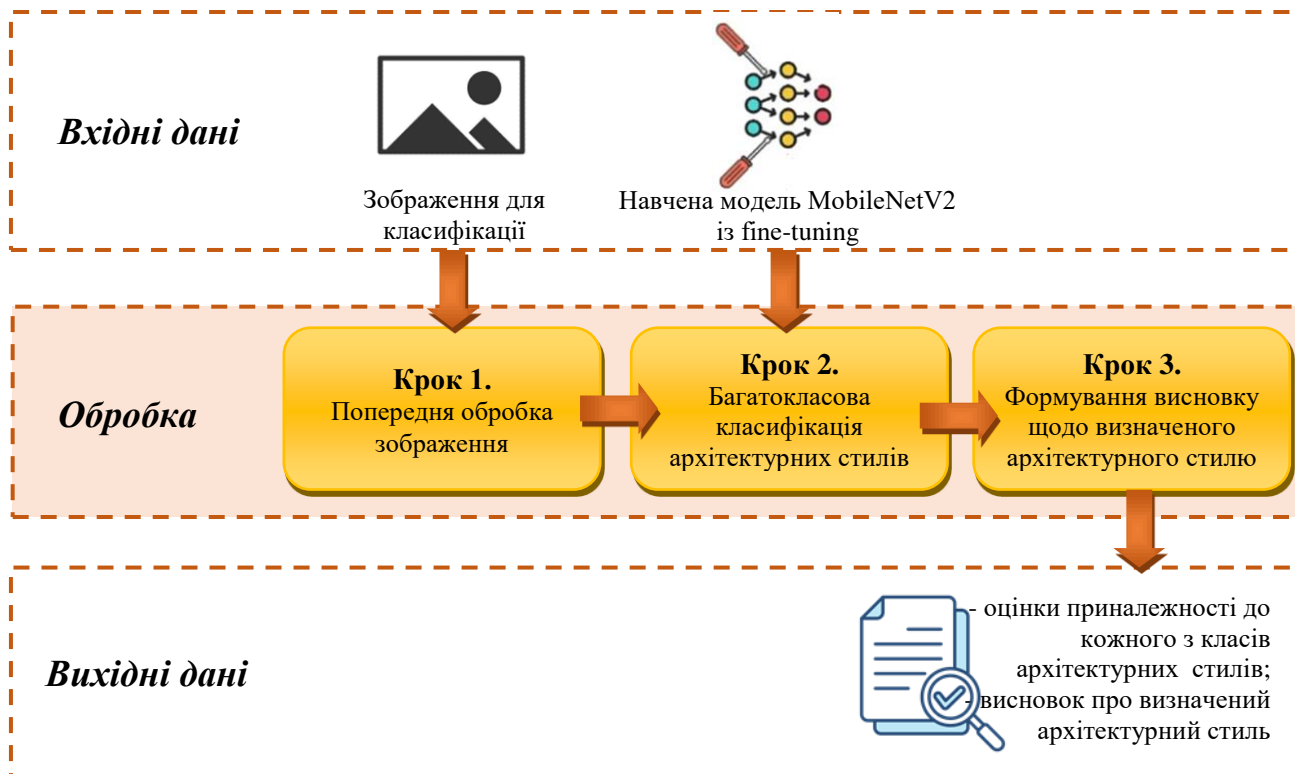
41. matplotlib.pyplot – Matplotlib 3.5.3 documentation. *Matplotlib – Visualization with Python*. URL: [https://matplotlib.org/3.5.3/api/\\_as\\_gen/matplotlib.pyplot.html](https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html) (date of access: 27.05.2025).

42. Module: `tf.keras.preprocessing` | TensorFlow v2.16.1. *TensorFlow*. URL: [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing) (date of access: 27.05.2025).
43. NumPy. *NumPy*. URL: <https://numpy.org/> (date of access: 27.05.2025).
44. Welcome to Flask – Flask Documentation (3.1.x). *Welcome to Flask – Flask Documentation (3.1.x)*. URL: <https://flask.palletsprojects.com/en/stable/> (date of access: 27.05.2025).
45. Understanding the Confusion Matrix in Machine Learning. Geeksforgeeks. URL: <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/> (date of access: 27.05.2025).
46. Accuracy vs. precision vs. recall in machine learning: what's the difference. *Evidentlyai*. URL: <https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall> (date of access: 27.05.2025).

# ДОДАТКИ

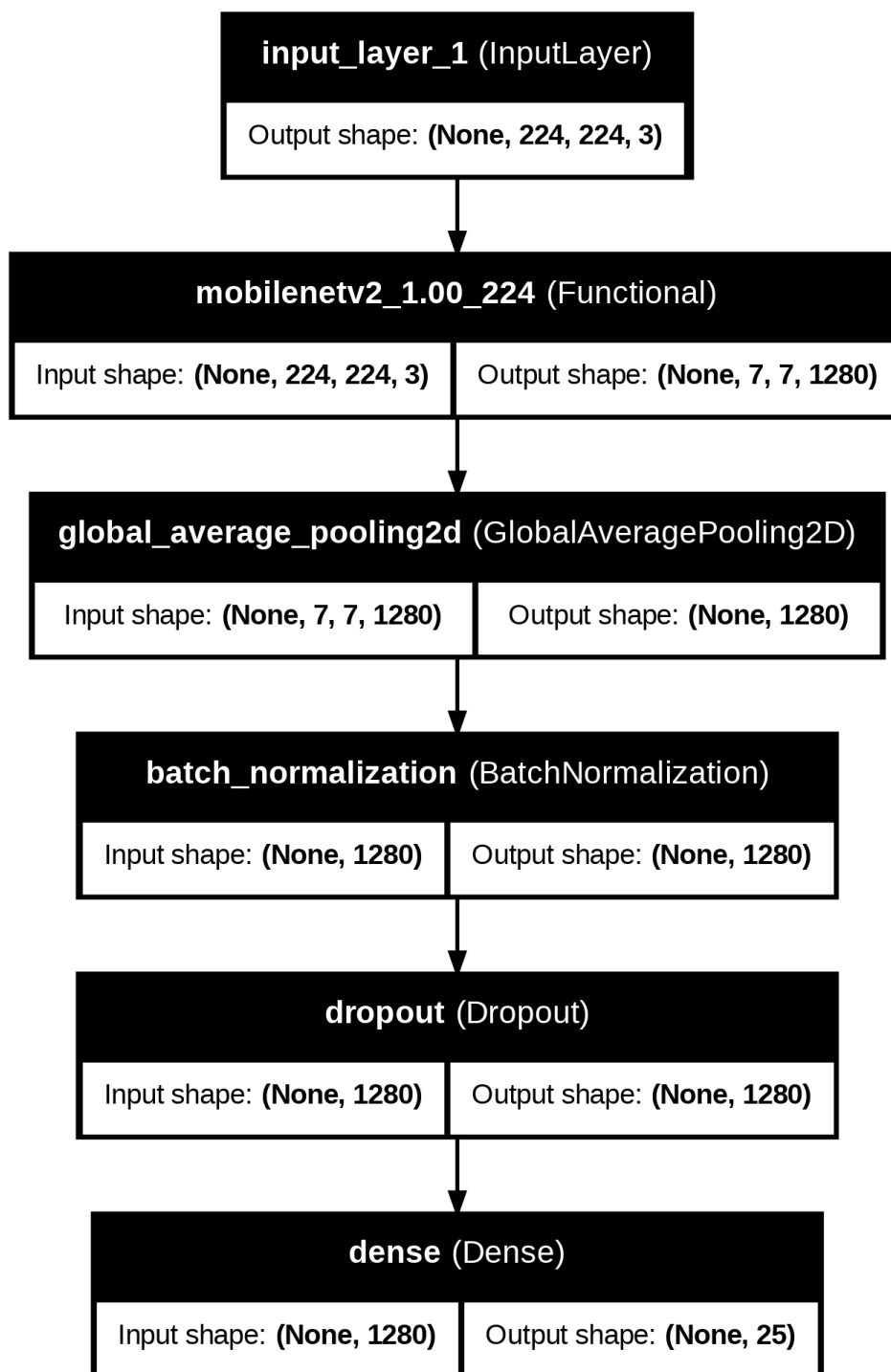
## Додаток А

## Схема методу визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж



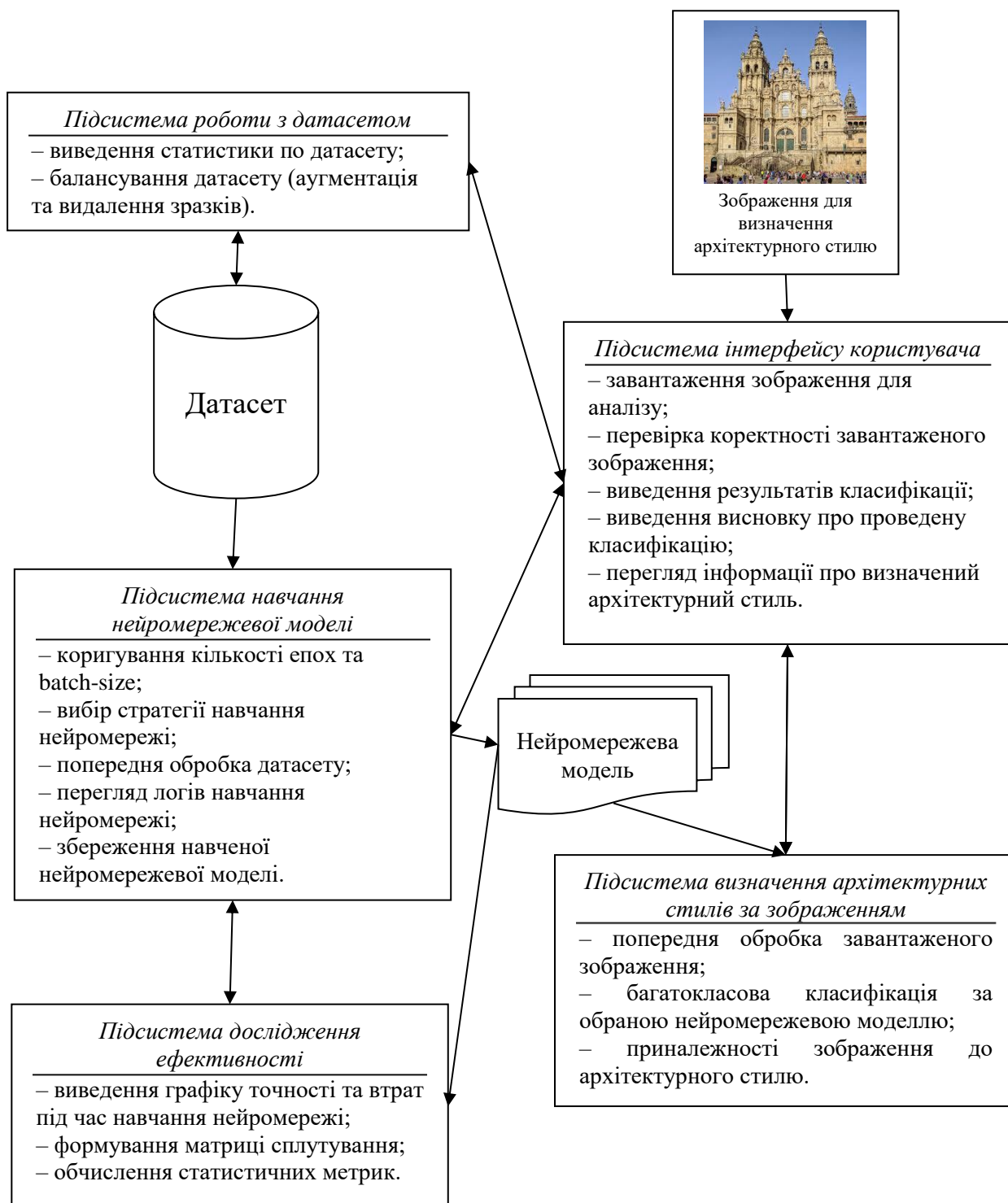
## Додаток Б

## Архітектура згорткової нейронної мережі MobileNetV2



## Додаток В

## Схема підсистем інтелектуальної системи автоматизованого визначення архітектурних стилів



## Додаток Г

## Діаграма класів системи автоматизованого визначення архітектурних стилів

arch_style_detection.modules.dataset_handler.DatasetHandler
dataset_dir : str upload_folder : str
allowed_file(filename) balance_dataset(target_count) get_dataset_stats() handle_dataset_upload(file) load_dataset(batch_size, validation_split)

arch_style_detection.modules.evaluator.Evaluator
history : NoneType
compute_metrics(model, generator) evaluate(model, generator) generate_confusion_matrix(model, generator) plot_training_history(history)

arch_style_detection.modules.model_trainer.ModelTrainer
dataset_path : str img_size : int logger : NoneType, RootLogger model : NoneType model_path : str
build_model(trainable_layers) preprocess_dataset() train(epochs, batch_size, trainable_layers) view_logs()

arch_style_detection.modules.predictor.Predictor
img_size : int model
classification_conclusion(prediction_result) get_style_info(predicted_class) load_image(image_path) predict_image(image_path) preprocess_image(img) validate_image(img)

## Додаток Д

### Програмні коди

Вихідний код розробленої інтелектуальної системи доступний у репозиторії GitHub: [https://github.com/mushtyn/architectural\\_styles](https://github.com/mushtyn/architectural_styles) (дата звернення: 10.06.2025).

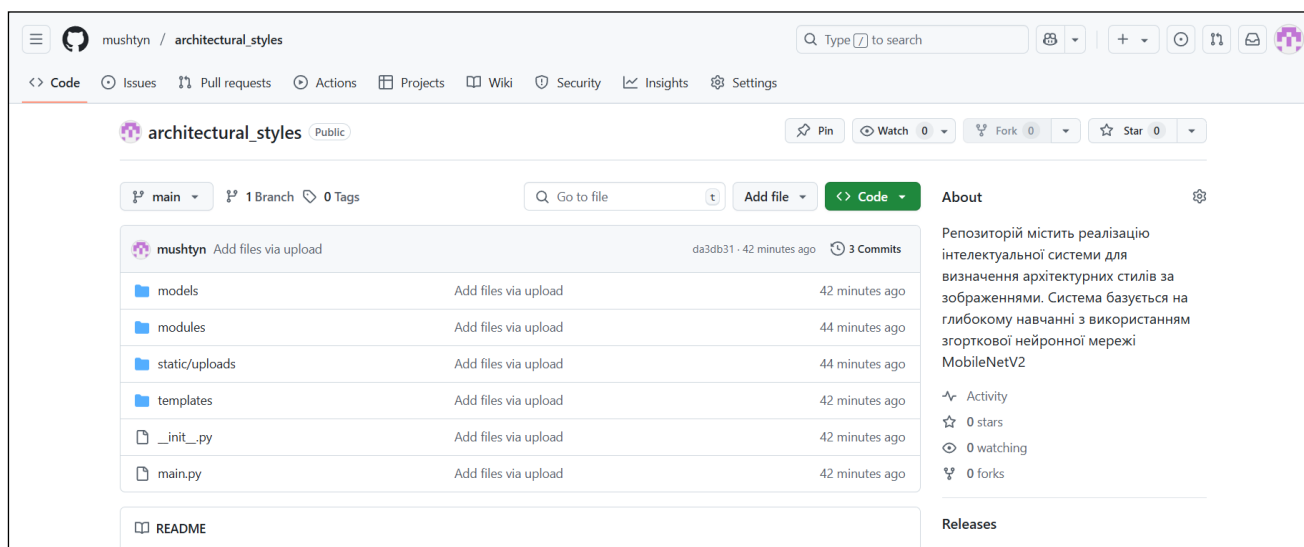


Рисунок Д.1 – Сторінка репозиторію architectural\_styles

Структура репозиторію architectural\_styles містить:

- models/ – збережені моделі для класифікації архітектурних стилів.
- modules/ – Python-модулі, описані у пункті 3.3.
- static/uploads/ – папка для завантажених користувачем зображень (веб-інтерфейс або API), які підлягають класифікації.
- templates/ – HTML-шаблони для веб-інтерфейсу фреймворку Flask.

## Додаток Е

### Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

# МЕТОД ВИЗНАЧЕННЯ АРХІТЕКТУРНИХ СТИЛІВ ЗА ЗОБРАЖЕННЯМ ЗАСОБАМИ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ



**Виконав:**  
*студент групи КНС-22-1*  
**Олександр МУШТИН**



**Керівник:**  
*асистент каф. КН*  
**Олена ТИЩЕНКО**

## Актуальність

У сучасну епоху стрімкого розвитку цифрових технологій автоматичне розпізнавання візуального контенту стає дедалі важливішим у різних галузях – від культурної спадщини до урбаністики та комерційного дизайну. Визначення архітектурних стилів за зображенням є складним завданням через велику різноманітність стилістичних рис, комбінацій елементів і вплив часу на зовнішній вигляд будівель. Традиційні підходи, що базуються на ручній класифікації або аналізі окремих ознак, виявляються недостатньо ефективними у випадках великого обсягу даних або складних змішаних стилів.

Глибокі нейронні мережі, зокрема згорткові нейронні мережі (CNN), довели свою високу ефективність у завданнях комп'ютерного зору, таких як класифікація об'єктів, розпізнавання образів та сегментація сцен. Їхня здатність автоматично виявляти приховані закономірності у великих масивах зображень робить їх ідеальним інструментом для розв'язання задачі визначення архітектурного стилю.

Таким чином, запропонований метод надає нові можливості для інтеграції сучасних технологій у сферу збереження архітектурної спадщини та розвитку смарт-міст.

## Мета і задачі роботи

**Об'єкт дослідження** – процес визначення архітектурних стилів за зображенням.

**Предмет дослідження** – методи та технології глибоких нейронних мереж для визначення архітектурних стилів за зображенням.

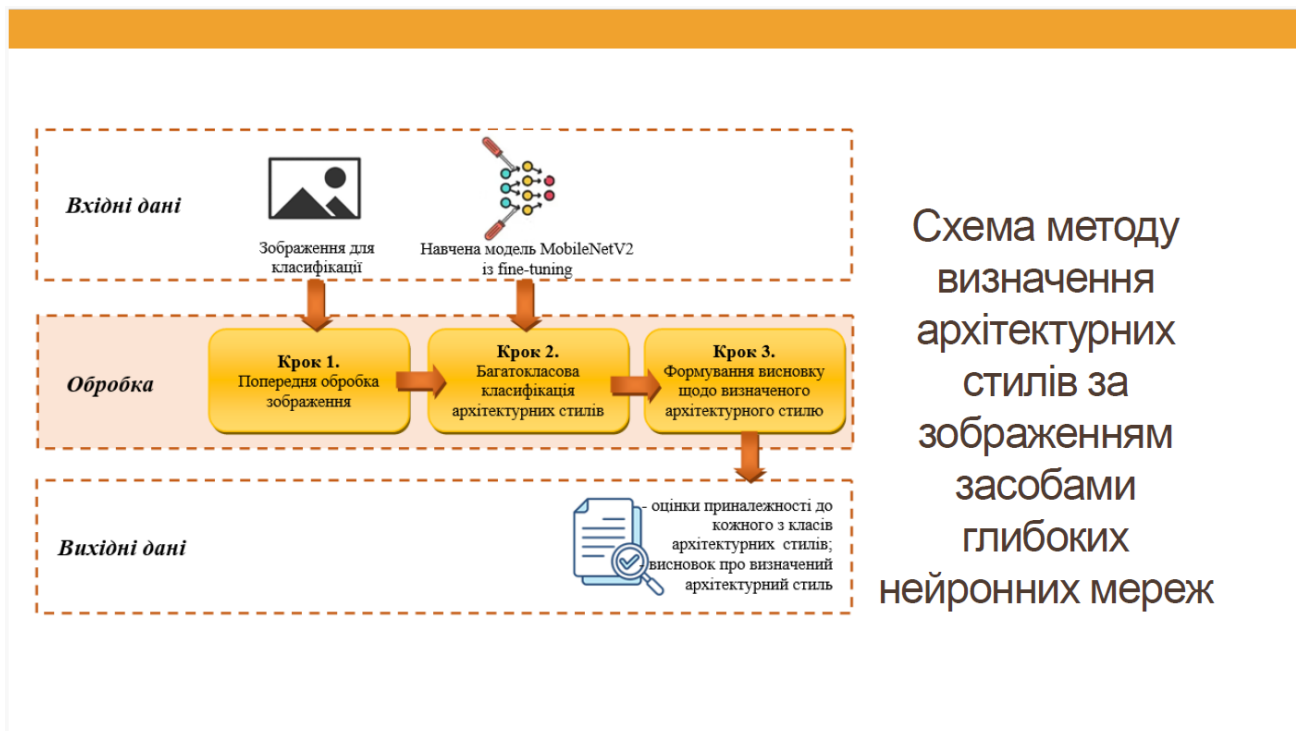
**Метою кваліфікаційної роботи бакалавра** підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

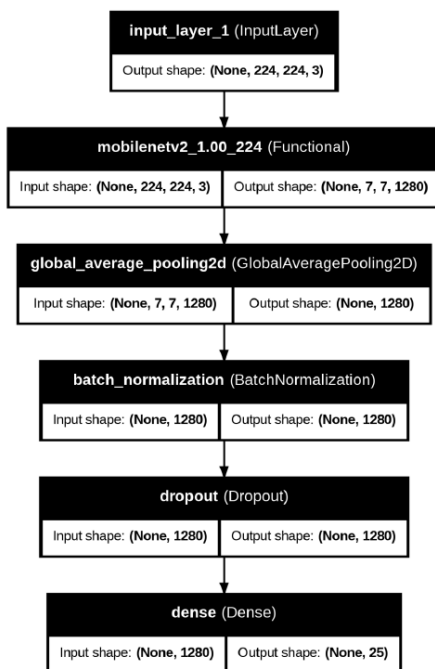
Для досягнення поставленої мети слід вирішити такі **завдання**:

- провести аналіз предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням;
- розробити метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж;
- здійснити проектування та програмну реалізацію спроектованої інтелектуальної системи на основі запропонованого методу;
- виконати дослідження запропонованого методу з використанням розробленої інтелектуальної системи визначення архітектурних стилів.

## Підхід до роботи

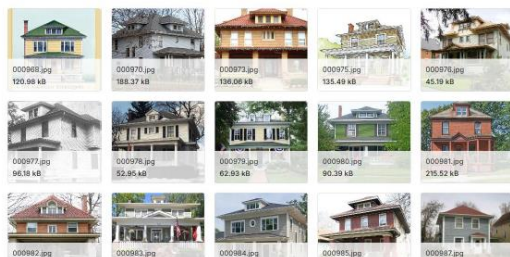




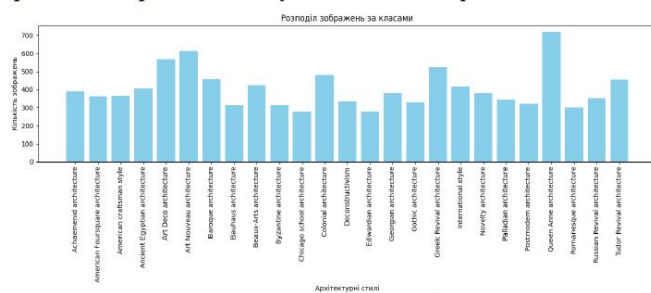


Архітектура  
неймережевої  
моделі для  
визначення  
архітектурних  
стилів за  
зображенням

### «Architectural Styles Dataset»



Приклади зображень з класу American Foursquare architecture датасету

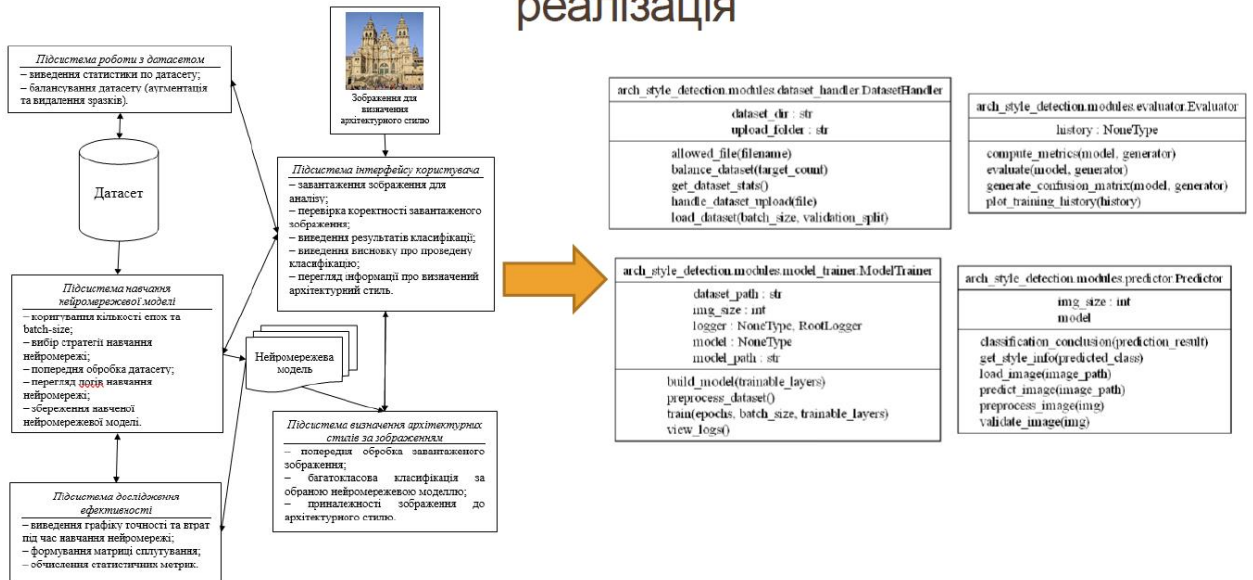


Класи датасету та кількість зображень у них

Формування  
датасету для  
навчання  
неймережевої  
моделі

Загальна кількість зразків у датасеті після балансування зразків у ньому становить 12500, з них для навчання використовуватиметься 10625, а для тестування 1875 зображень.

# Проектування інтелектуальної системи та програмна реалізація



# Інтелектуальна система визначення архітектурних стилів

## Інтелектуальна система визначення архітектурних стилів

1. Визначення архітектурних стилів за зображенням
2. Навчання нейронмережевої моделі
3. Робота з датасетом
4. Дослідження ефективності

### 1. Розпізнавання архітектурного стилю

Завантажити фотографію

0a26e998-71b-11ef-be2f-eeffbe1deba3.1220x600.jpeg

Визначити стиль

#### Результати класифікації

найбільш ймовірний стиль: Baroque architecture  
 Ймовірність: 87.37%  
 Зображення класифіковано як стиль 'baroque architecture' з впевненістю 0.8737.

Опис стилю: Бароковий стиль 17–18 століть із драматичними, складними дизайнами, витупими формами та величчю, наприклад, у Базиліці Святого Петра.



## Інтелектуальна система визначення архітектурних стилів

1. Визначення архітектурних стилів за зображенням
2. Навчання нейронмережевої моделі
3. Робота з датасетом
4. Дослідження ефективності

### 2. Навчання моделі

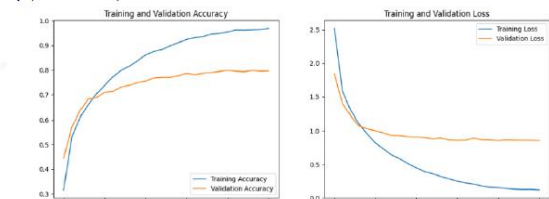
Кількість епох

Розмір batch

Кількість шарів для fine-tuning (0 для feature extraction)

Почати навчання

### Графік точності та втрат



# Інтелектуальна система визначення архітектурних стилів

## Інтелектуальна система визначення архітектурних стилів

1. Визначення архітектурних стилів за зображенням
2. Навчання нейромережевої моделі
3. Робота з датасетом
4. Дослідження ефективності

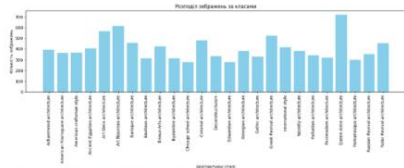
### 3. Робота з датасетом

Завантажити датасет (ZIP)

Вибрати файл arch\_dataset.zip

Завантажити

Отримати статистику



Цільова кількість зображень на клас (залиште порожнім для середнього)

400

Балансувати

## Інтелектуальна система визначення архітектурних стилів

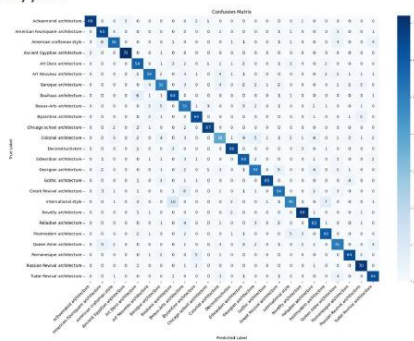
1. Визначення архітектурних стилів за зображенням
2. Навчання нейромережевої моделі
3. Робота з датасетом
4. Дослідження ефективності

### 4. Оцінка ефективності

Обчислення точності, precision, recall, F1 та побудова матриці сплутування.

Оцінити модель

#### Матриця сплутування

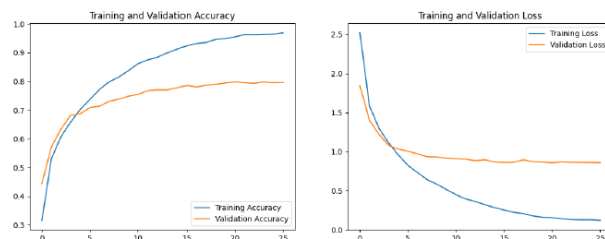


## Експериментальне дослідження

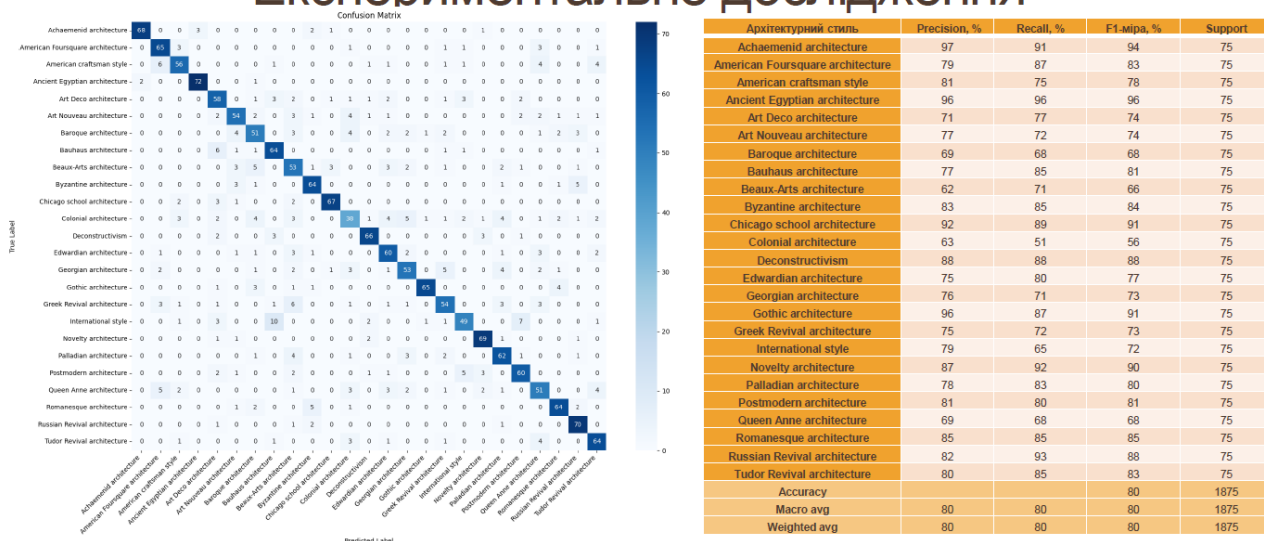
Залежність точності класифікації від кількості епох та batch-size та стратегії навчання

Епохи	batch-size	Accuracy, %	Стратегія навчання
30	32	72	fixed number of epochs
50	16	65	fixed number of epochs
20	32	78	early stopping
26	16	80	early stopping

Графіки точності та втрат під час навчання нейромережевої моделі MobileNetV2



## Експериментальне дослідження



Матриця сплутувань навченої нейронмережової моделі

Показники Precision, Recall, F1-міра для багатокласової класифікації

## Висновки

Було досягнуто мету кваліфікаційної роботи бакалавра підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж.

Для досягнення поставленої мети було поставлено та вирішено такі завдання:

- проведено аналіз предметної області, існуючих методів та публікацій, присвячених визначенню архітектурних стилів за зображенням;
- розроблено метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж;
- здійснено проєктування та програмну реалізацію спроектованої інтелектуальної системи на основі запропонованого методу;
- виконано дослідження запропонованого методу з використанням розробленої інтелектуальної системи визначення архітектурних стилів.

# Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 2.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. **Errors in the documents: 12%**

ID: 246617 Title: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж Added in a DB: 2025-06-17 Authors: Олександр МУШТИН Heads: Олена ТИЩЕНКО Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	81285	1186	3144 (4%)	48 (4%)

## Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Олександр МУШТИН

**Співавтор:**

**Назва:** КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

**Науковий керівник:** Олена ТИЩЕНКО, асистент. каф. КН

**Підрозділ:** Кафедра комп'ютерних наук

**Коефіцієнт подібності 1:** 4.3%

**Коефіцієнт подібності 2:** 2%

**Мікропробіли:** 0

**Заміна букв:** 3

**Інтервали:** 0

**Білі знаки:** 191

**Дата створення звіту:** 2025-06-17 20:26:39.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-17

Дата

експерт

*Л.В. Мигурський Р.Р.*

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК**

**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

Автор студент групи КНс-22-1 Олександр Муштин

Освітня програма Комп'ютерні науки

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 – Комп'ютерні науки

Науковий керівник: асистент каф. комп'ютерних наук Олена Тищенко

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідає</i>
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	<i>відсутні</i>

**Підтвердження:**

*Запозичення, виявлені в роботі Олександра Муштина, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.*

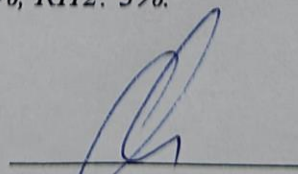
*Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:*

*- за системою Anti-Plagiarism: 2%;*

*- за системою StrikePlagiarism КП1: 4,3%, КП2: 3%.*

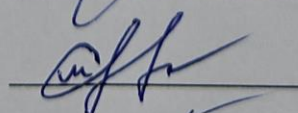
18.06.2025

Завідувач кафедри



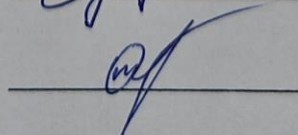
Олександр БАРМАК

Гарант освітньої програми



Олександр МАЗУРЕЦЬ

Керівник кваліфікаційної роботи



Олена ТИЩЕНКО



**ВІДГУК НАУКОВОГО КЕРІВНИКА**  
**на кваліфікаційну роботу бакалавра**

студента гр. КНс-22-1 Муштина Олександра Романовича

за темою Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

**1. Актуальність теми**

Автоматичне розпізнавання архітектурних стилів є актуальним завданням комп'ютерного зору, особливо в умовах зростання обсягів візуальних даних. Через складність стилістичних ознак і поєднання елементів традиційні методи аналізу виявляються малоефективними. Використання згорткових нейронних мереж відкриває можливості для точного й масштабованого визначення стилів, що важливо для збереження культурної спадщини та впровадження інтелектуальних технологій у міське середовище.

**2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки**

Керуючись стандартом, а саме описом предметної області, об'єктом роботи є процес визначення архітектурних стилів за зображенням. Метою роботи є підвищення точності автоматизованого визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж. При вирішенні поставленої задачі використано методи та технології глибоких нейронних мереж для визначення архітектурних стилів за зображенням. Тому результати виконання кваліфікаційної роботи бакалавра відповідають стандарту бакалавра спеціальності 122 – Комп'ютерні науки.

**3. Професійні та особистісні якості бакалавра**

У ході виконання кваліфікаційної роботи студент продемонстрував високий рівень академічної підготовки, поєднуючи теоретичні знання з практичними навичками. Особливо слід відзначити його здатність критично оцінювати джерела, структурувати матеріал дослідження та обґрунтовано підходити до прийняття технічних рішень. Практична частина роботи – створення інтелектуальної системи – виконана на високому рівні з урахуванням сучасних вимог до якості розробки, що підтверджує його компетентність як майбутнього фахівця у сфері інформаційних технологій.

#### **4. Ступінь самостійності під час виконання кваліфікаційної роботи**

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

#### **5. Ступінь оволодіння методами дослідження**

Виконана робота демонструє орієнтацію студента у спеціалізованих інструментах і програмних засобах, необхідних для реалізації обчислювальних і дослідницьких завдань, а також володіння сучасними методами збору, обробки та аналізу даних, методах та засобах штучного інтелекту.

#### **6. Повнота та якість розкриття теми роботи**

Робота характеризується науковою завершеністю: тема розкрита комплексно, проведено аналіз літератури за напрямом дослідження, обґрунтовано методологію, виконано всі поставлені завдання та створено інтелектуальну систему для експериментального дослідження запропонованого методу.

#### **7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу**

Робота має чітку структурну організацію, що сприяє розумінню логіки дослідження. Викладення думок є послідовним і аргументованим, термінологічно точним та відповідає вимогам літературної грамотності наукових текстів.

#### **8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин**

Розроблений у роботі метод та його програмна реалізація може бути використана архітекторами, істориками та туристами для надання висновків щодо визначеного архітектурного стилю за зображенням.

#### **9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота**

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник \_\_\_\_\_



асистент каф. КН Олена ТИЩЕНКО



## РЕЦЕНЗІЯ

### на кваліфікаційну роботу бакалавра

студента гр. КНс-22-1 Муштина Олександра Романовича

за темою: Метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж

#### 1. Актуальність обраної теми

Ідентифікація архітектурних стилів на основі зображень набуває особливої ваги у зв'язку з потребами цифровізації міст та охорони спадщини. Складність класифікації зумовлена стилістичною різноманітністю та змінами зовнішнього вигляду будівель з часом. Глибокі нейронні мережі, здатні виявляти візуальні закономірності, забезпечують ефективне рішення для автоматизації цього процесу, що робить їх корисними як у наукових дослідженнях, так і в прикладних системах міського управління.

#### 2. Повнота розкриття мети та завдань роботи

Робота відзначається цілісністю структури: автор послідовно реалізував визначену мету через виконання відповідних завдань, що засвідчує високий рівень розуміння предметної галузі та здатність до самостійного наукового пошуку.

#### 3. Зміст кожного розділу роботи

Пояснювальна записка побудована з дотриманням принципу методичної відповідності: кожен розділ деталізує окремий аспект дослідницьких завдань. Перший розділ містить характеристику предметної області визначення архітектурних стилів за зображенням. У другому розділі детально описано метод визначення архітектурних стилів за зображенням засобами глибоких нейронних мереж. А у третьому описано програмну реалізацію інтелектуальної системи та експериментальне дослідження методу.

#### 4. Оцінка розробленої інформаційної системи, її практична цінність

Розроблена система автоматично класифікує архітектурні об'єкти за стилістичними ознаками, що має високу практичну цінність у сфері цифрової спадщини та віртуального туризму. Практична цінність системи полягає в тому, що вона дозволяє користувачам швидко отримати інформацію про архітектурний контекст об'єктів, що сприяє збереженню культурної ідентичності.

5. Якість оформлення кваліфікаційної роботи бакалавра

Оформлення дослідження здійснено з дотриманням вимог щодо структури, мови та оформлення ілюстративного матеріалу, що свідчить про сформовані навички роботи над науковими дослідженнями та уважність до деталей.

6. Недоліки кваліфікаційної роботи бакалавра

Опис програмної реалізації можна доповнити додатковими діаграмами, що краще би пояснювали процес перетворення вхідних даних на вихідні. Є необхідність більш системного порівняння існуючих підходів з авторським. У вступі та висновках окремі формулювання можуть бути деталізовані для забезпечення кращої логічної завершеності. Недоліки мають переважно формальний характер і не порушують цілісності та важливості отриманих висновків.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент



проф. каф ЖІТС Савенко О.С.