

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

*Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління*

"На правах рукопису"

До захисту допущено

УДК 004.93

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

“ ” 20 21 р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

на здобуття ступеня магістра

за освітньо-науковою програмою

«Інформаційні управляючі системи та технології» зі

спеціальності 126 «Інформаційні системи та технології»

на тему:

«Система розпізнавання архітектурних стилів будівель за зображеннями»

Виконала:

студентка VI курсу, групи ІС-91му

Новіченко Неля Валеріївна

Керівник: доктор фізико-математичних наук,
професор

Гупал Анатолій Михайлович Консультант:

доцент, к.т.н., доцент,

Жданова Олена Григорівна

Рецензент:

професор, д.т.н., професор,

Стенін Олександр Африканович

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань. Студентка Новіченко
Неля Валеріївна

Київ – 2021 року

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

*Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління*

Рівень вищої освіти – *другий (магістерський)*

Спеціальність – *126 «Інформаційні системи та технології»*

Освітньо-наукова програма *«Інформаційні управляючі системи та технології»*

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ

«__» _____ 2021 р.

ЗАВДАННЯ на магістерську дисертацію студенту

Новіченко Нелі Валеріївни

1. Тема дисертації «Система розпізнавання архітектурних стилів будівель за зображеннями», науковий керівник дисертації Гупал Анатолій Михайлович, доктор фізико-математичних наук, професор, затверджені наказом по університету від « 12 » березня 2021 р. №809-с.

2. Строк подання студентом дисертації “ 10 ” травня 2021 р.

3. Об’єкт дослідження: класифікація зображень будівель за архітектурним стилем.

4. Перелік завдань, які потрібно розробити:

- 1) провести аналіз існуючих методів класифікації зображень;
- 2) модифікація існуючих методів для покращення точності та зменшення часу навчання;
- 3) створення вибірки зображень будівель;
- 4) програмна реалізація алгоритму та розробка програмного забезпечення;

- 5) порівняти ефективність модифікованого алгоритму з існуючими для розв'язання задачі визначення архітектурного стилю будівлі.

5. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- 1) схема структурна діаграми класів
- 2) схема структурна послідовності аутентифікації користувача;
- 3) схема структурна послідовності реєстрації користувача;
- 4) схема структурна послідовності аналізу зображення;
- 5) схема структурна розгортання;
- 6) схема структурна компонентів.

6. Орієнтовний перелік публікацій: чотири публікації – стаття у фаховому журналі та три тези доповідей на наукових конференціях.

7. Консультанти розділів дисертації

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

8. Дата видачі завдання “ 1 ” лютого 20 21 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	<i>Систематизація результатів огляду літератури</i>	01.02.2021	
2	<i>Порівняльний аналіз існуючих методів розв'язання задачі</i>	14.02.2021	
3	<i>Постановка та формалізація математичної моделі задачі</i>	27.02.2021	
4	<i>Модифікація існуючих методів розв'язання задачі</i>	04.03.2021	
5	<i>Розробка інформаційного та програмного забезпечення</i>	26.03.2021	
7	<i>Проведення експериментальних досліджень розроблених алгоритмів</i>	03.04.2021	
8	<i>Оформлення документації</i>	10.04.2021	
9	<i>Подання роботи на попередній захист</i>	22.04.2021	
10	<i>Подання роботи на основний захист</i>	20.05.2021	

Студент

Неля НОВІЧЕНКО

Науковий керівник

Анатолій ГУПАЛ

РЕФЕРАТ

Магістерська дисертація: с. 90, рис. 14, табл. 21, 30 джерел, 1 додаток.

Актуальність. Задача класифікації зображення будівель за архітектурним стилем відрізняється від інших задач класифікації наявністю міжкласових взаємозв'язків між різними стилями, тобто велика кількість рис та ознак є спільною для декількох архітектурних стилів. Іншим фактором, що робить задачу більш складною, є існування певних відмінностей у кожному стилі в залежності від території та часу, коли будівля була збудована. Необхідність мати можливість визначати стиль для зображень з різних ракурсів та різної якості накладає додаткові вимоги на алгоритми класифікації зображень. Визначення архітектурних стилів будівель – це поширена задача в документуванні архітектурної спадщини та моделюванні, проектуванні нових будівель у містах, тому що існує необхідно зберегти єдиний архітектурний стиль.

Розробка алгоритму, що дозволяє виконувати класифікацію зображень з тісними міжкласовими зв'язками, є актуальною у наш час не тільки для класифікації архітектурного стилю будівель, а також і в інших галузях.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалась на кафедрі автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Ефективні методи розв'язання задач теорії розкладів» (№ ДР 0117U000919).

Мета дослідження – удосконалення алгоритмів класифікації зображень з тісними міжкласовими взаємозв'язками, зменшення часу навчання нейронної мережі при збереженні точності навчання.

Для досягнення мети необхідно виконати наступні **завдання**:

— провести аналіз існуючих методів класифікації зображень;

- модифікація існуючих методів для покращення точності та зменшення часу навчання;
- створення вибірки зображень будівель;
- програмна реалізація алгоритму та розробка програмного забезпечення;
- порівняти ефективність модифікованого алгоритму з існуючими для розв’язання задачі визначення архітектурного стилю будівлі.

Об’єкт дослідження – процес навчання нейронної мережі для класифікації зображень з тісними міжкласовими взаємозв’язками та класифікація зображень будівель за архітектурним стилем.

Предмет дослідження – алгоритми-оптимізатори та методи, що використовуються під час навчання нейронної мережі для зміни її атрибутів, таких як ваги та швидкість навчання.

Наукова новизна отриманих результатів полягає в удосконаленні алгоритму коригування параметрів нейронної мережі під час її навчання для розв’язання задачі класифікації зображень з тісними міжкласовими взаємозв’язками.

Практичне значення одержаних результатів полягає у створенні системи розпізнавання архітектурних стилів будівель та методу навчання нейронної мережі, що за меншу кількість часу навчання досягає більшої точності.

Публікації. Матеріали роботи опубліковані в тезах науково-практичної конференції молодих вчених та студентів «Інформатика та обчислювальна техніка-ІОТ-2020», «Інформатика та обчислювальна техніка-ІОТ-2021» та прийнята до публікації стаття у науковому фаховому виданні «Системні технології».

МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, ОПТИМІЗАТОРИ,
КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, АРХІТЕКТУРНІ СТИЛІ

ABSTRACT

Master's Thesis: pp. 90, fig. 14, tab. 21, app. 30, sources 1.

Topicality. The problem of classifying the image of buildings by architectural style differs from other problems of classification by the presence of interclass relationships between different styles, ie a large number of features and characteristics are common to several architectural styles. Another factor that makes the task more difficult is the existence of certain differences in each style depending on the area and time when the building was built. The need to be able to determine the style for images from different angles and different quality imposes additional requirements on image classification algorithms. Defining the architectural styles of buildings is a common task in documenting the architectural heritage and modeling, designing new buildings in cities, because there is a need to maintain a single architectural style.

The development of an algorithm that allows the classification of images with close interclass relationships is relevant today not only for the classification of architectural style of buildings, but also in other areas.

Relationship of work with scientific programs, plans, themes. The work was performed at the Department of Automated Information Processing and Control Systems of the National Technical University of Ukraine «Kyiv Polytechnic Institute. Igor Sikorsky» in the framework of the «Effective methods for solving problems of schedule theory» (state registration number 0117U000919).

The aim of the research is to improve the algorithms for classifying images with close interclass relationships, reducing the learning time of the neural network while maintaining the accuracy of learning.

To achieve this goal the following tasks have to be completed:

- to analyze the existing methods of image classification;
- modification of existing methods to improve accuracy and reduce training time;
- creating a sample of images of buildings;
- software implementation of the algorithm and software development;

- compare the efficiency of the modified algorithm with the existing ones to solve the problem of determining the architectural style of the building.

The object of study is the process of learning a neural network for the classification of images with close interclass relationships and the classification of images of buildings by architectural style.

The subject of study is optimization algorithms and methods used during neural network training to change its attributes, such as weight and learning speed.

Scientific novelty of the obtained results is to obtain an algorithm-optimizer to reduce losses in the training of the neural network to solve the problem of classifying images with close interclass relationships.

The practical significance of the obtained results is to create a system for recognizing architectural styles of buildings and a method of learning the neural network, which in less time achieves greater accuracy.

Publications. Materials of the work were published in the abstracts of the scientific-practical conference of young scientists and students "Informatics and Computer Engineering-IOT-2020", "Informatics and Computer Engineering-IOT-2021" and accepted for publication in the scientific professional publication "System Technologies".

MACHINE LEARNING, NEURAL NETWORKS, OPTIMIZERS, IMAGE CLASSIFICATION, ARCHITECTURAL STYLES

ЗМІСТ

ВСТУП	10
1 ОГЛЯД МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ.....	12
1.1 Огляд існуючих рішень класифікації зображень будівель	12
1.2 Дослідження класифікації за допомогою глибинного навчання.....	17
1.3 Дослідження згорткових нейронних мереж (CNN).....	18
1.4 Дослідження архітектур згорткових нейронних мереж	22
1.5 Алгоритми зміни атрибутів нейронної мережі	26
1.6 Постановка задачі.....	28
Висновки до розділу	29
2 РОЗРОБКА АЛГОРИТМУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ	31
2.1 Опис загальних понять для розв'язання задачі	31
2.2 Опис процесу виділення кордонів та меж	33
2.2.1 Згортка зображення.....	33
2.2.2 Шари згортки у нейронній мережі	34
2.2.3 Зворотнє поширення у шарах згортки	34
2.3 Процес зменшення зображення	34
2.3.1 Шари об'єднання.....	35
2.3.2 Зворотнє поширення у шарах об'єднання.....	35
2.4 Опис процесу навчання нейронної мережі	35
2.5 Математична постановка задачі	38
2.6 Схема алгоритму розв'язання задачі.....	38
2.6.1 Алгоритм навчання	38
2.6.2 Алгоритм класифікації	39
2.7 Обґрунтування правильності роботи алгоритму	40

Висновки до розділу	41
3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	42
3.1 Вимоги до програмного продукту	42
3.1.1 Вимоги до функціональних характеристик.....	42
3.1.2 Вимоги до надійності.....	42
3.1.3 Вимоги до складу і параметрів технічних засобів.....	42
3.2 Засоби розробки.....	42
3.3 Архітектура програмного забезпечення	44
3.3.1 Діаграма класів	44
3.3.2 Діаграма послідовностей	44
3.3.3 Діаграма компонентів	45
3.4 Інструкція користувача	46
3.5 Опис технічного забезпечення.....	50
Висновки по розділу	51
4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ	53
4.1 Бенчмарк.....	53
4.2 Порівняння результатів аналізу зображень	54
Висновки по розділу	56
5 РОЗРОБКА СТАРТАП-ПРОЕКТУ	57
5.1 Опис ідеї проекту	57
5.1.1 Зміст ідеї.....	57
5.1.2 Аналіз переваг ідеї	58
5.2 Технологічний аудит ідеї проекту	60
5.3 Аналіз ринкових можливостей запуску стартап-проекту	61
5.4 Розроблення ринкової стратегії проекту.....	69

5.5 Розроблення маркетингової програми стартап-проекту	72
Висновки по розділу	76
ЗАГАЛЬНІ ВИСНОВКИ	78
ПЕРЕЛІК ПОСИЛАНЬ	80
ДОДАТОК А Графічний матеріал.....	84
Схема структурна діаграми класів	85
Схема структурна послідовності аутентифікації користувача.....	86
Схема структурна послідовності реєстрації користувача.....	87
Схема структурна послідовності аналізу зображення	88
Схема структурна розгортання	89
Схема структурна компонентів	90

ВСТУП

В епоху цифрових технологій, майже кожна людина має вільний доступ до пристроїв, що створюють цифрову інформацію – смартфони, фото- та відеокамери. Зі зростанням її кількості виникає потреба у її обробці та класифікації, а саме в автоматизації цих процесів. Класифікація інформації широко використовується у галузях медицини, безпеки, освіти та має широке застосування в екологічних та соціально-економічних галузях. Вдосконалення методів та алгоритмів обробки інформації принесуть велику користь і точність цих методів має велике значення якщо використовується, наприклад, в медицині.

Класифікація зображень, зроблених під час вимірювання архітектурного об'єкта, є важливим завданням у цифровій документації культурної та архітектурної спадщини. Для документації має бути оброблено велика кількість зображень, тому їх класифікація є нудним завданням (а це означає також схильним до помилок) і забирає багато часу. Наявність автоматичних методів для полегшення цих завдань сортування покращило б важливу частину процесу цифрової документації. Крім того, правильна класифікація наявних зображень дозволяє покращити управління та більш ефективний пошук за певними термінами, допомагаючи тим самим у вивченні та інтерпретації відповідного об'єкта спадщини.

Визначення архітектурного стилю також сприяє отриманню нових знань для туристів завдяки можливості використати систему і отримати опис архітектурного стилю з інформацією про його територіальні особливості завантаживши лише фотографію будівлі.

З іншого боку дослідження задачі визначення архітектурного стилю будівлі дозволяє знайти методи та розробити алгоритми, що будуть ефективними для задач класифікації з наявністю тісних міжкласових взаємозв'язків, тобто коли велика кількість основних рис та ознак є спільною для декількох стилів. На сьогоднішній день існує велика кількість алгоритмів для розв'язання задач класифікації за

класами зі значними відмінностями, проте з тісними міжкласовими взаємозв'язками – навпаки.

Розробка системи класифікації архітектурних стилів будівель за зображеннями є актуальною, тому що аналогів на ринку не було знайдено. Аналіз ринку показав, що продукт має велику кількість потенційних клієнтів.

У даній роботі розглянуто використання нейронних мереж для розв'язання задачі класифікації зображень та розглянуто способи підвищення точності результатів та зменшення часу, необхідного для навчання алгоритму. Методи можуть бути використані для розв'язання задачі класифікації будь-яких зображень.

1 ОГЛЯД МЕТОДІВ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

Задача класифікації зображення полягає у присвоєнні об'єкту (ситуації) деяким чином певний клас з існуючого переліку. У математичній статистиці завдання класифікації називаються також завданнями дискримінантного аналізу. Існує велика кількість методів розв'язання поставленої задачі, кожен з яких має свої особливості та має різну ефективність залежно від особливостей задачі та вихідних даних.

1.1 Огляд існуючих рішень класифікації зображень будівель

Провівши аналіз існуючих робіт присвячених визначенню архітектурного стилю будівель можна виділити наступні найбільш поширені підходи до розробки алгоритмів класифікації, що показали гарні результати:

- Discriminative local-based patches or patterns;
- Multinomial Latent Logistic Regression та SVM;
- Naive-Bayes Nearest-Neighbor (NBNN);
- Bag of words;
- Deep Learning Techniques.

Discriminative local-based patches or patterns. Велика кількість існуючих алгоритмів класифікації за архітектурним стилем зосереджені на вилученні точково шматків зображення – шаблонів – a discriminative local pattern descriptor for image classification [1][2][3][4][5]. Наприклад, у роботі «Visual pattern discovery for architecture image classification and product image search» [2] реалізовано алгоритм класифікації за чотирма архітектурними стилями, що за допомогою виділення візуальних шаблонів у просторі дозволив отримати точні результати при масштабуванні об'єкта, обертанні та деформації. Автори «Are buildings only instances? Exploration in architectural style categories» [4] довели ефективність методу пари слів та семантичних шаблонів, досягнувши майже ідеальних результатів для опублікованих наборів даних – зображень будівель з 5 архітектурних стилів.

Переваги: алгоритм є ефективним для класифікації зображень при малій кількості архітектурних стилів.

Недоліки: абсолютно різні ділянки зображення можуть мати близьке значення в просторі функцій, що погіршує ефективність локальних виправлень; такі алгоритми не можуть дати гарні результати для зображень архітектури, багатих на деталі.

MLLR, Latent SVM. «Multinomial Latent Logistic Regression» - алгоритм, що базується на додаванні прихованих змінних в метод логістичної регресії. Завдяки цьому «менш значні» ознаки, що оточують «більш значні» ознаки, що можуть мати приховане значення для об'єкта, що класифікують, будуть враховані.

Алгоритм, розроблений Z. Xu та представлений у роботі «Architectural Style Classification using Multinomial Latent Logistic Regression» [6] розроблявся з використанням вибірки зображень будівель розподілених за 25 архітектурними стилями. Вводячи термін «ймовірність» в аналіз архітектурного стилю, автори аналізують зв'язки архітектурних стилів з використанням ймовірностей. Крім точності класифікації, результати алгоритму показують відношення до усіх стилів та забезпечують аналіз архітектурного стилю для окремих будівель. Алгоритм працює з деформованою моделлю на основі частин, що описують зображення у багатомасштабному режимі піраміди функцій HOG. Модель складається з трьох частин: кореневого фільтра, який показує загальні обриси об'єкта; набору фільтрів деталей для зображення з подвоєною роздільною здатністю кореневого фільтра, що передають детальну інформацію об'єкта; та витрати на деформацію, що показують відхилення деталей від їх розташування за замовчуванням щодо кореневого фільтра.

MLLR та LSVM мають багато спільних характеристик – форма прихованих змінних та властивість напіввипуклості. Ці подібності дозволяють MLLR використовувати існуючі моделі прихованих змінних, підготовлені LSVM, наприклад, моделі на основі деформації. Однак між MLLR та LSVM є деякі значні відмінності, що робить MLLR більш ефективним для завдання класифікації

архітектурного стилю. По-перше, SVM має тенденцію до неефективності при використанні незбалансованих даних навчання, де негативних прикладів набагато більше, ніж позитивних, чого не уникнути для вибірки зображень розподілених за архітектурними стилями. Величезний клас «фон зображення» у структурі виявлення об'єктів вводить знаний дисбаланс між позитивними та негативними прикладами. Щобільше, в LSVM навчальний процес має виправляти приховане значення для позитивних прикладів, в той час, коли всі можливі приховані значення для негативних прикладів мають бути збережені. По-друге, домінуючим методом розв'язання багатокласових задач класифікації за допомогою SVM була заснована на зведенні однієї багатокласової задачі до множинних двійкових задач. Оскільки кожна двійкова проблема тренується незалежно, прийняття цієї стратегії є проблематичним, оскільки вона не може виявити кореляцію між усіма класами. Як результат, вихідні значення розв'язання кожної окремої задачі не можуть бути порівнянні між собою, і це є відома проблема калібрування. MLLR тренує всі класи одночасно і вводить єдину цільову функцію і таким чином не страждає від можливості отримання різних упереджень, що виникають із багатокласовою проблемою та незбалансованими даними навчання. Нарешті, SVM не забезпечує гарного імовірнісного аналізу з «м'якими» кордонами.

Переваги: не значні деталі на зображенні будуть враховані у результат класифікації; алгоритм ефективний для великої кількості класів.

Недоліки: не ефективний при роботі з будівлями незвичайної форми – архітектурних стилів contemporary та novelty або ж з зображеннями з не стандартних ракурсів.

Naive-Bayes Nearest-Neighbor (NBNN). Реалізація алгоритму NBNN для класифікації архітектурних стилів будівель розглянута у роботі «AUTOMATIC ARCHITECTURAL STYLE RECOGNITION» [7]. Ідея алгоритму: розрахувати усі дескриптори для навчальних зображень і відсортувати за різними класами. Після цього для кожного дескриптора вхідного зображення, розраховується найближчий сусід до кожного класу використовуючи апроксимацію. Сума евклідових відстаней

кожного дескриптора позначає відстань від зображення до класу. Клас з найменшою відстанню обирається як клас переможець. [7]

Переваги: алгоритм простий у реалізації та швидкий у навчанні, не потребує визначення додаткових параметрів, підтримує велику кількість класів класифікації, легко розширити кількість класів; дозволяє уникнути перенавчання.

Недоліки: необхідна дуже велика навчальна вибірка; результати експериментів, наведені у статті, не показують високу точність; погані результати при наявності декількох схожих класів.

Bag of words. Підхід для класифікації запропонований у роботі «Architectural Style Classification of Building Facade Windows» [8] полягає у виділенні текстур та форм зображення. Мета – виділити характерні особливості, такі як загострені арки, трикутні або круглі форми, при цьому мінімізуючи вплив нерелевантних для аналізу елементів. При навчанні, кожен знайдений елемент та ознака проходить кластеризацію, щоб сформувати візуальну лексику – кодову книгу. При аналізі, так само як і при навчанні, виявляються основні ознаки, після чого алгоритм визначає з якою з кодових книг найбільша кількість співпадінь і клас, що відповідає книзі, обирається як переможець.

Переваги: алгоритм має високу точність при класифікації архітектурного стилю окремих структурних елементів будівлі, такі як куполи, вежі, колони.

Недоліки: відображення у вікнах, текстури будівельних матеріалів на стінах, зайві предмети на зображенні мають значний вплив на результат аналізу; підхід був перевірений лише на окремих частинах будівель (вікнах), де кількість ознак невелика у порівнянні з зображенням будівлі в цілому; в роботі вказано, що підхід не дасть гарних результатів при класифікації фасадів будівель у цілому.

Deep Learning Techniques. Для розв’язання задачі класифікації зображень будівель у роботах «Classification of Architectural Heritage Images Using Deep Learning Techniques» [9] та «3D Façade Labeling over Complex Scenarios: A Case Study Using Convolutional Neural Network and Structure-From-Motion» [10] було використано нейронні мережі та методи глибинного навчання. Існує велика

кількість літератури що описує різні методи глибокого навчання для класифікації зображень, як загальні, так і специфічні, такі як медичні зображення, аерофотознімки, номерні знаки та транспортні засоби, класифікація мікроорганізмів, міського середовища та багато іншого. Існує також література, що стосується класифікація зображень за архітектурним стилем з використанням таких прийомів як пошук шаблонів, фільтри Габора та векторний апарат підтримки, алгоритм комп'ютерного бачення, кластеризація та вивчення локальних особливостей, або багаточленна прихована логістична регресія [6]; але за винятком [9,10] відсутні посилання щодо класифікації зображень архітектурної спадщини за допомогою глибокого навчання.

У роботі [9] розглянута класифікація будівель для певних елементів архітектури – колона, дзвіниця, купол і т.п. Моделі побудовані шляхом введення набору навчальних даних, для яких класи попередньо позначені для вивчення алгоритмом. Модель використовується шляхом введення набору вхідних даних, що відрізняються від раніше використовуваних для навчання, що дозволяє моделі прогнозувати класи на основі даних, на яких відбувалося навчання. Для реалізації алгоритму класифікації було використано вибірку зображень, що відносилися до десяти категорій. Жоден критерій не застосовувався для вибору цих категорій; він призначений лише для встановлення початкової бази, щоб згодом розширити кількість категорій, що вивчаються, включаючи ті, які вважаються більш корисними для цього типу завдань.

Дослідження «3D Façade Labeling over Complex Scenarios: A Case Study Using Convolutional Neural Network and Structure-From-Motion» [10] присвяченні визначенню простих фасадних елементів, таких як двері, вікна, балкони та дахи. Складність завдання полягає у нескінченних варіаціях форми, композиції матеріалів та непередбачуваних можливостях оклюзій. Повна методологія методу складається з трьох етапів: контрольована модель CNN для семантичної сегментації; придбання геометрії сцени (3D-реконструкція) через пайплайн SfM, MVS, постобробка та 3D-маркування за допомогою аналізу трасування променів. Створена авторами мережа складається з кодера, з мережею VGG16 та декодера з

архітектурою повністю згорткової мережі (FCN). Кодер з топологічною структурою згорткових шарів VGG16, спочатку складений з 13 згорткових шарів, за якими слідують відповідні шари об'єднання та повністю зв'язані (FC). У алгоритмі FC кодер був замінений на згортковий шар 1×1 , який приймає вихідні дані останнього шару об'єднання (так званий pool5) розміром $39 \times 12 \times 512$ і генерує сегментацію з низькою роздільною здатністю розміром 39×12 . Ця зміна робить мережу меншою та легшою для навчання. Після наведених операцій, FCN декодер приймає матрицю 39×12 як вхід для своїх 3 згорткових шарів, який нарешті виконує операцію підвищення вибірки, що призводить до піксельного прогнозування.

Переваги: методи, що засновані на глибокому навчанні, значно перевершують результати досягнуті усіма іншими представленими методами.

Недоліки: дані роботи не присвячені класифікації за архітектурним стилем.

Отже, проаналізувавши існуючі розв'язання класифікації зображень будівель, можна зробити висновок, що методи, засновані на глибокому навчанні, мають набагато кращі результати, ніж інші існуючі методами. [11]

1.2 Дослідження класифікації за допомогою глибинного навчання

Роки досліджень в газулі маркування (класифікації) зображень принесли результати що є сьогодні «золотим» стандартом для розв'язання задач сегментації та класифікації – це використання Deep-Learning (DL). Глибинне навчання – це галузь машинного навчання, в основі якої лежить набір алгоритмів, що моделюють абстракції даних високого рівня за допомогою різних архітектур з безліччю нелінійних перетворень. Глибинне навчання може бути контрольованим та неконтрольованим.

В останні кілька років глибокі згорткові нейронні мережі та інші подібні варіанти, такі як залишкові мережі, стали одними з найбільш популярних архітектур для класифікації зображень. Галузь комп'ютерного зору набула значного значення для швидкого та масштабованого навчання, яке може

забезпечити чудові результати в розпізнаванні об'єктів, виявлення об'єктів, семантичній сегментації, розпізнаванні дій об'єкта, відстеженні об'єктів та багатьох інших завданнях. Ці моделі досягли значних успіхів у сучасному розпізнаванні малюнків, мови, виявлення предметів, облич та інші.

Глибинне навчання – це одна з галузей машинного навчання, що дозволяє обчислювальним моделям з декількома етапами обробки вивчати подання на кількох рівнях абстракції. Характеристики верхнього рівня походять від характеристик нижчого рівня для формування ієрархічного подання, тому глибинне навчання також називають ієрархічним. Термін "глибокий" відноситься до кількості шарів (етапів) обробки. Якщо сказати коротко, методи глибинного навчання існують для розпізнавання структури у величезному обсязі даних, використовуючи, наприклад, контрольоване навчання з концепцією зворотного поширення. [12]

У 1943 році з'явилася перша штучна нейронна мережа – ANN. Маючи лише кілька зв'язків, автори змогли продемонструвати, як комп'ютер може імітувати процес навчання людини. У 1968 р. Hubel, D.H. та Wiesel, T. N. запропонували пояснення як ссавці візуально сприймають світ, використовуючи шарувату архітектуру нейронів аналогічну структурі у мозку. Потім, у 1989 році, нейронна модель почала привертати увагу не лише завдяки своїм результатам, а й схожості з біологічною візуальною системою, з модулями обробки та відчуття. LeCun, Y. та ін. представили витончену нейронну модель для розпізнавання символів написаних від руки – згорткову нейронну мережу (CNN), саме шляхом послідовних математичних операції згортки на зображенні. З тих пір багатьох інженерів надихає розробка подібних алгоритмів розпізнавання образів в комп'ютерному зорі.

1.3 Дослідження згорткових нейронних мереж (CNN)

Нейронна мережа являє собою алгоритм машинного навчання, який побудований за принципом організації та функціонування біологічних нейронних мереж. Ця концепція виникла в спробі імітувати процеси, що відбуваються в мозку, Уорреном Мак-Каллоком та Уолтером Піттсом у 1943 році.

Нейронні мережі складаються з окремих одиниць, які називаються нейронами. Нейрони розташовані в ряді груп - шарів. Нейрони в кожному шарі з'єднані з нейронами наступного шару. Дані надходять від вхідного шару до вихідного шару вздовж цих шарів. Схема нейронної мережі зображена на рисунку 1.1. Кожен окремий вузол виконує простий математичний розрахунок. Потім він передає свої дані у всі вузли, до яких він підключений.

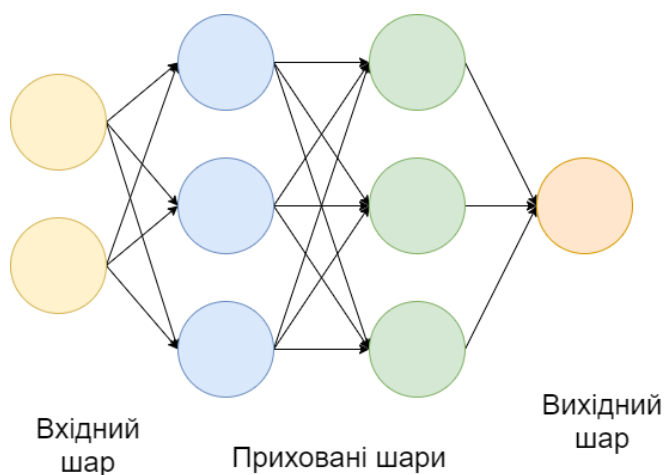


Рисунок 1.1 – Схема простої нейронної мережі

Зі збільшенням обчислювальної потужності та збільшенням досвіду використання нейронних мереж з'явилося глибинне навчання, структури нейронних мереж стали більш складними і здатними вирішувати більш широкий спектр завдань, які раніше не можна було ефективно вирішити за допомогою нейронних мереж. Яскравим прикладом є класифікація зображень.

Згорткові нейронні мережі (convolutional neural networks) – це особлива архітектура нейронних мереж, запропонована Яном ЛеКуном у 1988 році. CNN використовую деякі особливості та механізми зорової кори ока, що робить її найбільш ефективною для роботи з зображеннями серед інших архітектур нейронних мереж. Наприклад, Facebook використовує CNN для алгоритмів автоматичного маркування зображень, Amazon - для генерації рекомендацій щодо продуктів, а Google - для пошуку серед фотографій користувачів.

Основним завданням класифікації зображень є прийняття вхідного зображення та наступне визначення його класу. Це навичка, якій люди вчаться з народження і здатні легко визначити, що зображення на картині - це кіт. Але

комп'ютер бачить зображення зовсім інакше. На рисунку 1.2 зображена різниця як сприймає зображення око, а як комп'ютер: замість зображення комп'ютер бачить масив пікселів. Наприклад, якщо розмір зображення становить 300 x 300 пікселів, то розмір масиву буде 300x300x3, де 300 – це ширина, наступні 300 – це висота, а 3 - значення каналів RGB. Кожному з цих номерів комп'ютеру присвоюється значення від 0 до 255. Це значення описує інтенсивність пікселя в кожній точці.

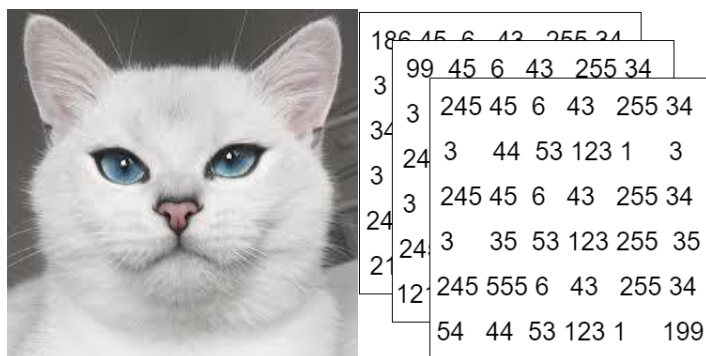


Рисунок 1.2 – Представлення рисунку у пам'яті комп'ютера

Для розв'язання цієї проблеми комп'ютер шукає характеристики базового рівня. У розумінні людиною такими характеристиками є, наприклад, очі або вуха. Для комп'ютера ці характеристики є межами або кривизнами. А потім за допомогою груп згорткових шарів комп'ютер будує більш абстрактні поняття. Зображення проходить через ряд згорткових, нелінійних, об'єднуючих шарів та повністю зв'язаних шарів, а потім генерує вихідні дані.

Convolution layer. Спочатку завжди йде шар згортки. На вхід подається зображення (матриця зі значеннями пікселів). Уявіть, що зчитування вхідної матриці починається у верхньому лівому куті зображення. Далі програмне забезпечення вибирає там меншу матрицю, яка називається фільтром (або нейроном, або ядром). Потім фільтр робить згортку, тобто рухається вздовж вхідного зображення. Завдання фільтра - помножити його значення на початкові значення пікселів. Всі ці множення підсумовуються. У результаті виходить одне число. Оскільки фільтр розпочав обробку зображення у верхньому лівому куті, він рухається вправо на 1 одиницю і виконує подібну операцію. Після передачі фільтра по всіх положеннях виходить нова матриця, менша за вхідну, як зображено на

рисунку 1.3. Зазвичай, мережа складається з кількох згорткових шарів, змішаних з нелінійними та об'єднуючими шарами. Коли зображення проходить через один шар згортки, вихід першого шару стає входним для другого шару. І це повторюється для кожного згорткового шару мережі.

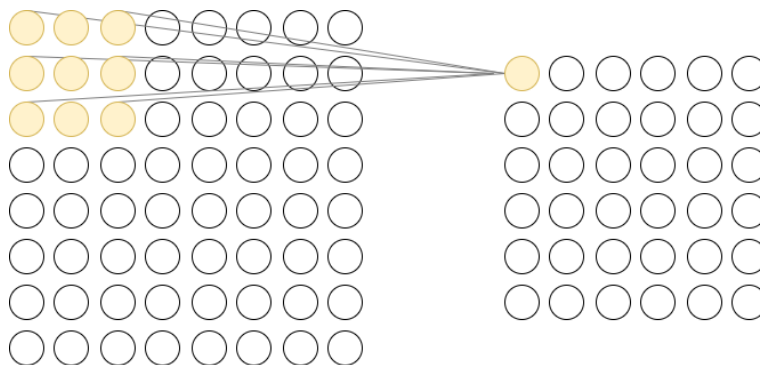


Рисунок 1.3 – Процес згортки

Ця операція аналогічна визначенню меж та простих кольорів на зображенні оком людини. Але для того, щоб розпізнати такі властивості вищого рівня, як око або вуха, потрібні всі шари мережі.

Nonlinear layer. Нелінійний шар додається після кожної операції згортки. Він має функцію активації, яка приносить нелінійні властивості. Без цієї властивості мережа була б недостатньо «розумною» і не змогла б моделювати відповідь (мітку класу).

Pooling layer. За нелінійним шаром слідує шар об'єднання. Він працює з шириною та висотою зображення та виконує на них операцію зменшення дискретизації. В результаті зменшується об'єм зображення. Це означає, що якщо деякі ознаки (наприклад, межі) вже були ідентифіковані в попередній операції згортки, то детальне зображення більше не потрібно для подальшої обробки, і воно стискається до менш детального.

Ці шари достатньо прості – їх задача полягає у розділенні зображення на регіони, на кожному з яких виконується певна операція, вихід якої є елементом результуючого зображення. Існує декілька варіантів операцій, що можна виконати

на регіонах, найбільш популярні – average, global та maximum pooling. Розглянемо детальніше кожен з них.

Average Pooling – це операція об'єднання, при якій розраховується середнє значення регіону пікселів в кожній частині зображення.

Max Pooling – це операція об'єднання, при якій обирається найбільше значення регіону пікселів в кожній частині зображення. Використовується, щоб акцентувати наявність якоїсь ознаки в регіоні. На практиці, Max Pooling дає набагато кращі результати ніж Average Pooling. Причина полягає в тому, що об'єкти, як правило, кодують просторову присутність якогось візерунка або концепції на різних плитках карти об'єктів (у декількох вимірах матриці зображення) і більш інформативним є розгляд максимальної присутності різних об'єктів ніж їх середньої присутності.

Global Pooling – це операція об'єднання, при якій розрахунок проводиться не на окремих регіонах, а на зображенні в цілому. Така операція використовується коли необхідно «агресивно» виділити наявність якоїсь ознаки на зображенні, наприклад, вертикальної лінії, що проходить через все зображення. Інколи Global Pooling шар використовується замість Fully connected layer і передає результати одразу до виходу моделі.

Fully connected layer. Після згорткових, нелінійних та об'єднуючих шарів зображення потрапляє на повністю пов'язаний шар. Цей рівень приймає вихідну інформацію із згорткових мереж. Повністю підключений шар зводить зображення до N-мірного вектора, де N - кількість класів, з яких модель вибирає бажаний клас. На повністю пов'язаному шарі і завершується операція класифікації. [13]

1.4 Дослідження архітектур згорткових нейронних мереж

В останні роки з'явилося багато різних архітектур згорткових нейронних мереж. Збільшення обчислювальної потужності дозволило розробляти настільки глибокі моделі, що їх стало неможливо повністю візуалізувати і більшість людей розглядають їх як «чорний ящик». Проте між архітектурами є різниця і для

розв'язання поставленої задачі необхідно розуміти її. Розглянемо найбільш відомі архітектури CNN.

VGG Net. Ця група архітектур вважається найпростішою серед інших, проте для певних задач вона перевищує за точністю більшість складних архітектур. Існує 6 архітектур VGGNet, найбільш популярні VGG-16 та VGG-19. Ідея VGG полягає у збільшенні розміру фільтру у кожному наступному згортковому шарі, тобто якщо шар 1 має 16 фільтрів, то шар 2 повинен мати 16 або більше фільтрів. Ще однією особливістю є те, що у VGG усі фільтри мають розмір 3x3. Ідея в тому, що два фільтри 3x3 майже покривають те ж, що й фільтр 5x5, проте за загальною кількістю операцій множення вони є «дешевшими».

Недоліками цієї архітектури є те, що вона не підходить для глибоких мереж, оскільки чим глибша мережа, тим більше вона схильна до проблеми зникаючих градієнтів. Для створення глибокої VGG необхідно набагато більше навчання та параметрів. Проте, VGG Net ідеально підходять для Transfer Learning та невеликих задач класифікації. [14]

ResNet. Залишкові мережі - це перші дійсно глибокі мережі – 152 шари, які показали гарні результати. До цієї архітектури навчання глибоких мереж не давало задовільних результатів. Особливістю ResNet для навчання такої глибокої (152 шари) мережі полягає в тому, що вона має залишкові зв'язки. У VGG кожен шар з'єднаний зі своїм попереднім шаром, від якого він отримує свої вхідні дані.

При поширенні від шару до шару, найбільш корисні функції передаються, проте менш важливі функції губляться. Через це, останні шари не отримують ту інформацію, що отримали перші шари. ResNet вирішує цю проблему шляхом підключення не лише попереднього шару до поточного шару, але й шару перед попереднім шаром, як показано на рисунку 1.4. Завдяки цьому, шар може бачити не лише спостереження попереднього шару.

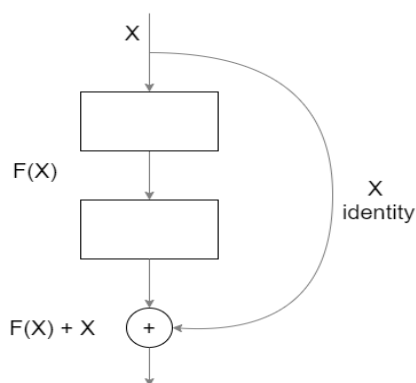


Рисунок 1.4 – Підключення декількох шарів в архітектурі ResNet

Навчання таких глибоких залишкових мереж можливо за допомогою використання шарів нормалізації після кожного згорткового шару. Шари нормалізації підсилюють значення ваг, а отже можна досягнути більшої точності під час навчання, що дозволяє прискорити навчання та мінімізувати проблему зникаючих градієнтів.

Існує багато варіацій ResNet, але основна ідея полягає у використанні вихідних даних з згорткового шару Conv2D як вхідного параметра x . Необхідно застосувати стопку згорткових шарів Conv2D до вхідного параметра x , а потім додати вихідні дані до x і надсилати це на вхід для наступного шару.

Перевага ResNet полягає в тому, що ми можемо навчити більш глибокі мережі за допомогою цієї архітектури. [15]

DenseNet. У ResNet результат з стопки згорткових шарів додавався до вхідного шару, в той час як у DenseNet для кожного поточного шару усі інші шари, що йдуть перед ним, з'єднанні з поточним шаром. При такому зв'язку шарів можливо зменшити кількість фільтрів, що мінімізує проблему зникаючого градієнта оскільки всі шари безпосередньо підключені до вихідних даних, градієнти можуть бути обчислені безпосередньо з вихідних даних для кожного шару.

DenseNet має набагато більше проміжних зв'язків у порівнянні з ResNet. Крім того, це дозволяє використовувати меншу кількість фільтрів у Dense шарах, що є великою перевагою для малих моделей. [16]

Inception Net. Ідея Inception Net полягає в тому, щоб зробити мережу ширшою. Це можна зробити шляхом паралельного з'єднання декількох шарів, що мають різні фільтри з подальшим об'єднанням усіх цих паралельних шарів для переходу до наступних шарів, як показано на рисунку 1.5.

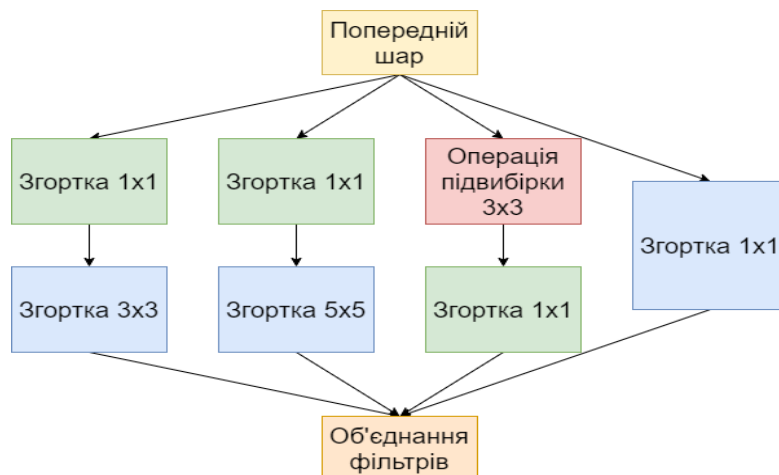


Рисунок 1.5 – Модуль архітектури Inception для зменшення розмірності

Існує безліч варіантів Inception Net, з наступними основними відмінностями:

- використання фільтрів 3x3 замість 5x5 для підвищення ефективності обчислювань (як у VGGNet);
- використання шару згортки Conv2D 1x1 з меншим числом фільтрів перед виконанням будь-якого шару згортки Conv2D з більшими розмірами фільтрів, оскільки фільтр 1x1 зменшить глибину вхідних даних і підвищить ефективність обчислення;
- замість фільтру 3x3, використати фільтр 1x3, а потім фільтр 3x1. Це також суттєво покращить обчислювальну ефективність.

Inception Net включає в себе майже усі переваги вище розглянутих архітектур і більше, оскільки такі мережі не просто глибші, а й ширші, і при додавання багатьох шарів кількість вихідних параметрів для навчання буде меншою, у порівнянні з іншими архітектурами. [17]

Xception Net. Xception Net це модифікація InceptionNet з точки зору обчислювальної ефективності. Різниця полягає в тому, що в Inception Net

виконуються звичайні згорткові операції, тоді як у Xception Net виконуються глибоко відокремлені згорткові (Depthwise Separable Convolutional) операції.

Глибоко відокремлені згортки відрізняються від звичайних згорток тим, що в звичайному шарі Conv2D для зображення розмірності, наприклад, $32 \times 32 \times 3$ ми можемо використовувати будь-яку кількість фільтрів на рівні згортки і кожен із цих фільтрів буде працювати за всіма трьома каналами, а вихідний результат - це сума всіх відповідних значень. Але в глибоко відокремлюваних згортках кожен канал має лише одне ядро для здійснення згортки. Отже, виконуючи глибоко відокремлювані згортки, ми можемо зменшити обчислювальну складність, оскільки кожне ядро є лише двовимірним і здійснює згортку лише за одним каналом.

Xception Net є гарною альтернативою для пристроїв з низькою потужністю, оскільки в рівні згортки менше обчислень, проте точність досить схожа в порівнянні зі звичайним шаром згортки. [18]

1.5 Алгоритми зміни атрибутів нейронної мережі

Під час навчання нейронної мережі відбувається коригування ваг та порогової функції за допомогою ітераційного процесу – зміна значень ваг, зміщення та швидкості навчання для мінімізації функції втрат.

Для знаходження мінімуму функції можна використати існуючі методи, переваги та недоліки розглянуті нижче.

Гرادієнтний спуск - це алгоритм оптимізації, що використовується при навчанні нейронної мережі для пошуку локального мінімуму диференційованої функції. Він базується на ітеративному налаштування параметрів опуклої функції для знаходження її локального мінімуму. Алгоритм є популярним завдяки простоті обчислень та легкості реалізації, проте він має суттєві недоліки: існує висока ймовірність зупинитися в локальному мінімуму; ваги мережі оновлюються після обчислення градієнта для всього набору вхідних даних, тому для великого об'єму даних знаходження мінімуму може зайняти дуже багато часу; обчислення потребують багато пам'яті для обчислення градієнта.

Частину недоліків алгоритму градієнтного спуску вирішує алгоритм стохастичного градієнтного спуску. У цьому алгоритмі похідна обчислюється для кожної точки окремо, на відміну від градієнтного спуску, що потребує обчислення для усіх точок одночасно. Проте, це створює іншу проблему - час обчислень для 1 епохи навчання більший за градієнтний спуск.

Існує інша модифікація - стохастичний градієнтний спуск з імпульсом – він має всі переваги алгоритму стохастичного градієнтного спуску та збігається швидше, ніж алгоритм градієнтного спуску. Проте, на кожній ітерації потрібно обчислити ще одну змінну для кожного оновлення. Алгоритм стохастичного градієнтного спуску варто використовувати лише для неглибоких нейронних мереж.

Вище розглянуті алгоритми ніяк не впливають на такий параметр нейронної мережі як швидкість навчання, тому розглянемо AdaGrad – алгоритм, що працює з адаптивним рівнем навчання для кожного з ваг. AdaGrad коригує загальну швидкість навчання на кожному часовому кроці для кожного параметра на основі значень минулих градієнтів. Головна перевага - не потрібно встановлювати швидкість навчання вручну, оскільки вона адаптивно змінюється з ітераціями. Основний недолік – це накопичення значень квадратних градієнтів, їх сума постійно зростає під час навчання, внаслідок чого швидкість навчання постійно зменшується і може бути зведена до нуля.

Проблему зменшення швидкості навчання вирішує алгоритм AdaDelta - замість зберігання суми попередніх квадратних градієнтів, вона рекурсивно обчислюється як середнє усіх минулих квадратних градієнтів. Існує алгоритм ідентичний AdaDelta – це RMSprop, їх розробили майже одночасно для розв'язання недоліків AdaGrad.

Найбільш популярним алгоритмом є Adam – поєднання RMSprop та стохастичного градієнтного спуску з імпульсом. Він має переваги обох алгоритмів і має найменшу вартість навчання. Проте є один недолік – якщо градієнт певної ваги великий (тобто значення функції втрат сильно змінюється), то алгоритм не такий ефективний, як стохастичний градієнтний спуск.[19]

1.6 Постановка задачі

Проблема класифікації зображень є дуже актуальною в наш час. Кількість даних постійно зростає, технології цифрової фотографії дозволяють мати камеру в кожному телефоні. При великій кількості цифрових зображень постає питання про їх автоматизований аналіз.

Розробка призначена для прогнозування архітектурних стилів будівель за зображенням різного формату та якості.

Метою розробки є спрощення процесу визначення архітектурного стилю будівлі та зниження похибки при визначенні стилю завдяки аналізу отриманого зображення.

Завдяки розробленому алгоритму та створеній системі визначенню архітектурного стилю будівлі документування архітектурної спадщини може бути автоматизований, що зменшить кількість часу, необхідну для документування та підвищить достовірність результатів, у порівнянні з мануальним визначенням архітектурних стилів. Це сприяє збереженню, підтримці в належному стані та відновленню будівель завдяки можливості дослідження еволюції архітектурної споруди.

Також, задача класифікації за архітектурним стилем має певні особливості, що дозволить провести дослідження щодо класифікації зображень при наявності тісних міжкласових взаємозв'язках.

Отже, необхідно створити алгоритмічне забезпечення, що зможе ефективно класифікувати зображення будівель. Алгоритм має враховувати наявність тісних міжкласових взаємозв'язків та мати мінімальну похибку.

Для досягнення поставленої мети мають бути вирішені такі завдання:

- провести аналіз методів класифікації зображень;
- модифікувати існуючі методи для покращення точності та зменшення часу навчання нейронної мережі;
- створити вибірку зображень будівель розподілених за різними архітектурними стилями;

- реалізувати алгоритм програмно та розробити систему аналізу архітектурного стилю;
- порівняти ефективність модифікованого алгоритму з вже існуючими для розв’язання задачі визначення архітектурного стилю за зображеннями будівлі.

Висновки до розділу

Класифікація зображень є фундаментальним завданням, яке намагається зрозуміти, аналізувати зображення як ціле. Мета - аналіз зображення та присвоєння його певній мітці.

Було проведено аналіз та порівняння існуючих алгоритмів для розв’язання задачі розпізнавання архітектурних стилів будівель та описано їх особливості. Проте, на даний момент систем, що можуть визначити архітектурний стиль, не було знайдено. В останні роки з’явилося багато різних архітектур згорткових нейронних мереж, що є найбільш популярними в наш час, проте не було знайдено жодного дослідження присвяченого класифікації за архітектурним стилем.

Задача класифікації архітектурного стилю будівель відрізняється від стандартних задач класифікації наявністю міжкласових взаємозв’язків між різними стилями, тобто велика кількість рис та ознак є однаковою для декількох стилів.

Підчас дослідження літератури, яка відповідає класифікації зображень будівель за архітектурним стилем, було знайдено багато дійсно гарних підходів для розв’язання поставлених задач. Наприклад, алгоритм, розроблений Z. Xu та представлений у роботі «Architectural Style Classification using Multinomial Latent Logistic Regression» [6], враховує навіть маленькі деталі на зображенні, що впливають на результат класифікації. Алгоритм вміє працювати з великою кількістю класів, адже розроблявся з використанням якісної вибірки зображень будівель 25 архітектурних стилів.

Проаналізувавши рішення класифікації зображень будівель «Classification of Architectural Heritage Images Using Deep Learning Techniques» [9] та «3D Façade

Labeling over Complex Scenarios: A Case Study Using Convolutional Neural Network and Structure-From-Motion» [10], можна зробити висновок, що методи, засновані на глибокому навчанні, значно перевершують досягнуті результати за всіма іншими представленими методами.

Проте, існуючі методи, описані в літературі, або зовсім не розглядають задачу класифікації за архітектурним стилем, а одже не враховували особливості цієї задачі, або ж алгоритми не підходять для зображень архітектури, багатих на деталі.

Розглянуті існуючі методи зміни атрибутів нейронної мережі мають недоліки, що є суттєвими для розв'язання задачі класифікації зображень, так як задача потребує ефективної роботи з обчисленнями і часом навчання, тому що зображення є «важкими» даними.

Отже, зазначені недоліки є вагомими і роблять неможливим ефективне застосування існуючих методів для розв'язання задачі класифікації зображень будівель за архітектурним стилем. Проте ідеї, описані в дослідженій літературі, є корисними і можуть бути застосовані при розробці алгоритму для вирішення поставленої проблеми.

2 РОЗРОБКА АЛГОРИТМУ КЛАСИФІКАЦІЇ ЗОБРАЖЕНЬ

2.1 Опис загальних понять для розв'язання задачі

Для розв'язання задачі розпізнавання архітектурного стилю будівлі, тобто класифікації зображення, використаємо згорткові нейронні мережі – нейронні мережі з попередньою обробкою зображення.

Згорткові нейронні мережі – це така архітектура нейронних мереж, що працює аналогічно механізму зорової кори ока, що робить цю архітектуру найбільш ефективною для роботи з зображеннями. Згорткові нейронні мережі імітують роботу мозку людини – так само, як мозок сприймає вхідні дані, оброблює їх і генерує певний результат, так і працює ця архітектурна нейронних мереж. Отримання вхідних даних, обробка отриманої інформації та генерування вихідних даних представлені у вигляді шарів нейронної мережі - вхідного, прихованих та вихідного. Отже, зображення передається через згорткові, нелінійні, об'єднуючі та повністю пов'язані шари, на виході з яких отримуємо вектор з заданих для навчання класів.

Зображення у пам'яті комп'ютера. Зображення представлене в пам'яті комп'ютера як матриця розміру $n \times n \times 3$ - де 3 відповідає трьом кольоровим каналам - червоному, синьому та зеленому, матриця містить значення від 0 до 255 – кількість певного кольору на заданому пікселі (RGB). Такий підхід є ефективним для зберігання інформації про зображення (256 значень ідеально вписуються в один байт) та є достатнім для людського ока, що виділяє обмежену кількість відтінків одного кольору.

Обробка зображення. Алгоритм CNN включає в себе процеси згортки та об'єднання, що представлені у вигляді шарів нейронної мережі. Після прогону матриці зображення через шари згортки та об'єднання зображення достатньо зменшене в розмірах і готове до подальшого аналізу. Матриця потрапляє на «fully connected layer», де відбувається проекція матриці зображення на вектор вихідного

шару. Саме така структура мережі дає можливість аналізувати зображення різних розмірів та якості. [20]

Повний навчальний процес нейронної мережі включає два етапи: пряме і зворотнє поширення.

Пряме поширення. Зображення подається на вхідний шар мережі у вигляді матриці цифр, кожна з яких є інтенсивністю пікселів на зображенні. Нейрони мережі в прихованих шарах застосовують певні математичні операції над цими значеннями (детальний опис операцій наведений нижче). Для виконання математичних операцій існують певні значення параметрів, які випадковим чином встановлюються на початку навчання. Результат виконання передається на вихідний рівень, який генерує остаточний результат класифікації.

Зворотнє поширення. Після отримання результату, його необхідно порівняти з вірним значенням щоб визначити наскільки воно близьке до фактичного значення, тобто обрахувати помилку, або ж втрату. В залежності від значення помилки параметри мережі оновлюються. Процес прямого поширення повторюється з використанням нових значень параметрів і генеруються нові результати до тих пір, поки помилка не стане мінімальною або задана кількість ітерацій не буде досягнута.

Гіперпараметри нейронної мережі. Важливим етапом формування нейронної мережі є вибір параметрів обраної архітектури: кількість шарів, кількість нейронів у кожному шарі, функція активації, значення швидкості навчання та кількість епох навчання. Функція активації є одним з ключових елементів нейронної мережі: без неї нейронна мережа стала б комбінацією лінійних функцій. Елемент нелінійності забезпечує гнучкість для створення складних функцій під час навчального процесу. Функція активації суттєво впливає на швидкість навчання, що є одним з головних критеріїв її вибору.

2.2 Опис процесу виділення кордонів та меж

Операція згортки аналогічна процесу виділення людиною кордонів та меж предметів на зображенні. Завданням шару згортки є зменшити зображення, не втративши важливу для аналізу інформацію.

2.2.1 Згортка зображення

Для згортки зображення обирається матриця, що є частиною загальної матриці – фільтр або ядро. З цієї матриці шляхом множення значень фільтру на початкові значення пікселів отримується згортка. Значення згортки Z обчислюються за формулою (2.1), де X – вхідне зображення позначається, f – фільтр, m та n – індекси рядків та стовпців результуючої матриці:

$$Z_{m,n} = (X * f)_{m,n} = \sum_j \sum_k f_{j,k} X_{m-j, n-k}. \quad (2.1)$$

Згортку часто математично зображають зірочкою *: $Z = X * f$.

Фільтр застосовують для частин зображення, виконуючи множення для кожного елементу, значення яких підсумовуються і в результаті на виході отримують одне число. Операція повторюється для кожного фільтру, що отримується переміщенням вбік або вниз на s одиниць – крок, виконуючи подібну операцію. Після проходження фільтру по всіх позиціях отримують нову меншу матрицю. Оскільки зображення зменшується кожного разу при виконанні згортки, зробити її можливо обмежену кількість разів, перш ніж зображення повністю зникне. Щобільше, коли фільтр рухається по зображенню, вплив пікселів, розташованих на околиці зображення, значно менший, ніж у центрі зображення. Щоб виправити цю проблему, додамо відступ p , значення якого обчислюємо за формулою (2.2), де f – розмірність фільтру:

$$p = \frac{f-1}{2}. \quad (2.2)$$

Розмірність вихідної матриці n_{out} з урахуванням відступу p та кроку s для вхідної матриці розмірності n_{in} можна обчислити, використовуючи формулу (2.3):

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - f}{s} + 1 \right\rfloor. \quad (2.3)$$

Для обробки кольорових зображень існує декілька обмежень: фільтр і зображення, до якого він застосовується, повинні мати однакову кількість каналів; щоб використовувати кілька фільтрів на одному і тому ж зображенні, згортка має проводитися для кожного з них окремо, результати накладаються один на інший і об'єднуються в одне ціле. Розмірність вихідної 3D матриці обчислюється за формулою (2.4):

$$[n, n, n_c] * [f, f, f_c] = \left[\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor, \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor, n_f \right], \quad (2.4)$$

де n - розмір зображення, f - розмір фільтра, n_c - кількість каналів на зображенні, p - відступ, s - крок, n_f - кількість фільтрів. [21]

2.2.2 Шари згортки у нейронній мережі

Шари згортки виконують наступні дії над зображенням:

- Для поточного шару згортки обчислюється значення згортки вхідного зображення з використанням результату згортки з попереднього шару, до якого додається зміщення.
- Застосування нелінійної функції активації до проміжного значення.

Ці дії забезпечують нелінійність та покращують результати навчання.

2.2.3 Зворотнє поширення у шарах згортки

Мета зворотнього поширення у шарах згортки така сама, як і в повністю зв'язаних шарах – розрахувати похідні, що будуть використовувати для оновлення значень параметрів у шарах згортки. Процес зворотнього поширення аналогічний процесу, описаного у пункті 2.4.

2.3 Процес зменшення зображення

Для забезпечення ефективної обробки алгоритму зображень великого розміру з великими зображеннями, додамо шари, що додатково зменшать зображення. Так як основні ознаки зображення ідентифіковані після операції згортки, велике детальне зображення більше не потрібно для подальшої обробки.

Об'єднуючі шари дозволяють скоригувати розміри матриці пікселів – зменшать розмірність стовбців та рядків.

Ця операція дозволить навчати нейронну мережу на вибірці великих зображеннями без шкоди для швидкості навчання.

2.3.1 Шари об'єднання

Для зменшення зображення використаємо підхід max pooling – матриця розбивається на частини $X_{k,o}$, в кожній з яких обирається максимальне значення і розміщується у відповідному місці у вихідній матриці $P_{i,j}$. Як і у випадку із шаром згортки, застосовуються розмір фільтра f_2 . При виконанні об'єднання для багатоканального зображення, об'єднання для кожного каналу здійснюється окремо. [22]

$$P_{i,j} = \max X_{k,o},$$

$$k = \overline{i \dots i + f_2}, \quad o = \overline{j \dots j + f_2}.$$

2.3.2 Зворотнє поширення у шарах об'єднання

В шарі об'єднання немає параметрів, які необхідно оновлювати, тому потрібно лише розподілити градієнти належним чином. При прямому поширенні для максимального об'єднання обирається максимальне значення з кожної області і передається на наступний шар. Тому під час зворотного поширення градієнт не повинен впливати на елементи матриці, які не були включені в прямий прохід. На практиці це досягається створенням маски, яка запам'ятовує положення значень, використаних на першому етапі, яка згодом може бути використана для передачі градієнтів.

2.4 Опис процесу навчання нейронної мережі

Навчання нейронної мережі полягає у коригуванні параметрів нейронної мережі - ваг та швидкості навчання за допомогою ітераційного процесу таким чином, щоб функція втрат була зведена до мінімуму. Щоб знайти мінімум функції

можна використати різні методи описані в пункті 1.5, проте вони мають недоліки, що є суттєвими для розв'язання задачі класифікації зображень.

На кожній ітерації t необхідно обчислити градієнт функції втрат що покаже напрямок найшвидшого зростання функції втрат $grad(L)$ і перерахувати ваги мережі W враховуючи швидкість навчання α .

$$g_t = grad(L), \quad (2.5)$$

$$W_t = W_{t-1} - \alpha g_t, \quad (2.6)$$

де α - швидкість навчання, $grad$ – градієнт функції

Вибір швидкості навчання α є вирішальним – якщо встановити занадто низьку, то нейронна мережа буде вчитися дуже повільно, якщо встановити занадто високою, то можна не досягти мінімуму. Для визначення швидкості навчання необхідно провести велику кількість експериментів на даних, що будуть використовуватися для навчання мережі, що може зайняти роки, тому пропонується використовувати адаптивну швидкість навчання для кожної ітерації.

Змінюючи ваги за формулою (2.6) ми знайдемо локальний мінімум функції втрат, проте не маємо можливості контролювати зміни ваг – робити більші «кроки» коли градієнт не змінюється на велике значення, щоб пришвидшити навчання, або ж навпаки менші щоб не «перестрибнути» мінімум. Для цього при обчисленні градієнту, обчислимо наступні значення ковзне середнє градієнту та його квадратів:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad (2.7),$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \quad (2.8).$$

де β_1 та β_2 – параметри, що контролюють вплив минулих значень на зміну ваг.

Змінюючи ваги мережі ми контролюємо «крок» домножаючи зміну на швидкість навчання α :

$$W_t = W_{t-1} - \alpha \frac{m_t}{\sqrt{v_t + \varepsilon}}. \quad (2.9)$$

Таким чином, при малій зміні градієнту m_{it} приблизно дорівнює $\sqrt{v_t}$, отже зміна ваг повністю контролюється швидкістю навчання. При різкій зміні градієнту m_{it} значно менше ніж $\sqrt{v_t}$, а значить швидкість навчання зменшується.

При навчанні нейронних мереж важливо мати малі значення W , це дозволить уникнути перенавчання та створити більш узагальнену модель. Існуючі методи вирішують цю проблему додавши доданок μW_t у формулу розрахунку градієнту (2.5), проте це впливає на значення m_{it} та v_{it} – вони будуть відслідковувати не лише зміну градієнту але також і регуляцію ваг мережі. Для вирішення цієї проблеми додамо регуляцію ваг як наведено в (2.10):

$$W_t = W_{t-1} - \alpha \frac{\beta_1 m_{t-1} + (1 - \beta_1)(g_t + \mu W_{t-1})}{\sqrt{v_{t-1} + \epsilon}}. \quad (2.10)$$

В цьому випадку зниження ваги також нормується значенням $\sqrt{v_t}$: при великих значеннях W або великому значенню градієнта (різко змінюється) значення v_t також буде мати велике значення, тому великих змін ваг відбуватися не буде. Аналогічно для малих значень. [23]

Нейронна мережа зазвичай використовує не лише ваги у кожному нейроні, а також зсув, швидкість навчання, параметри для ковзне середнє градієнту та його квадратів. Зміни параметрів аналогічні процесу, розглянутому вище. Загальна схема навчання нейронної мережі наведена на рисунку 2.1.

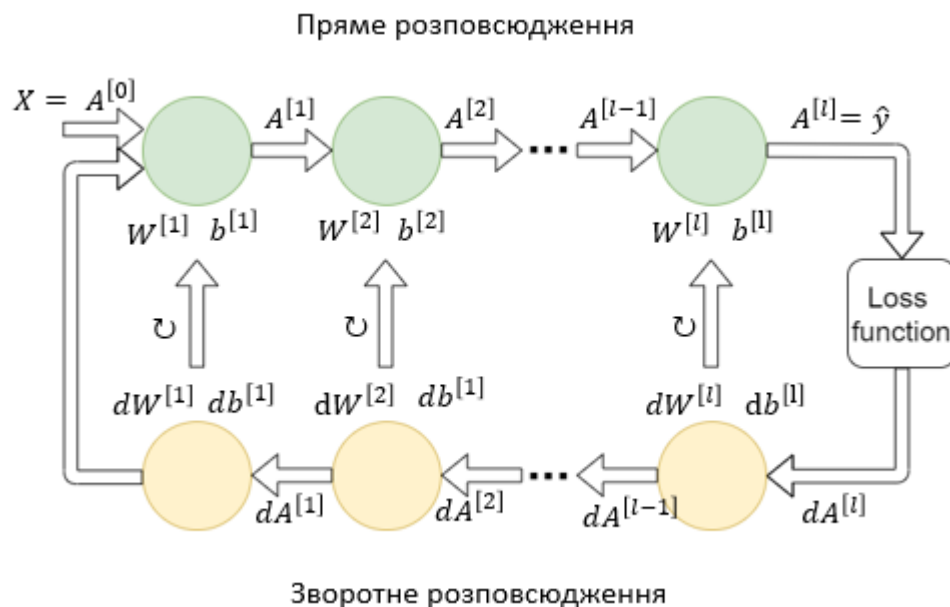


Рисунок 2.1 – Схема прямого і зворотнього поширення

Функція втрат. Функція втрат, функція помилки, вартості навчання показує оцінку, що узагальнює різницю між фактичним та прогнозованим розподілом ймовірностей для усіх класів задачі. Значення показує ефективність моделі

класифікації і становить значення від 0 до 1. Ця оцінка має бути зведена до мінімуму, тобто ідеальне значення перехресної ентропії має дорівнювати 0.

2.5 Математична постановка задачі

Сформулюємо цільову функцію поставленої задачі. Для довільної вибірки зображень, розподілених за s класами, необхідно виконати навчання нейронної мережі так, щоб функція втрат була мінімальною.

Дано:

c_i – клас класифікації, тобто архітектурний стиль;

X_j – зображення будівлі;

N – кількість зображень;

p_j – ймовірність, що зображення X_j належить до класу c_{X_j} ;

Цільова функція розв'язання задачі:

Необхідно мінімізувати значення функції втрат, що відповідає за загально похибку мережі при навчанні. Функція втрат показує, яка частина зображень при використанні мережі може бути класифікована невірно:

$$L = -\frac{1}{N} \sum_{j=1}^N \ln p_j \rightarrow \min. \quad (2.11)$$

Обмеження:

$$i, j, N \in \mathbb{N};$$

$$1 \leq N < \infty;$$

$$2 \leq i < \infty;$$

$$0 \leq L < \infty;$$

2.6 Схема алгоритму розв'язання задачі

2.6.1 Алгоритм навчання

Навчання нейронної мережі полягає у коригуванні ваг та порогової функції за допомогою ітераційного процесу. Розглянемо загальний процес навчання нейронної мережі:

Крок 1: Ініціалізація всіх ваг $W_{uv}^{(l)}$ зв'язків нейронної мережі випадковими значеннями та зміщення b , де $W_{uv}^{(l)}$ – вага зв'язку від нейрону u у $(l - 1)$ шарі до нейрону v у шарі l .

Крок 2: Виконати наступні дії:

Крок 2.1: Дані з навчаючої вибірки встановлюються як значення вхідного шару мережі, а значення результату у вихідний шар.

Крок 2.2: Розрахування значення входу для кожного нейрону у наступному прихованому шарі та застосування функції активації. Повторити для всіх наступних шарів:

$$R^{(l)} = Y^{(l-1)}W^{(l-1)},$$

$$Y^{(l)} = \sigma(X^{(l)}).$$

Крок 3: Зворотний прогін:

Крок 3.1: Розраховується матриця градієнтів з предостаннього шару до останнього – вихідного шару ($l = n$).

Крок 3.2: Розраховується значення градієнту на попередніх шарах та оновлюються ваги за формулою 2.10. Процес повторюється доки всі ваги нейронної мережі не будуть оновлені.

Крок 4: Якщо виконали задану кількість ітерацій (епох) або функція втрат досягла мінімуму – завершити процес навчання. Інакше – перейти на крок 2.

2.6.2 Алгоритм класифікації

Алгоритм класифікації використовує вже навчену нейронну мережу з заданими значеннями ваг W та іншими параметрами. Тому при класифікації необхідно лише «пропустити» зображення через усі шари нейронної мережі і інтерпретувати результуючий вектор:

Крок 1: Беремо вхідні дані і встановлюємо значення вхідного шару мережі (шар 0) у значення даних та значення результату у вихідний шар.

Крок 2: Розрахувати значення для входу для кожного нейрону у наступному прихованому шарі $R_j^{(l)}$ та застосувати функцію активації, щоб отримати вихідне значення для наступного прихованого шару. Повторити для всіх наступних шарів:

$$R^{(l)} = Y^{(l-1)}W^{(l-1)},$$

$$Y^{(l)} = \sigma(R^{(l)}).$$

Крок 3: Встановити відповідність між вихідним вектором та назвами класів та повернути результат класифікації. [24]

2.7 Обґрунтування правильності роботи алгоритму

Оскільки під час навчання нейронної мережі є висока ймовірність не досягти мінімуму функції втрат, а зупинитися в локальному мінімуму, або ж прогавити мінімум, алгоритм має дуже обережно змінювати ваги мережі. При занадто малих кроках час навчання буде дуже великим, а при великих – може страждати точність навчання. Для вирішення цієї проблеми було використано адаптивну швидкість навчання – при незначних значеннях градієнту ваги змінюються швидко, для пришвидшення досягнення точки мінімуму, а коли градієнт має велике значення, це означає, що функція втрат починає зменшуватися та варто робити менші кроки, щоб не прогавити мінімум. Така стратегія дозволяє підвищити швидкість пошуку найкращого рішення не впливаючи на точність навчання.

Іншою проблемою навчання нейронної мережі є перенавчання. Для уникнення цієї проблеми необхідно при зміні ваг нейронної мережі тримати їх значення невисокими. Тому кожного разу при зміні ваг буде відбуватися їх регуляція – коригування значень таким чином, щоб значення ваг не збільшувались.

Такий підхід використовують існуючі алгоритми, проте вони додавали регуляцію ваг до формули розрахунку градієнту, що впливало на процес пошуку мінімуму функції втрат та могло привести до вибору невірного напрямку зміни ваг. Для вирішення цієї проблеми регуляція додається у формулу розрахунку нових ваг

мережі вже після обчислення градієнту, що дозволяє тримати значення ваг низькими не впливаючи на процес пошуку мінімуму.

Висновки до розділу

У розділі «Розробка алгоритму класифікації зображень» була сформульовано математичну постановку задачі, цільову функцію задачі та наведено схеми алгоритмів навчання та класифікації зображень. Було розглянуто загальні поняття та підходи, що були використані при розробці алгоритму класифікації та процес навчання нейронної мережі з гіперпараметрами, що використовуються при її створенні. У розділі детально описані методи та процеси виділення кордонів та меж предметів на зображенні, обрані підходи до зменшення зображення. Також, було обґрунтовано правильність роботи алгоритму навчання нейронної мережі.

3 ОПИС ПРОГРАМНОГО ТА ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вимоги до програмного продукту

3.1.1 Вимоги до функціональних характеристик

Система має надавати користувачу наступні функціональні можливості:

- а) створити обліковий запис;
- б) увійти в обліковий запис;
- в) визначити архітектурний стиль будівлі:
 - 1) завантажити зображення;
 - 2) здійснити класифікацію;
 - 3) переглянути результат;
 - 4) додати опис зображення;
- г) переглянути раніше збережені результати класифікації;
- д) переглянути документацію та інструкцію користувача.

3.1.2 Вимоги до надійності

Приватна інформація користувача (пароль) мають бути збережені у зашифрованому вигляді. Дані, введені користувачем, мають бути перевірені на формальну правильність.

3.1.3 Вимоги до складу і параметрів технічних засобів

Мінімальними вимогами до апаратної частини ПК для коректної роботи веб-застосування є:

- а) оперативна пам'ять об'ємом не менше 1 ГБ;
- б) процесор з тактовою частотою не нижче 1 ГГц;
- в) підключення до Інтернету.

3.2 Засоби розробки

При реалізації алгоритму було використано платформу та мову програмування Python, що є найбільш популярною для реалізації алгоритмів

машинного навчання. Ця мова є достатньо простою, проте існує велика кількість модулів, бібліотек, що реалізують різні математичні операції, що дозволяє швидко реалізувати алгоритм. [25]

При розробці нейронної мережі та алгоритму обробки зображення було використано наступні бібліотеки, що використовуються в data science та data analysis: NumPy, SciPy, Scikit-Learn, Matplotlib. Ці бібліотеки широко використовуються для створення алгоритмів машинного навчання і є ефективними. Також, Python має готові реалізації популярних методів такі як класифікація, рекомендації, регресія та кластеризація.

Для створення та навчання нейронної мережі було використано Python версії 3.5.2 та пакетний менеджер pip. Були використані наступні бібліотеки Python:

- h5py – це інтерфейс Pythonic для формату бінарних даних HDF5. Дозволяє зберігати величезну кількість числових даних і легко маніпулювати цими даними з NumPy;
- keras – бібліотека, що містить API для роботи з згортковими нейронними мережами, що може оброблювати зображення на CPU та GPU. Бібліотека використовує TensorFlow;
- matplotlib – бібліотека для побудови графіків функцій, гістограм, діаграм, графіків розсіювання, тощо;
- numpy – бібліотека, що надає можливість використати готові функції, призначені для роботи з багатовимірними масивами;
- opencv-python – неофіційний, але дуже поширений пакет OpenCV для мови програмування Python. Дозволяє працювати з зображеннями і виконувати операції редагування пікселів, геометричні перетворення і т.п.;
- pillow – бібліотека для роботи з растровими зображеннями;
- scipy – бібліотека, що містить функції для модулі для оптимізації, обробки зображень, розв'язування диференціальних рівнянь та багато іншого. Бібліотека розроблюється розробниками Scilab та Matlab;

— tensorflow – бібліотека з відкритим вихідним кодом для створення моделей машинного навчання, розроблена компанією Google.

Було розроблено веб-застосування з використанням фреймворку Angular та з двох мікросервісів, один з яких реалізовано з використанням фреймворку Flask, інший – ASP.NET Core. Для збереження даних користувачів, результати їх класифікацій зображень було використано MsSQL Server.

3.3 Архітектура програмного забезпечення

3.3.1 Діаграма класів

У ході розробки інформаційної системи була створена структура класів доменної області системи наведена на схемі структурній класів представлений у додатку А. Доменна область складається з користувача, результату аналізу зображення, самого зображення та дозволу переглядати збережений результат іншим користувачам. Результат аналізу завжди належить існуючому користувачу, що створив його, і завжди містить посилання на зображення.

3.3.2 Діаграма послідовностей

Схема структурна послідовності інформаційної системи класифікації зображень будівель за архітектурним представлена у додатку А.

На схемі структурній послідовності аутентифікації користувача, представлений у додатку А, зображена схема послідовності аутентифікації зареєстрованого користувача. Користувач має ввести свій логін та пароль, натиснути кнопку «вхід у систему» та після перевірки введених даних користувач буде перенаправлений на домашню сторінку. Після успішної перевірки даних аутентифікації, сервіс генерує токен аутентифікації, що зберігається у браузері. Усі запити до компонентів аналізу та збережених результатів будуть відправлятися з отриманим токеном для перевірки користувача.

Схема структурна послідовності реєстрації користувача представлена у додатку А.

Користувач має заповнити форму реєстрації, де вказує своє ім'я, електронну пошту та пароль, натиснути кнопку «zareestruватися» та після перевірки введених даних користувач буде перенаправлений на домашню сторінку. Після успішного створення нового користувача, сервіс одразу генерує токен аутентифікації, що зберігається у браузері. Це необхідно для уникнення процесу входу в систему після реєстрації.

Схема структурна послідовності аналізу зображення представлена у додатку А. Користувач має завантажити зображення та натиснути кнопку «аналіз». Сторінка аналізу зображення доступна як користувачам, що увійшли в систему, так і анонімним користувачам, проте лише авторизовані користувачі зможуть зберегти результати аналізу та переглянути раніше збережені результати.

3.3.3 Діаграма компонентів

Схема структурна компонентів інформаційної системи класифікації зображень будівель за архітектурним стилем наведено у додатку А. Система складається з 4 компонентів:

- компонент аналізу зображень, що використовує навчену нейронну мережу. Компонент відкритий всім користувачам, не вимагає попередньої реєстрації для використання;
- компонент збережених результатів: дозволяє зберегти результати аналізу та переглянути раніше збережені результати. Компонент доступний лише зареєстрованим користувачам;
- компонент аутентифікації: дозволяє створити нового користувача, отримати аутентифікаційний токен користувача та провести перевірку токена;
- компонент інтерфейсу користувача: містить форми аутентифікації, реєстрації, аналізу зображення, перегляд збережених результатів та документацію.

3.4 Інструкція користувача

Для початку роботи з програмним забезпеченням користувач має перейти на сторінку веб-застосування. Інструкції з розгортання програмного забезпечення в локальному середовищі наведені в інструкціях кодової бази. [26][27][28]

Розглянемо детальніше форму домашньої сторінки користувача, головне меню сайту та вміст сторінки, що доступний незареєстрованому користувачу (рисунок 3.1):

- назва застосування;
- посилання на домашню сторінку;
- посилання на сторінку з документацією;
- посилання на сторінку для визначення архітектурних стилів зображень;
- посилання на сторінку входу в систему.

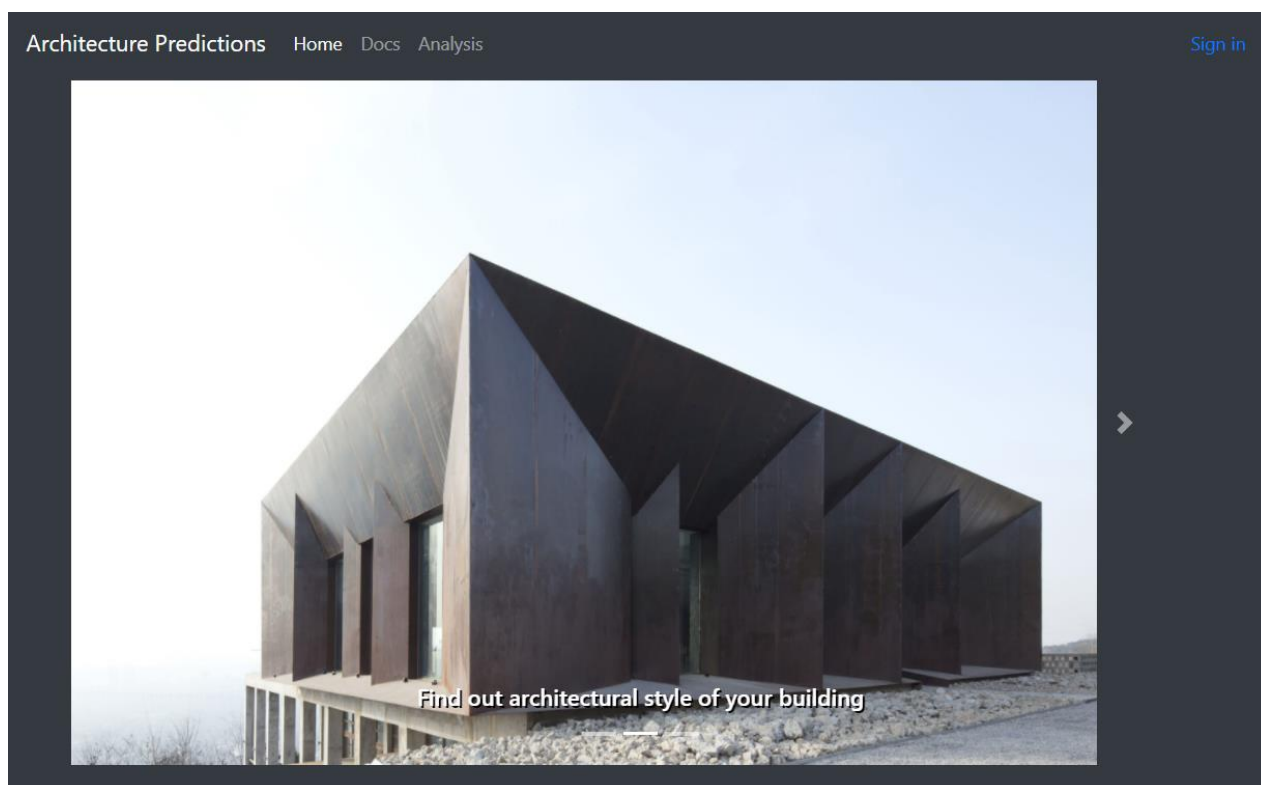
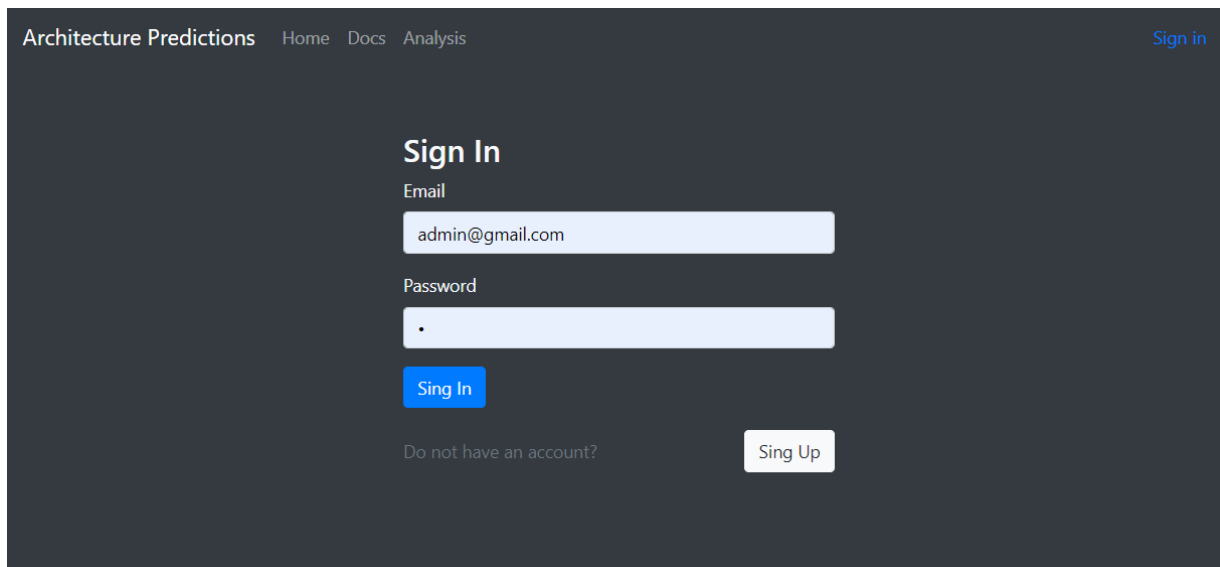


Рисунок 3.1 – Домашня сторінка

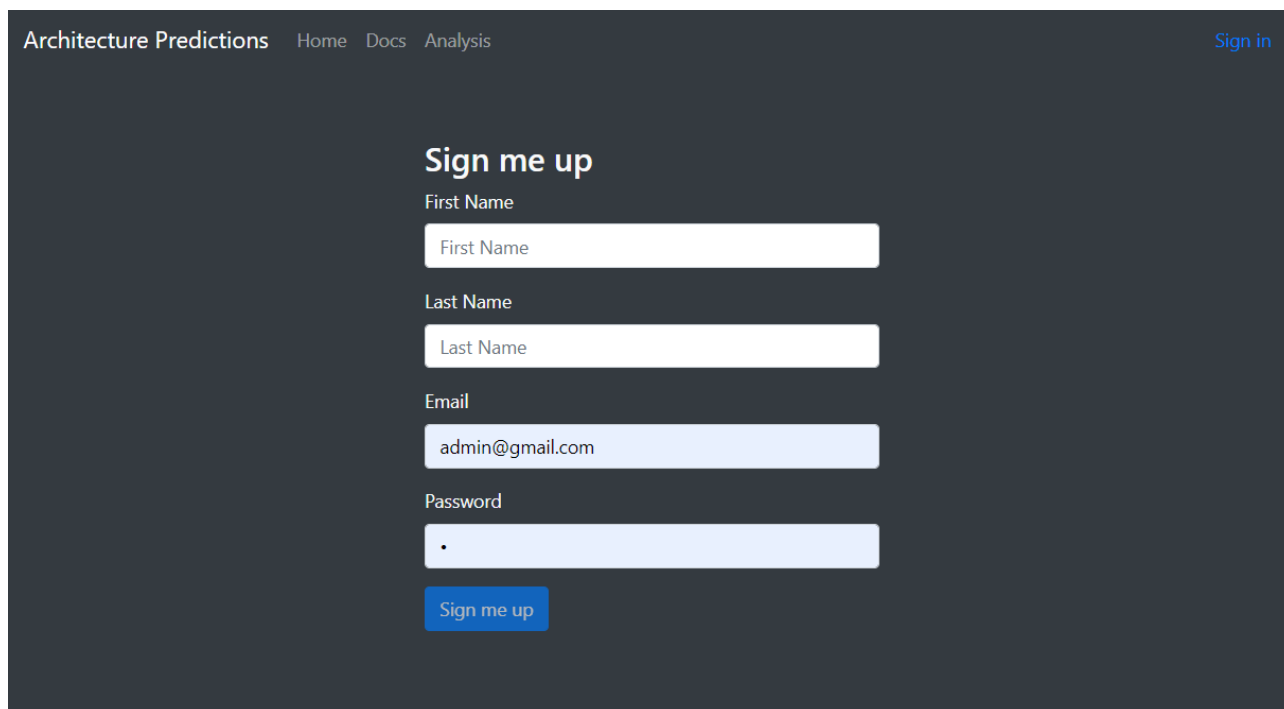
Розглянемо форму входу в систему (рисунок 3.2). Тут користувач має заповнити дані для входу в систему та натиснути кнопку. Якщо користувач не має

облікового запису в системі, він може створити його через форму реєстрації, натиснувши на кнопку зареєструватися (рисунок 3.3).



The screenshot shows a web application interface with a dark background. At the top, there is a navigation bar with the text 'Architecture Predictions' and links for 'Home', 'Docs', and 'Analysis'. In the top right corner, there is a 'Sign in' link. The main content area is titled 'Sign In'. It contains two input fields: 'Email' with the value 'admin@gmail.com' and 'Password' with a single dot. Below the password field is a blue button labeled 'Sing In'. At the bottom of the form, there is a link 'Do not have an account?' and a white button labeled 'Sing Up'.

Рисунок 3.2 – Форма входу в систему



The screenshot shows the same web application interface as Figure 3.2. The main content area is titled 'Sign me up'. It contains four input fields: 'First Name' with the placeholder 'First Name', 'Last Name' with the placeholder 'Last Name', 'Email' with the value 'admin@gmail.com', and 'Password' with a single dot. Below the password field is a blue button labeled 'Sign me up'.

Рисунок 3.3 – Форма реєстрації

Розглянемо сторінку документації (рисунок 3.4), де користувач потрапляє на сторінку з описом архітектурних стилів та результатів.

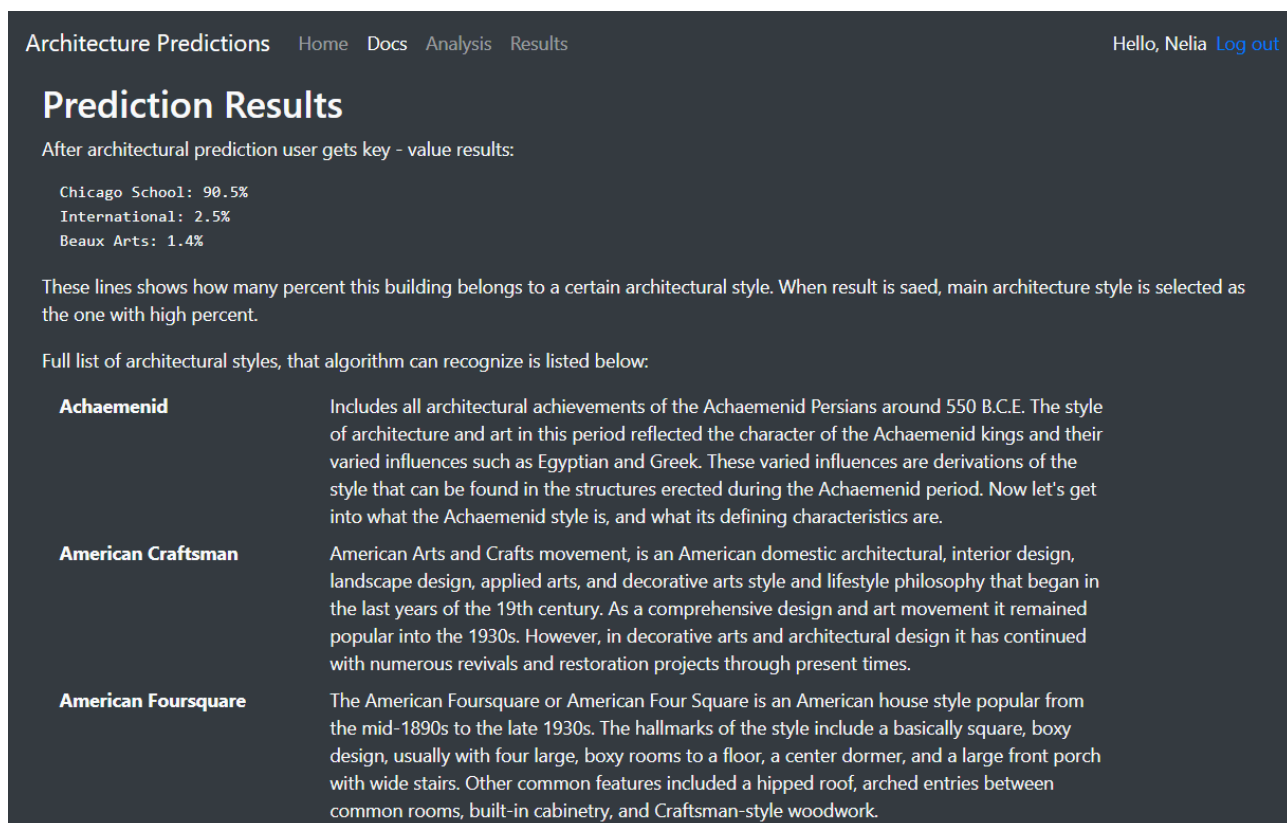


Рисунок 3.4 – Форма документації

Перейдемо на сторінку визначення архітектурного стилю і завантажимо зображення для аналізу (рисунок 3.5). На сторінці наявний компонент для завантаження зображень, де показано інструкції для завантаження файлу або ім'я завантаженого файлу. Поряд розташовані кнопки обрати файл та відправити зображення на аналіз, що є не активною поки зображення не обране. Поряд зображенням після натиску на кнопку аналізу зображення, відображено результати аналізу. Форма та кнопка збереження результатів доступна лише користувачам, що увійшли в систему.

Architecture Predictions [Home](#) [Docs](#) [Analysis](#) [Results](#) Hello, Nelia [Log out](#)

Hi there! Try out Architectural styles prediction by uploading your building image!

[Browse](#) [Predict style](#)



Queen Anne: 71.7%
Tudor Revival: 7.6%
Art Nouveau: 4.1%

Fill in details

Title:

Description:

Date image taken:
 

[Save results](#)

Рисунок 3.5 – Форма документації

Після аналізу та збереження результатів користувач може переглянути їх на формі збережених результатів (рисунок 3.6). Тут відображені результати у вигляді таблиці, що підтримує сортування за деякими полями та видалення результатів.

Architecture Predictions [Home](#) [Docs](#) [Analysis](#) [Results](#) Hello, Nelia [Log out](#)

Saved results

Title ↕	Description	Date ↕	Style ↕	Date Image Taken ↕	Image Preview	
Queen Anne building		4/17/21, 11:59 AM	Queen Anne	4/17/21, 12:00 AM		
test	scfdg	4/17/21, 11:11 AM	Chicago School	4/21/21, 12:00 AM		

Рисунок 3.6 – Форма збережених результатів

3.5 Опис технічного забезпечення

Схема структурна розгортання інформаційної системи класифікації зображень будівель за архітектурним стилем наведено на рисунку схемі структурній розгортання у додатку А.

Так як система складається з 2 мікросервісів та застосування інтерфейсу користувача, то обмін повідомленнями між ними здійснюється через протокол HTTPS.

Схема підготовлена для розгортання системи в середовищі Amazon Web Services з використанням нативних технологій провайдера. Amazon Web Services пропонує широкий набір глобальних хмарних продуктів, включаючи обчислення, зберігання, бази даних, аналітику, мережу, мобільні пристрої, інструменти для розробників, засоби управління, IoT, безпеку та корпоративні програми. Ці послуги допоможуть проекту розвиватися швидше, знижуючи ІТ-витрати та масштаби. AWS довіряють найбільшим підприємствам та найгарячішим стартапам для забезпечення найрізноманітніших навантажень, включаючи: веб- і мобільні додатки, розробку ігор, обробку та зберігання даних, зберігання, архівування та багато інших. [29]

Система складається з 4 компонентів: компонент аналізу зображень, що використовує навчену нейронну мережу; компонент збережених результатів; компонент аутентифікації; компонент інтерфейсу користувача.

В якості системи доменних імен використано Amazon Route 53 для перенаправлення кінцевих користувачів до інтернет-додатків завдяки перетворення доменних імен (наприклад, www.architecture-recognition.com) в формат цифрових IP-адрес (наприклад, 192.0.2.1). Сервіс Amazon Route 53 направляє запити користувачів до інфраструктури AWS, наприклад до інстанси Amazon EC2, балансувальник навантаження Elastic Load Balancing або кошиках Amazon S3. В нашому випадку запит клієнта буде перенаправлено на Amazon CloudFront - сервіс глобальної мережі доставки контенту (CDN), що забезпечує швидку і безпечну передачу даних, відео, додатків і API клієнтам з низькими затримками і високою швидкістю.

CloudFront, в залежності від запиту, перенаправить його до служби зберігання статичних файлів Amazon S3, де зберігаються файли інтерфейсу користувача, або до Elastic Load Balancing.

Elastic Load Balancing автоматично розподіляє вхідний трафік додатків за кількома цілями – контейнерами AWS Fargate. Кожен контейнер являє собою один з компонентів – компонент аналізу зображень, що використовує навчену нейронну мережу; компонент збережених результатів; компонент аутентифікації. AWS Fargate - це безсерверний обчислювальний механізм для контейнерів. Fargate позбавляє потреби надавати сервери та керувати ними, дозволяє вам визначати та оплачувати ресурси за програму, а також покращує безпеку завдяки ізоляції додатків за проектом. Fargate розподіляє потрібну кількість обчислень, усуваючи необхідність вибору екземплярів та масштабування ємності кластера. Fargate запускає кожне завдання або підсистему у своєму власному ядрі, забезпечуючи завдання та підрозділи своїм власним ізольованим обчислювальним середовищем. Це дозволяє вашому додатку мати ізоляцію робочого навантаження та покращений дизайн за проектом.

Для зберігання даних про користувачів та збережені результати використаємо Amazon Aurora Serverless – це служба реляційних баз даних, що підтримує автоматично масштабовану конфігурацію. Вона передбачає автоматичне підключення, відключення та масштабування ресурсів залежно від потреб додатків. Для збереження зображень будівель використаємо Amazon S3 – дешево та надійно збереження файлів.

База даних та Amazon S3 bucket для збереження зображень будівель мають бути розгорнуті в приватній підмережі для уникнення доступу до даних користувачів не авторизованим клієнтам.

Висновки по розділу

У даному розділі було розглянуто технологічні аспекти інформаційної системи класифікації зображень будівель за архітектурним стилем.

Було описано технології, обрані для створення та розгортання інформаційної системи – Python Flask Microframework, .NET Core Web API, СУБД Ms SQL та фреймворк для створення SPA – Angular. Було обґрунтовано вибір технологій та наведені переваги їх використання. Також описано вимоги до технічного забезпечення системи, функціональні вимоги та вимоги до надійності.

За допомогою діаграми послідовності було показано основні шляхи використання системи користувачем – авторизація користувача, аналіз зображення та його зображення.

Було розглянуто керівництво користувача та надані інструкції по розгортанню системи.

Було описано архітектуру системи її у вигляді діаграм класів, компонентів та розгортання з їх детальним описом.

4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Бенчмарк

Щоб перевірити ефективність удосконаленого алгоритму, порівняємо час та результати навчання нейронної мережі на прикладі задачі розпізнавання архітектурного стилю будівлі. Для порівняння, пропонується використати удосконалений алгоритм, Adagrad та RMSprop на архітектурах нейронних мереж ResNet та DenseNet. Для навчання було сформовано навчаючу вибірку з 1000 зображень та тестову з 500 зображень, що містять зображення будівель розподілених за 6 архітектурними стилями.

Навчання було проведене при однакових умовах – апаратне та програмне забезпечення не змінювалось та інших операцій паралельно не проводилось.

Основними критеріями для порівняння було обрано час навчання та значення функції втрат на момент навчання. На рисунку 4.1 зображена зміна функції втрат при навчанні. Удосконалений алгоритм має кращий результат на обох архітектурах ResNet та DenseNet ніж алгоритми Adagrad та RMSprop. Після завершення навчання, модель, навчена з використанням удосконаленого алгоритму, показала в 1.5 рази меншу функцію втрат порівнянно з іншими моделями.

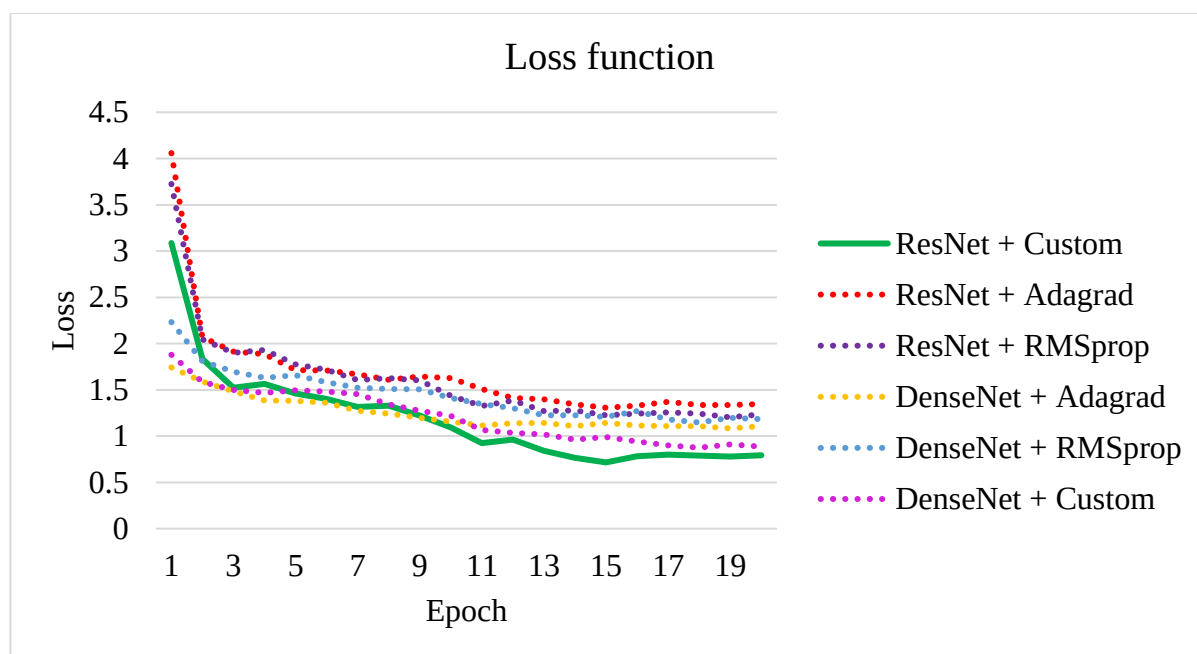


Рисунок 4.1 – Графік залежності функції втрат від епохи

На рисунку 4.2 зображена кількість часу у секундах, необхідна для навчання моделі при 20 епохах. Навчання моделі з використанням удосконаленого алгоритму завершилося в 1.13 раз швидше, ніж з використанням інших алгоритмів. [30]

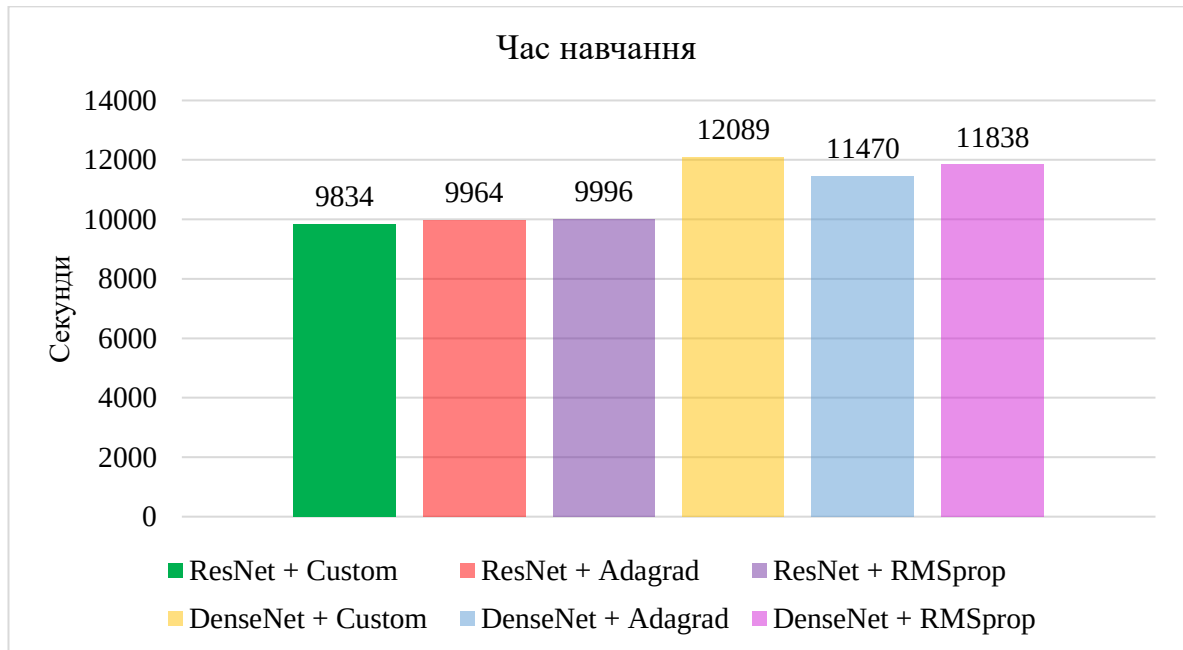


Рисунок 4.2 – Час затрачений на навчання мережі

4.2 Порівняння результатів аналізу зображень

Порівняємо правильність класифікації на навчених алгоритмах використовуючи 12 контрольних зображень будівель для 6 архітектурних стилів.

Результати наведені як ймовірність того, що зображення має певний архітектурний стиль, що показує наскільки навчена мережа впевнена в класифікації.

Правильним результатом є 1, тому що зображення 100% має заданий архітектурний стиль. В таблиці результатів розглядаються значення ймовірностей архітектурного стилю, до якого відноситься дане зображення, інші архітектурні стилі не розглядаються.

Ймовірність правильної класифікації була розрахована як середнє усіх проведених експериментів і показує наскільки навчена мережа впевнено класифікує зображення.

Можна помітити, що ймовірності є доволі низькими, це пов'язано з кількістю епох, обраною для навчання. Для досягнення більшої точності необхідно провести набагато більшу кількість ітерацій навчання.

Таблиця 4.1. Порівняння результатів аналізу

Зображення	Правильний результат	ResNet + Adagrad	ResNet + Custom	ResNet + RMSprop	DenseNet + Adagrad	DenseNet + Custom	DenseNet + RMSprop
Baroque 1	1	0.55	0.68	0.63	0.53	0.64	0.61
Baroque 2	1	0.44	0.57	0.52	0.42	0.53	0.5
Chicago School 3	1	0.49	0.62	0.57	0.47	0.58	0.34
Chicago School 4	1	0.51	0.61	0.56	0.49	0.57	0.54
Deconstructivism 5	1	0.49	0.61	0.56	0.47	0.57	0.54
Deconstructivism 6	1	0.49	0.62	0.57	0.47	0.58	0.45
Gothic 7	1	0.46	0.59	0.34	0.46	0.42	0.52
Gothic 8	1	0.5	0.63	0.58	0.48	0.59	0.58
Novelty 9	1	0.54	0.67	0.62	0.52	0.63	0.6
Novelty 10	1	0.52	0.65	0.6	0.5	0.58	0.55
Romanesque 11	1	0.36	0.49	0.44	0.34	0.45	0.42
Romanesque 12	1	0.39	0.52	0.47	0.37	0.48	0.45
Ймовірність правильної класифікації		0.48	0.61	0.54	0.46	0.55	0.51

Висновки по розділу

У цьому розділі було розглянуто порівняння розробленого алгоритму навчання нейронної мережі з існуючими алгоритмами на прикладі задачі класифікації зображень будівель за архітектурним стилем.

Запропонований метод навчання нейронної мережі для класифікації архітектурних стилів будівель за зображеннями будівель за меншу кількість часу навчання досягає більшої точності. В середньому запропонований алгоритм показав в 1.5 рази менше значення функції втрат при завершенні навчання та в 1.13 раз швидше, ніж з використанням інших алгоритмів.

Ймовірність правильної класифікації також є вищою ніж у інших мереж, що означає отримання вищої точності класифікації при навчанні за однаковий час.

5 РОЗРОБКА СТАРТАП-ПРОЕКТУ

5.1 Опис ідеї проекту

5.1.1 Зміст ідеї

У таблиці 5.1 описано зміст ідеї стартап-проекту системи розпізнавання архітектурних стилів будівель за зображеннями, напрямки його застосування, а також переваги, які отримає користувач від використання системи.

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення системи для розпізнавання архітектурних стилів будівель на основі глибинного навчання з розробкою удосконаленого алгоритму-оптимізатору з можливістю збереження результатів.	1. Документування архітектурної спадщини	Організації та комітети, що спеціалізуються на технологіях вимірювання та візуалізації документації та збереження спадщини (такі як CIPA та ISPRS) з використанням розробленої системи зможуть автоматизувати дослідження архітектури.
	2. Збереження та підтримка належного стану архітектурних споруд, загальної архітектури міста	Система може бути використана міськими департаментами містобудування та архітектури при погодження архітектурно-планувальних частин нових проектів, реконструкції існуючих.
	3. Туризм	Використання системи туристами для отримання інформації про архітектурні пам'ятки під час мандрівок

Продовження таблиці 5.1

	4. Класифікація будь-яких зображень	Розроблений оптимізатор може бути використаний для розробки алгоритмів класифікації будь-яких зображень, не лише будівель, що дозволить отримати більшу точність класифікації за менший час навчання
--	---	--

Ідея проекту полягає у розробці системи розпізнавання архітектурних стилів будівель за зображеннями за допомогою методів глибинного навчання. Під час розробки системи був розроблений удосконалений алгоритм-оптимізатор. Система класифікації підтримує 25 різних архітектурних стилів, включаючи античну архітектуру та сучасну. Розроблений алгоритм-оптимізатор дозволяє пришвидшити навчання нових класифікаторів іншими розробниками та може бути використаний для класифікації будь-яких даних.

5.1.2 Аналіз переваг ідеї

Під час дослідження схожих рішень не було виявлено систем з схожою функціональністю, проте існує ряд розроблених підходів до класифікації архітектурних стилів, що можуть бути використані для створення системи, веб застосунку:

- Конкурент 1 – «Houses by Architectural Style Using Convolutional Neural Networks»;
- Конкурент 2 – «Architectural style classification of building facade windows»;
- Конкурент 3 – «Classification of Architectural Heritage Images Using Deep Learning Techniques».

Порівняння характеристик проекту з конкурентами наведено у таблиці 5.2.

Таблиця 5.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/ п	Техніко- економічні характерис- тики ідеї	(потенційні) товари/концепції конкурентів				W (слабк а сторон а)	N (нейтр альна сторон а)	S (сильн а сторон а)
		Мій проект	Конку рент 1	Конку рент 2	Конку рент 3			
1.	Покривають найбільш поширені архітектурні і стилі	+	-	-	+			+
		25	лише 5	лише 4				
		архіте	пошир	архіте				
		ктурни	ених у	ктурни				
		х	США	х				
		стилів		стиля				
2.	Потребують попередньої обробки зображення користуваче м	-	-	+	+			+
3.	Точність результатів	+	+	-	+			+
4.	Безпека даних користувача	+	+	+	+		+	
5.	Швидкість аналізу	+	+	+	+		+	
6.	Універсаль ний метод	+	+	-	-			+

Конкурент 1 розроблявся лише для 5 найбільш поширених архітектурних стилів у США, а конкурент 2 – лише 4 архітектурних стилів. Конкуренти 2 та 3 вміють розпізнавати архітектурні стилі лише за певними елементами на зображенні – вікна, колони, горгульї, дзвіниці – що потребує попереднього аналізу зображення на предмет наявності необхідних елементів, інакше алгоритм працювати не буде.

Із наведеної таблиці можна зробити висновок, що система має значні (сильні) переваги перед існуючими методами.

5.2 Технологічний аудит ідеї проекту

Розглянемо наявність та доступність технологій, що необхідні для реалізації проекту (таблиця 5.3).

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Алгоритм для розпізнавання архітектурних стилів будівель	Бібліотека для створення моделей для машинного навчання	Наявна – tensorflow	Доступна, бібліотека з відкритим вихідним кодом
		API для роботи з згортковими нейронними мережами	Наявна – keras	Доступна
		Бібліотеки для роботи з зображеннями, матрицями	Наявна – pillow та scipy	Доступна
		Фреймворк, для створення веб API для доступу до моделі	Наявна – Flask	Доступна

Продовження таблиці 5.3

№ п/ п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
2.	Збереження результатів в аналізі	База даних	Наявна – MsSQL	Доступна
		O/RM платформа для роботи з базою даних	Наявна – Entity Framework Core	Доступна
		Фреймворк, для створення веб API для доступу	Наявна – ASP .NET Core	Доступна
37.	Інтерфейс користувача	Фреймворк для створення веб застосувань	Наявна – Angular	Доступна

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Як було визначено раніше, конкурентів на ринку немає. Проте, динаміка ринку зростає, що сприяє на попит продукту. На даний момент для виходу на ринок жодних вимог до сертифікації та стандартизації немає. Характеристики потенційного ринку стартап-проекту наведені у таблиці 5.4.

Таблиця 5.4. Характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	0
2.	Динаміка ринку (якісна оцінка)	Зростає
3.	Наявність обмежень для входу (вказати характер обмежень)	Немає
4.	Специфічні вимоги до стандартизації та сертифікації	Немає

Розглянемо потенційних клієнтів стартап-проекту, їх відмінності та вимоги до продукту (таблиця 5.5). Формування потреб потенційних клієнтів допоможе обрати цільову аудиторію та розвивати проект в потрібному напрямку.

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Документування архітектурної спадщини	Компанії, що займаються технологіями	У приватних компаній та міських департаментів	Можливість збереження результатів,
2.	Збереження та підтримка стану архітектурних споруд, архітектури міста	вимірювання та візуалізації документації та збереження спадщини (CIPA та ISPRS); міські департаменти містобудування та архітектури.	різні потреби до збереження даних, різні вимоги до безпеки системи.	точності результатів, проведення аналітики. Наявність веб-застосунку або ж десктопного застосунку.

Продовження таблиці 5.5

3.	Туризм	Туристи, власники мобільних девайсів.	Вимога до доступності та швидкості аналізу.	Наявність мобільного застосування з інформацією про відмінності в архітектурних стилях в різних країнах.
4.	Класифікація інших зображень	Розробники систем класифікації зображень.	Компанії, що розробляють системи класифікації будь-яких зображень.	Доступність алгоритму-оптимізатору для використання іншими розробниками. Наявність документації.

Необхідно провести аналіз ринкового середовища. Розглянемо фактори, що перешкоджають та сприяють впровадженню проекту, наведені у таблиці 5.6 та таблиці 5.7 відповідно.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Невідповідність вимогам ринку	Відсутність на ринку аналогів не дає можливості проаналізувати вимоги та типові приклади використання системи. Це може привести до відсутності критичної функціональності, що необхідна користувачу.	Вихід на ринок з мінімальними функціональними можливостями, пошук потенційних користувачів системи та збір вимог до системи завдяки спілкуванню з потенційними користувачами.
2.	Відсутність інфраструктури	Для аналізу зображень та збереження результатів класифікації необхідна достатня обчислювальна потужність та сервера для збереження даних.	Використання хмарних технологій, що дозволяють швидко розширюватись та нарощувати потужність.
3.	Відсутність необхідних архітектурних стилів	Архітектурні стилі дуже відрізняються в залежності від країни, навчена мережа може бути не готова для аналізу таких даних	Можливість постійного донавчання мережі на даних, що має користувач системи.

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Відсутність конкурентів	Відсутність на ринку аналогів розширює цільову аудиторію та рівень зацікавленості до удосконалення продукту користувачами.	Вихід на ринок з мінімальними функціональними можливостями та подальше удосконалення продукту в залежності від вимог клієнту.
2.	Доступність	Використання хмарних технологій дозволяють за мінімальною ціною розмістити продукт в мережі Інтернет.	Для запуску проекту не треба велика кількість коштів на інфраструктуру, що дозволить швидше знайти користувачів системи.

Проведемо аналіз пропозиції та визначимо загальні риси конкуренції на ринку (таблиця 5.8).

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції – чиста	На ринку відсутні конкуренти	Відсутність можливості аналізу необхідної функціональності системи до виходу на ринок. Активний розвиток продукту після виходу на ринок

Продовження таблиці 5.8

2. Рівень конкурентної боротьби – міжнародний	Проект призначений для використання на міжнародному ринку	Необхідність урахування потреб міжнародного ринку, необхідність підтримки глобалізації, перекладу різними мовами
3. За галузевою ознакою – міжгалузева	Проект орієнтований на використання в різних галузях – туризм, міські департаменти містобудування та компанії, що займаються документацією та збереженням спадщини	Необхідний аналіз усіх галузей та вивчення їх особливостей
4. Конкуренція за видами товарів – товарно-видова	З появою аналогів на ринку, з'явиться конкуренція між аналогічними продуктами	Необхідність постійно удосконалювати продукт, розширюючи його можливості, залишаючи продукт доступним
5. За характером конкурентних переваг – нецінова	Конкуренція за рахунок високої якості та ширших можливостей	Покращення продукту та розширення функціональних можливостей
6. За інтенсивністю – не марочна	Продукт є новим на ринку, тому марка не відіграє роль при запуску	При виході на ринок необхідно орієнтуватися на нецінові переваги продукту, а не на торгівельну марку

Стартап-проект системи для розпізнавання архітектурних стилів будівель має гарні можливості: відсутність конкурентів та широка цільова аудиторія дає

велику кількість можливостей для швидкого виходу на ринок. Проводити більш детальний аналіз умов конкурентності в галузі за М. Портером на початковому етапі на має сенсу через відсутність конкурентів. Проте необхідно розглянути фактори конкурентоспроможності стартап-проекту, що дозволять продукту бути конкурентоспроможним при появі конкурентів на ринку (таблиця 5.9).

Таблиця 5.9. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Великий перелік архітектурних стилів	Система класифікації підтримує 25 різних архітектурних стилів, включаючи античну архітектуру та сучасну. Цей перелік є достатньо широким для більшості потреб ринку.
2.	Точність результатів	Алгоритм був розроблений з використанням широкої вибірки зображень, що дозволяє отримати точні результати класифікації
3.	Адаптивність	Можливість донавчання\перенавчання алгоритму дозволяє коригувати можливості системи залежно від потреб користувача
4.	Швидкість аналізу та точність	При використанні алгоритму для туристу швидкість аналізу може бути більш важливою характеристикою ніж точність. Можливість вибору цих параметрів робить алгоритм універсальним для використання в різних галузях.
5.	Можливість збереження результатів	При документування спадщини збереження результатів аналізу є важливим фактором вибору системи.

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities). Матриця наведена у таблиці 5.10.

Таблиця 5.10. SWOT- аналіз стартап-проекту

Сильні сторони: 1. Підтримка великої кількості архітектурних стилів 2. Точність результатів 3. Можливість збереження результатів 4. Можливість обирати швидкість аналізу 5. Адаптивність	Слабкі сторони: 1. Великі вимоги до пам'яті для збереження результатів 2. Невідповідність вимогам ринку 3. Відсутність інфраструктури 4. Відсутність необхідних архітектурних стилів певної території
Можливості: 1. Відсутність конкурентів 2. Доступність 3. Можливість використання в різних галузях	Загрози: 1. Продукт залишиться непоміченим після виходу на ринок. 2. Компанії не зможуть швидко перейти на використання системи – відсутність користувачів довгий час 3. Відсутність вимог до системи і як наслідок, незадоволення потреб користувачів

Отже, стартап-проект має слабкі сторони та загрози, проте його сильні сторони та можливості переважають недоліки. Розглянемо альтернативи ринкової поведінки для введення проекту на ринок та орієнтований оптимальний час їх реалізації для зменшення ризиків (таблиця 5.11).

Таблиця 5.11. Альтернативи ринкового впровадження стартап-проекту

№ № п/ п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Розробка мобільного застосунку або мобільної версії веб-застосунку	Висока ймовірність отримання ресурсів	3 місяців
2.	Повна розробка системи до виходу на ринок	При наявності функціональних вимог для перших користувачів системи, ймовірність отримання ресурсів середня	5 місяців

5.4 Розроблення ринкової стратегії проекту

При розробці ринкової стратегії, спочатку необхідно визначити стратегію охоплення ринку – описати цільові групи потенційних споживачів (таблиця 5.12).

Таблиця 5.12. Вибір цільових груп потенційних споживачів

№ п/ п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів в сприйнят и продукт	Орієнтовн ий попит в межах цільової групи (сегменту)	Інтенси вність конкуре нції в сегменті	Просто та входу у сегмент
1.	Компанії, що займаються технологіями вимірювання та візуалізації документації та збереження спадщини (CIPA та ISPRS); міські департаменти містобудування та архітектури.	Середня готовність	Високий попит	Невисока конкуренція	Складно
2.	Туристи, власники мобільних девайсів.	Висока готовність	Високий попит	Висока конкуренція	Складно
3.	Розробники систем класифікації зображень.	Висока готовність	Середній попит	Висока конкуренція	Складно

За результатами аналізу потенційних груп споживачів, було обрано цільові групи для яких пропонуватиметься товар – компанії, що займаються технологіями вимірювання та візуалізації документації, туристи та розробники систем класифікації зображень.

Пропонується працювати зі всім ринком и обрати стандартизовану програму – масовий маркетинг. Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 5.13).

Таблиця 5.13. Визначення базової стратегії розвитку

№ п/ п	Обрана альтерна тива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспромо жні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1.	Розвиток в усіх доступних сегментах ринку	Охопити усі доступні сегменти	Конкуренції на ринку немає.	Стратегія диференціації

Далі необхідно визначити стратегію конкурентної поведінки (таблиця 5.14).

Таблиця 5.14. Визначення базової стратегії конкурентної поведінки

№ п/ п	Чи є проект «пер- шопрохідце м» на ринку?	Чи буде компанія шукати нових спо живачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конку- рентної поведінки*
1.	Так	Компанія буде шукати нових спо живачів	Ні, не буде	Стратегія заняття конкурентної ніші

На основі вимог споживачів обраних сегментів до стартап-проекту та до продукту, в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки, розробляється стратегія позиціонування (таблиця 5.15).

що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 5.15. Визначення стратегії позиціонування

№ п / п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Точність результатів, можливість збереження результатів	Стратегія диференціації	Висока точність результатів, можливість збереження та перегляду результатів, аналітика	Доступність, відсутність аналогів, якість продукту

Отже, цільовою групою споживачів є компанії, що займаються технологіями вимірювання та візуалізації документації, туристи та розробники систем класифікації зображень, тобто весь ринок, тому було обрано стратегію диференціації як базову стратегію розвитку. Стартап проект буде шукати нових користувачів, тому за стратегію конкурентної поведінки було обрано стратегію заняття конкурентної ніші.

5.5 Розроблення маркетингової програми стартап-проекту

Спочатку необхідно сформувати маркетингову концепції товару, який отримає споживач. Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару (таблиця 5.16).

Таблиця 5.16. Визначення переваг концепції потенційного товару

№ п / п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Документування архітектурної спадщини	Можливість збереження результатів. Наявність веб-застосунку або ж десктопного застосунку.	Точність результатів, підтримка 25 різних архітектурних стилів, включаючи античну та сучасну. Цей перелік є широким для більшості потреб ринку.
2.	Збереження та підтримка стану архітектурних споруд, архітектури міста	Точності результатів, проведення аналітики	Алгоритм був розроблений з використанням широкої вибірки зображень, що дозволяє отримати точні результати класифікації.
3.	Туризм	Наявність мобільного застосунку з інформацією про відмінності архітектурні стилі в різних країнах.	Наявність мобільного застосунку.

Продовження таблиці 5.16

4.	Класифікація інших зображень	Доступність алгоритму-оптимізатору для використання іншими розробниками. Наявність документації.	Доступність.
----	------------------------------	---	--------------

Трирівнева маркетингова модель товару наведена у таблиці 5.17.

Таблиця 5.17. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Веб застосування для визначення архітектурного стилю будівлі		
II. Товар у реаль- ному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Якість	Нм	Тх Тл
	2. Доступність	Нм	Тх Тл
	Якість: тестування програмного продукту, тестування вимог до безпеки		
	Марка: Система розпізнавання архітектурних стилів		
III. Товар із підкріпленням	До продажу		
	Після продажу		
За рахунок чого потенційний товар буде захищено від копіювання: Отримання патентів на корисні моделі та винаходи.			

Наступним кроком визначимо цінові межі на потенційний товар (таблиця 5.18), формування системи збуту (таблиця 5.19) та концепції маркетингових комунікацій (таблиця 5.20).

Таблиця 5.18. Визначення меж встановлення ціни

№ п/ п	Рівень цін на товари- замінники	Рівень цін на товари- аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	Інформація не доступна через відсутність товарів-замінників.	Інформація не доступна через відсутність товарів-замінників.	10 000 грн/міс	500 - 12 000 грн/міс

Таблиця 5.19. Формування системи збуту

№ п/ п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальн а система збуту
1.	Оплата за використання пам'яті для збереження результатів аналізу. Оплата за надання послуг підтримки продукту, внесення змін	Реклама, надання послуг підтримки продукту.	Низька	Проводити збут власними силами

Таблиця 5.20. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Приватні компанії та міські департаменти не знайдуть продукт через рекламу.	Інтернет, електронна пошта	Доступність, точність, зручність	Звернути увагу на продукт та запевнити потенційного користувача у його надійності	Найкраще співвідношення ціна/якість
2.	Туристи мають вимогу до доступності та швидкості аналізу.	Інтернет	Доступність, зручність використання	Звернути увагу на продукт	Швидко та пізнавально під час подорожей, доступно у смартфоні

Висновки по розділу

У даному розділі було описано зміст ідеї стартап-проекту розпізнавання архітектурних стилів будівель, напрямки його застосування, проведено аналіз ринкових можливостей запуску стартап-проекту. Після аналізу конкурентів можна зробити висновок, що система має значні переваги перед існуючими методами. Стартап-проект може бути легко реалізований, тому що усі технології необхідні для його розробки існують та є доступними.

Під час аналізу ринкових можливостей запуску стартап-проекту було виявлено, що серед потенційних цільових груп клієнтів є високий попит на продукт, динаміка ринку зростає, що також сприяє на попит продукту.

На даний момент для виходу на ринок жодних вимог до сертифікації та стандартизації немає, що спрощує вихід продукту на ринок.

На ринку відсутні аналоги продукту, що є плюсом, проте це не дає можливості проаналізувати вимоги та типові приклади використання системи, що може вплинути на стратегії розробки продукту та виходу на ринок.

При аналізі конкуренції було виявлено, що рівень конкурентної боротьби – міжнародний, що створює ризик –архітектурні стилі дуже відрізняються в залежності від країни, система може бути не готова для аналізу таких даних.

Проте продукт має можливість постійного донавчання мережі на даних, що має користувач системи.

При реалізації проекту доцільно обрати розробку мобільного застосунку або мобільної версії веб-застосунку як альтернативу впровадження. Стартап-проект має слабкі сторони та загрози, проте його сильні сторони та можливості перевищують недоліки. Тому подальша імплементація проекту є доцільною.

ЗАГАЛЬНІ ВИСНОВКИ

Отже, у роботі було розглянуто використання нейронних мереж для розв'язання задачі класифікації зображень, відрізняється від інших задач класифікації наявністю міжкласових взаємозв'язків між різними стилями. та способи підвищення точності результатів та зменшення часу, необхідного для навчання алгоритму. Актуальність розробки алгоритму полягає в тому, що він дозволить виконувати класифікацію зображень з тісними міжкласовими зв'язками не тільки для класифікації архітектурного стилю будівель, а також і зображень в інших галузях. Наукова новизна полягає в удосконаленні алгоритму коригування параметрів нейронної мережі під час її навчання для розв'язання задачі класифікації зображень з тісними міжкласовими взаємозв'язками. Практичне значення одержаних результатів полягає у створенні системи розпізнавання архітектурних стилів будівель та методу навчання нейронної мережі, що за меншу кількість часу навчання досягає більшої точності.

Під час виконання роботи було проведено аналіз та порівняння існуючих алгоритмів для розв'язання задачі розпізнавання архітектурних стилів будівель та описано їх особливості, виявлено, що на даний момент систем зі схожим функціоналом не існує. Було виявлено, що методи, засновані на глибокому навчанні, значно перевершують досягнуті результати за всіма іншими представленими методами.

У розділі розробка алгоритму класифікації зображень описано математичну постановку задачі, цільову функцію та наведено схеми алгоритмів навчання мережі та класифікації зображення з її використанням. Розглянуто загальні поняття та підходи, використані при розробці алгоритму класифікації, процес навчання нейронної мережі та коригування параметрів. У розділі описано методи та процеси виділення кордонів та меж предметів на зображенні, підходи до зменшення зображення. Також, було обґрунтовано правильність роботи алгоритму навчання нейронної мережі.

У розділі опис програмного та технічного забезпечення було розглянуто технологічні аспекти інформаційної системи класифікації зображень будівель за

архітектурним стилем. Було описано технології, обрані для створення та розгортання інформаційної, обґрунтовано їх вибір та наведені переваги їх використання – Python Flask Microframework, .NET Core Web API, СУБД Ms SQL та фреймворк для створення SPA – Angular. Також описані вимоги до технічного забезпечення системи, функціональні вимоги та вимоги до надійності. За допомогою діаграми послідовності показано основні шляхи використання системи користувачем – авторизація користувача, аналіз зображення та його зображення. Розглянуто керівництво користувача, надані інструкції по розгортанню системи, описано архітектуру системи у вигляді діаграм класів, компонентів та розгортання з їх детальним описом.

У розділі аналізу отриманих результатів було порівняно удосконалений алгоритм навчання нейронної мережі з існуючими алгоритмами на прикладі задачі класифікації зображень будівель за архітектурним стилем. Запропонований метод навчання нейронної мережі для класифікації архітектурних стилів будівель за зображеннями будівель за меншу кількість часу навчання досягає більшої точності. В середньому запропонований алгоритм показав в 1.5 рази менше значення функції втрат при завершенні навчання та в 1.13 раз швидше, ніж з використанням інших алгоритмів. Ймовірність правильної класифікації також є вищою ніж у інших мереж, що означає отримання вищої точності класифікації при навчанні за однаковий час.

У розділі розробки стартап-проекту описано зміст ідеї стартап-проекту, напрямки його застосування, проведено аналіз ринкових можливостей запуску стартап-проекту. Після аналізу конкурентів можна зробити висновок, що система має значні переваги перед існуючими методами. Стартап-проект може бути легко реалізований, тому що усі технології необхідні для його розробки існують та є доступними. Стартап-проект має слабкі сторони та загрози, проте його сильні сторони та можливості перевищують недоліки. Тому подальша імплементація проекту є доцільною.

ПЕРЕЛІК ПОСИЛАНЬ

1. Berg, A.C. Parsing images of architectural scenes / Berg, A.C, Grabler, F, Malik, J // IEEE 11 th International Conference on Computer Vision, / Berg, A.C, Grabler, F, Malik, J., 2007. – С. 2–8.
2. Wei-Ta Chu. ICMR '12: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval. / Wei-Ta Chu, Ming-Hung Tsai // 27. – 2012. – URL: <https://doi.org/10.1145/2324796.2324831>.
3. What makes paris look like paris? / Doersch C, Singh S, Gupta A, Sivic J. // ACM Transactions on Graphics (TOG). – 2012. – №31.
4. Goel A. Are buildings only instances?: exploration in architectural style categories / Goel A, Juneja M, Jawahar C, 2012. – (Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing). – С. 1–12.
5. Object retrieval with large vocabularies and fast spatial matching / Philbin J, Chum O, Isard M, Sivic J // Computer Vision and Pattern Recognition / Philbin J, Chum O, Isard M, Sivic J., 2007. – С. 1–8.
6. Architectural Style Classification using Multinomial Latent Logistic Regression / Z. Xu, D. Tao, Y. Zhang, J. Wu. // European Conference on Computer Vision (ECCV2014),. – 2014.
7. AUTOMATIC ARCHITECTURAL STYLE RECOGNITION / M. Mathias, A. Martinovic, J. Weissenberg, S. Haegler. // PSI/VISICS, Department of Electrical Engineering. – 2011. – С. 1–6.
8. Shalunts G. Architectural Style Classification of Building Facade Windows / G. Shalunts, Y. Haxhimusa, R. Sablatnig. // Vienna University of Technology, Institute of Computer Aided Automation Computer Vision Lab, Institute of Computer Graphics and Algorithms, Pattern Recognition and Image Processing Lab. – 2011.
9. Classification of Architectural Heritage Images Using Deep Learning Techniques / J.Llamas, P. Lerones, R. Medina, E. Zalama. // Applied Sciences. – 2017. – №992. – С. 1–25.

10. 3D Façade Labeling over Complex Scenarios: A Case Study Using Convolutional Neural Network and Structure-From-Motion / Rodolfo Georjute Lotte, Norbert Haala, Mateusz Karpina, Luiz Eduardo Oliveira e Cruz de Aragão. // *Remote Sensing*. – 2018. – №1435. – С. 1–28.
11. Новіченко Н. В. Порівняння алгоритмів класифікації зображень для вирішення задачі визначення архітектурного стилю будівель / Неля Валеріївна Новіченко. // *Матеріали V всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ICTY-2020)* – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського» – 2020. – С. 119–123.
12. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* 2015, 521, 436–444. URL: <https://pubmed.ncbi.nlm.nih.gov/26017442/>.
13. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2012; pp. 1097–1105.
14. Simonyan K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman. // *Computer Vision and Pattern Recognition*. – 2015. – №6.
15. Dive into Deep Learning / Zhang A., Lipton Z. C., Mu Li, Smola A. J.. – 2020. – №15. – С. 263–296.
16. Densely Connected Convolutional Networks / Gao Huang, Zhuang Liu, Kilian Q. Weinberger, Laurens van der Maaten. // *arXiv*. – 2016. – С. 1–12.
17. Going deeper with convolutions / Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet. // *arXiv*. – 2014. – С. 1–12.
18. Chollet F. Xception: Deep Learning With Depthwise Separable Convolutions / Francois Chollet. // *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. – 2017. – №1. – С. 1251–1258.
19. Khandelwal R. Overview of different Optimizers for neural networks [Електронний ресурс] / Renu Khandelwal // *DataDrivenInvestor*. – 2019. – URL:

<https://medium.datadriveninvestor.com/overview-of-different-optimizers-for-neural-networks-e0ed119440c3>.

20. Sorokina K. Image Classification with Convolutional Neural Networks / Ksenia Sorokina // A Medium Corporation. – 201. – URL: <https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>.
21. Новіченко Н. В. Застосування згорткових нейронних мереж при класифікації зображень будівель за архітектурним стилем / Неля Валеріївна Новіченко. // Матеріали IV всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – 2020. – С. 143–146.
22. Rossana M. S. Cruz. Artificial Neural Networks and Efficient Optimization Techniques for Applications in Engineering / Rossana M. S. Cruz, Helton Maia Peixoto, Rafael Magalhães // Artificial Neural Networks - Methodological Advances and Biomedical Applications / Rossana M. S. Cruz, Helton Maia Peixoto, Rafael Magalhães. – London: IntechOpen, 2011. – С. 1–26.
23. Новіченко Н. В. Розпізнавання архітектурних стилів будівель за допомогою методів машинного навчання / Н. В. Новіченко, О. В. Гавриленко. // Матеріали IV всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – 2021. – С. 120–125.
24. Chen Wang. Convolutional Neural Network for Image Classification [Електронний ресурс] / Chen Wang, Yang Xi. – 2015. URL: <http://www.cs.jhu.edu/~cwang107/files/cnn.pdf>.
25. Python 3.7.10 documentation – 2021. URL: <https://docs.python.org/3.7/>.
26. Новіченко Н. Architecture Prediction Web API [Електронний ресурс] / Неля Новіченко. URL: <https://github.com/NeliaNovichenko/architecture-prediction-api/blob/master/README.md>.

- 27.Новіченко Н. Architecture Recognition Client [Електронний ресурс] / Неля Новіченко. URL: <https://github.com/NeliaNovichenko/architecture-recognition-client/blob/master/README.md>.
- 28.Новіченко Н. Architecture Recognition Web API [Електронний ресурс] / Неля Новіченко. URL: <https://github.com/NeliaNovichenko/architecture-recognition-api/blob/master/README.md>.
29. Wittig A. Amazon Web Services in Action / A. Wittig, M. Wittig. – Shelter Island, NY, USA: Manning Publications Co, 2019. – 498 с.
- 30.Гавриленко О. В. Дослідження оптимізаторів нейронної мережі для класифікації зображень будівель за архітектурним стилем / О. В. Гавриленко, Н. В. Новіченко. // Системні технології. – 2021. – №5.

ДОДАТОК А Графічний матеріал

Схема структурна діаграми класів

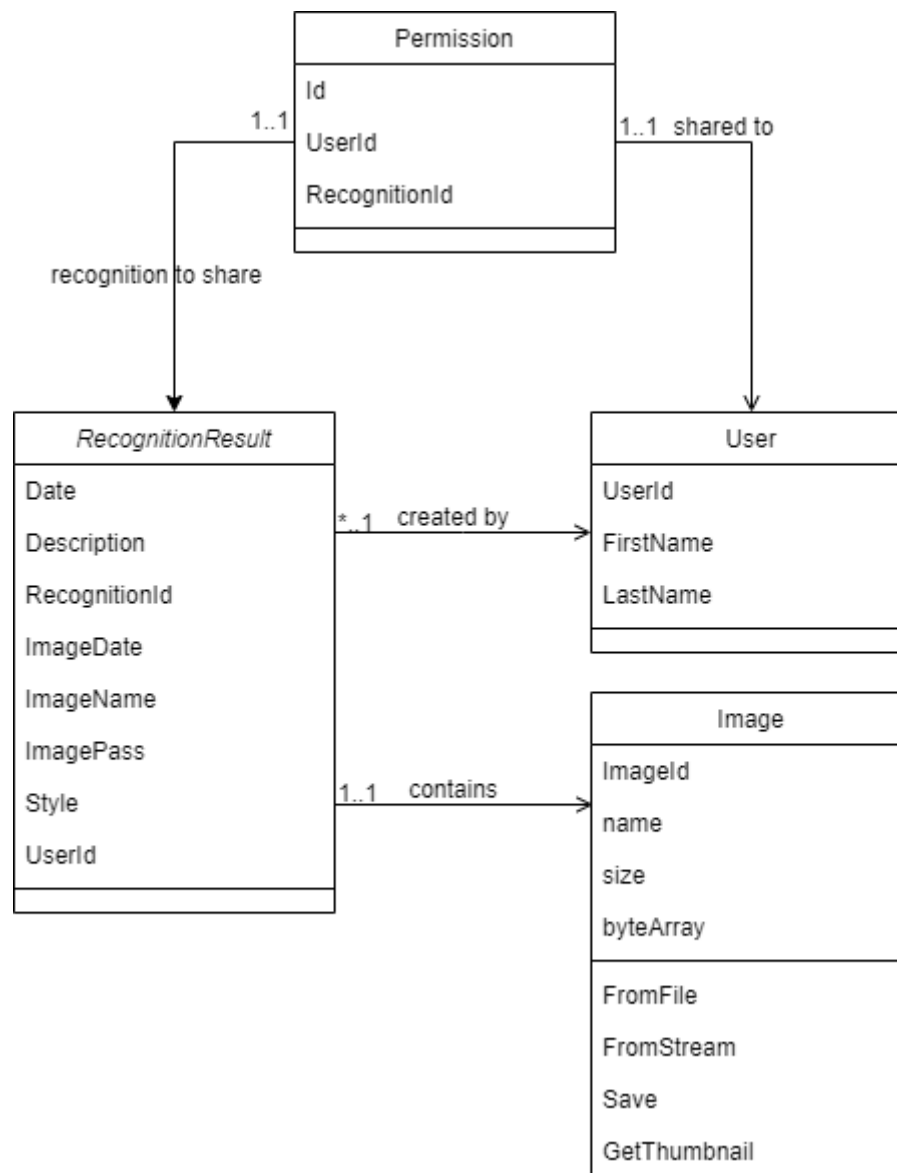


Схема структурна послідовності аутентифікації користувача

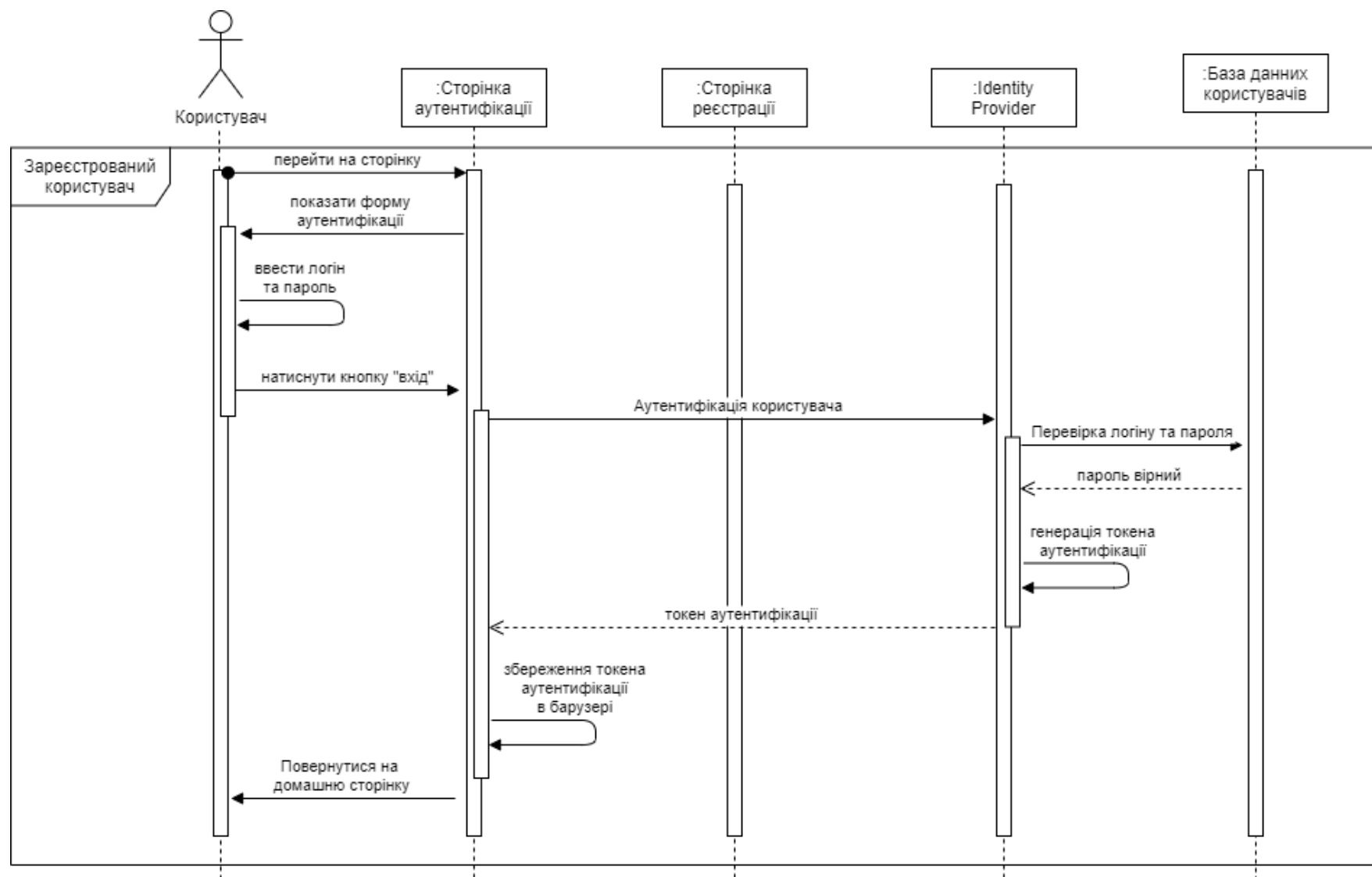


Схема структурна послідовності реєстрації користувача

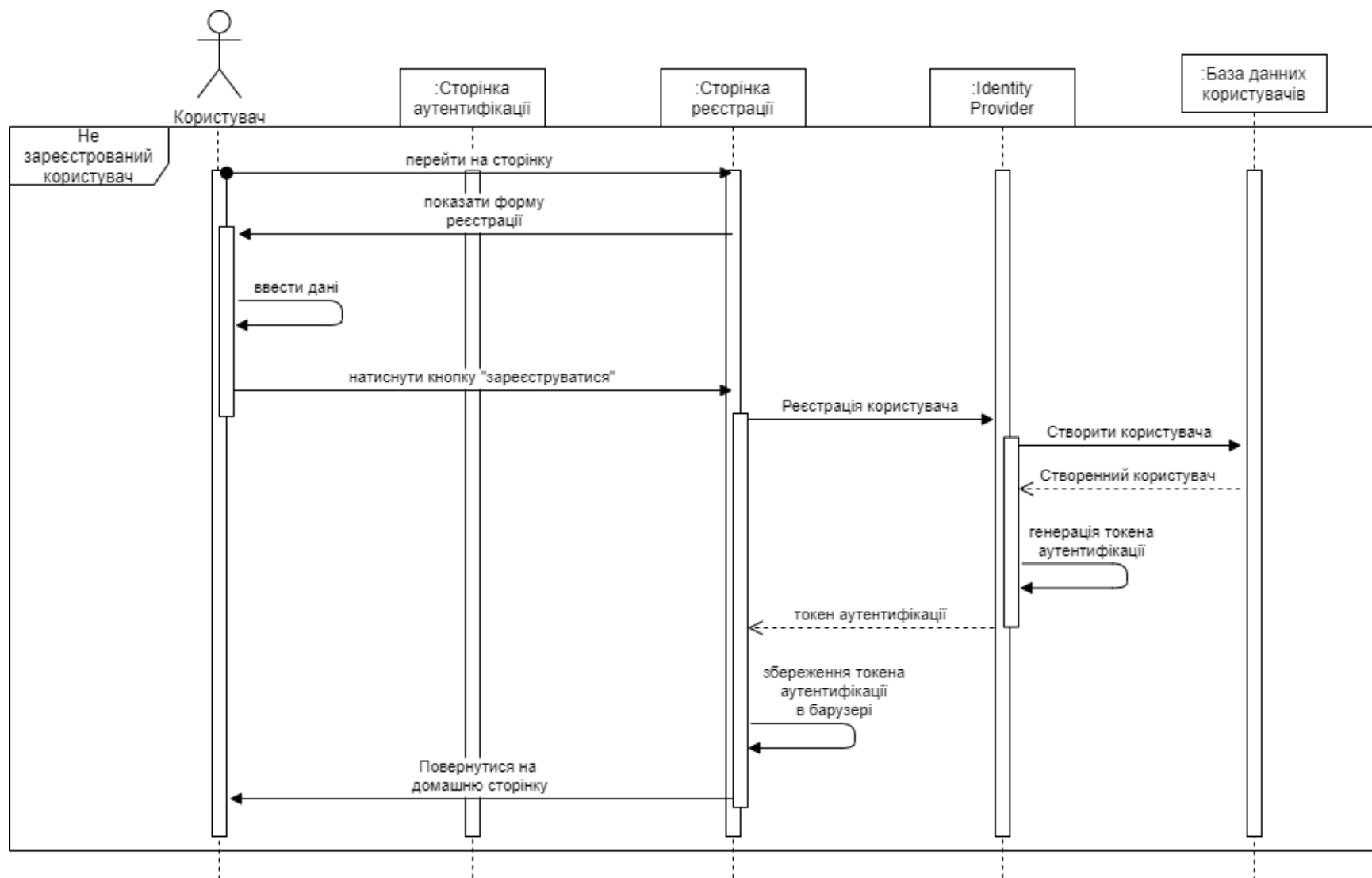


Схема структурна послідовності аналізу зображення

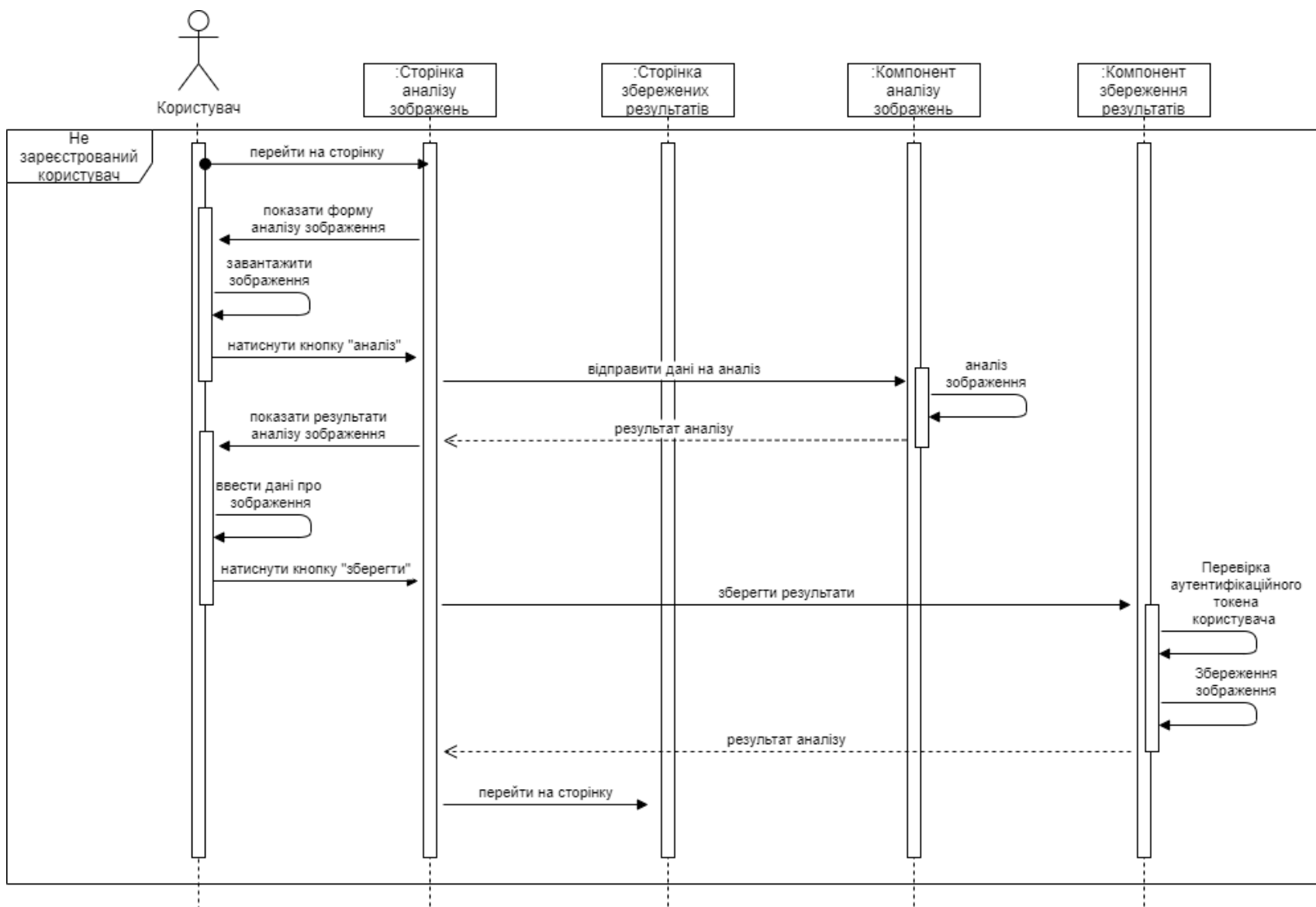


Схема структурна розгортання

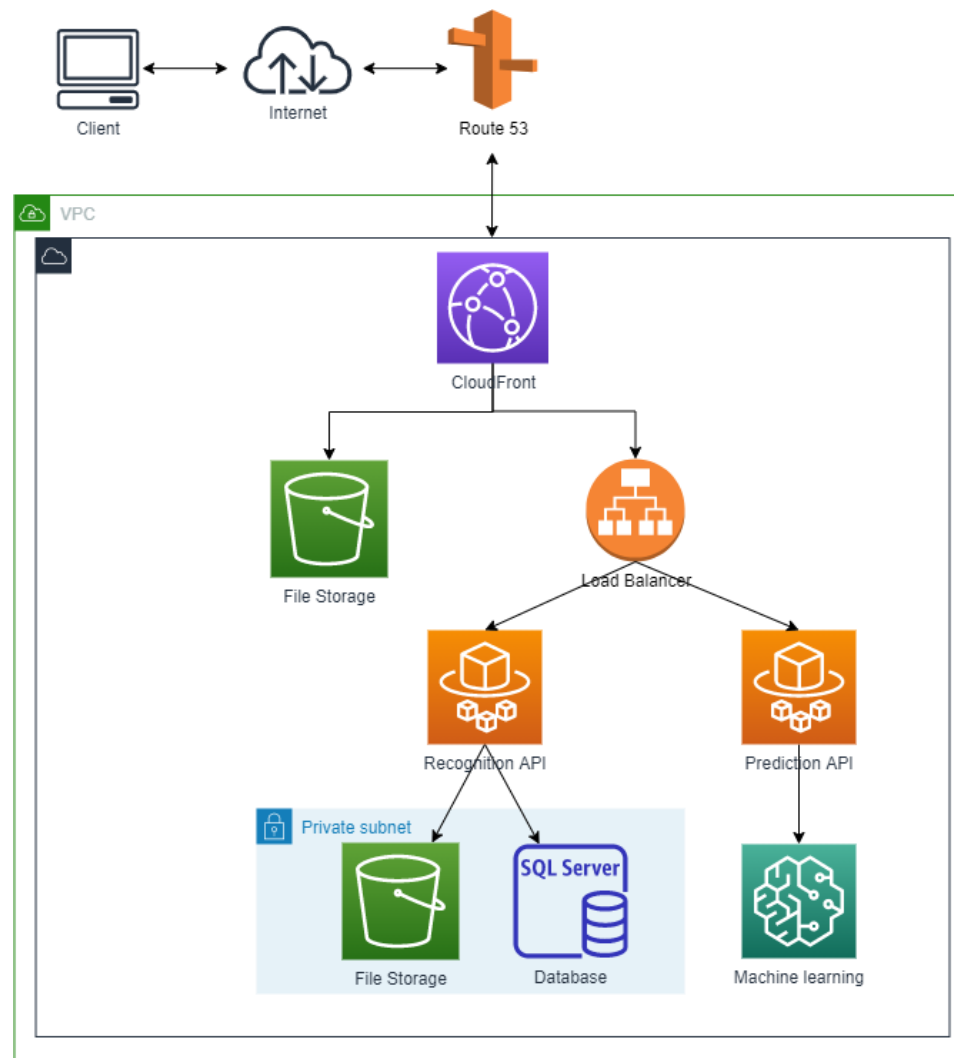


Схема структурна компонентів

