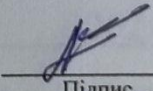
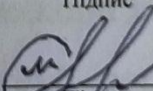
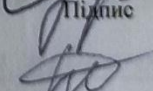
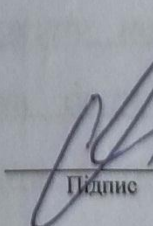


КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Метод нейромережевої класифікації зображень документів що посвідчують особу

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

Виконав: студент 4 курсу, група КН-19-2  А.В. Похитун
Курс, група виконавця Підпис Ініціали, прізвище
Керівник: к.т.н., доцент кафедри КН  О.В. Мазурець
Науковий ступінь, посада Підпис Ініціали, прізвище
Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КН, д.т.н., професор  О.В. Бармак
Підпис Ініціали, прізвище

05 06 2023 р.

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

Освітня програма освітньо-професійна програма підготовки бакалавра

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

« 06 » 03 2023 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

1. Тема кваліфікаційної роботи бакалавра: «Метод нейромережевої класифікації зображень документів що посвідчують особу»

2. Завдання видано студенту Похитуну Андрію Вікторовичу

(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КН Мазурець Олександр Вікторович

(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 01 » 03 2023 р. № 5

5. Дата видачі завдання студенту: « 03 » 03 2023 р.

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

Провести аналіз предметної області класифікації зображень документів, визначити особливості застосування згорткових нейронних мереж для задач класифікації при роботі з зображеннями документів. Виконати аналіз існуючих рішень щодо подібних задач. Розробити метод нейромережевої класифікації зображень документів. Спроектувати архітектуру нейронної мережі для класифікації зображень документів. Розробити структуру інформаційної системи нейромережевої класифікації зображень документів й структуру відповідної бази даних. Спроектувати та створити застосунок за розробленою технологією, виконати його тестування.

7. Календарний план виконання кваліфікаційної роботи бакалавра:

№	Назва етапів (розділів) кваліфікаційної роботи бакалавра	Термін виконання	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи бакалавра з керівником	грудень 2022	виконано
2	Ознайомлення з предметною областю, формулювання мети та задач дослідження, визначення об'єкта та предмета дослідження	січень 2023	виконано
3	Робота над розділом 1 – Характеристика предметної області та постановка задачі	січень 2023	виконано
4	Робота над розділом 2 – Технологія нейромережевої класифікації зображень документів що посвідчують особу	березень 2023	виконано
5	Робота над розділом 3 – Програмна реалізація інформаційної системи класифікації зображень документів	квітень 2023	виконано
6	Оформлення пояснювальної записки згідно вимог	травень 2023	виконано
7	Попередній захист кваліфікаційної роботи бакалавра	травень 2023	виконано
8	Захист кваліфікаційної роботи бакалавра на засіданні Екзаменаційної комісії	червень 2023	виконано

Виконавець: студент 4 курсу, група КН-19-2

Курс, група виконавця


Підпис

А.В. Похитун
Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КН

Науковий ступінь, посада


Підпис

О.В. Мазурець
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Метод нейромережевої класифікації зображень документів що посвідчують особу

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-19-2 Похитун Андрій Вікторович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Мазурець Олександр Вікторович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
68	40	10	26	6

Метою кваліфікаційної роботи бакалавра є розробка та програмна реалізація технології нейромережевої класифікації зображень документів що посвідчують особу, яка дозволяє за зображеннями документа визначати його вид за допомогою згорткової нейромережі.

Результатом виконання кваліфікаційної роботи бакалавра є створений метод нейромережевої класифікації зображень документів що посвідчують особу та інформаційна система класифікації зображень документів, яка використовує розроблену технологію нейромережевої класифікації зображень документів що посвідчують особу.

Ключові слова: класифікація зображень документів, згорткова нейронна мережа, зображення документа, нейромережа, система класифікації зображень, документи.

Виконавець:

студент 4 курсу, група КН-19-2

Курс, група виконавця


Підпис

А.В. Похитун

Ініціали, прізвище

Зміст

Перелік скорочень	4
Вступ.....	5
Розділ 1 Характеристика предметної області та постановка задачі	7
1.1 Аналіз предметної області класифікації зображень документів	7
1.2 Особливості застосування згорткових нейронних мереж для задач класифікації	12
1.3 Аналіз існуючих рішень для подібних задач	14
1.4 Мета, задачі та вимоги до реалізації програмної системи.....	18
Розділ 2 Метод нейромережевої класифікації зображень документів що посвідчують особу.....	19
2.1 Опис і кроки методу нейромережевої класифікації зображень документів	19
2.2 Особливості використання методу нейромережевої класифікації зображень документів.....	21
2.2.1 Навчання нейронної мережі навчальною множиною зображень документів.....	21
2.2.2 Самоперевірки нейронної мережі тестовою множиною зображень документів.....	23
2.2.3 Нейромережева класифікація робочого зображення документу	24
2.3 Розробка архітектури нейронної мережі для класифікації зображень документів що посвідчують особу	26
2.3.1 Використання бібліотеки TensorFlow для класифікації зображень документів.....	26
2.3.2 Налаштування гіперпараметрів моделі TensorFlow	28
2.3.3 Операції згорткової нейронної мережі TensorFlow для класифікування зображень.....	30
2.3.4 Архітектура нейронної мережі	34
2.4 Підготовка зразків робочих вхідних даних для класифікатора	36

2.5	Проектування структури інформаційної системи класифікації зображень документів.....	41
2.6	Даталогічна модель бази даних інформаційної системи класифікації зображень документів.....	43
2.7	Висновки до розділу 2	46
Розділ 3	Програмна реалізація інформаційної системи класифікації зображень документів.....	48
3.1	Вибір засобів розробки інформаційної системи класифікації зображень документів.....	48
3.2	Структура модулів інформаційної системи класифікації зображень документів та їх взаємозв'язок	49
3.3	Особливості реалізації інформаційної системи	50
3.4	Опис функціональних можливостей інформаційної системи класифікації зображень документів.....	54
3.4.1	Інструкція користувача	54
3.4.2	Тестування інформаційної системи класифікації зображень документів.....	58
3.5	Дослідження ефективності застосування методу нейромережевої класифікації зображень документів що посвідчують особу.....	61
3.6	Висновки до розділу 3	63
Висновки	65
Перелік посилань.....		67
Додатки		

Перелік скорочень

Скорочення, термін, позначення	Пояснення
НР	Нейронна мережа
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
CNN	Convolutional neural network
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interface
PHP	Hypertext Preprocessor
MVC	Model-View-Controller

Вступ

Кваліфікаційна робота бакалавра присвячена розробці методу нейромережевої класифікації зображень документів що посвідчують особу та відповідної інформаційної системи, яка дозволяє за зображеннями документа визначати його вид за допомогою згорткової нейромережі.

Актуальність. У ХХІ столітті документи є невід'ємною частиною кожної сучасної людини. Особливо велика кількість часу витрачається на роботу з документами у різних державних установах або банках. Для того щоб провести огляд документів та отримання звідти потрібної інформації людині може знадобитись деякий час. Якщо взяти до уваги те що документів в день вони розглядають досить велику кількість було б ефективним рішенням автоматизувати цей процес. Отже на даний момент цей напрямок досить актуальний, адже може дозволити заощадити досить велику кількість часу.

Для ідентифікації особи, автоматизованої авторизації та автоматизованого витягу персональних даних ефективним шляхом є розпізнавання візуальних відомостей, що містяться на офіційних документах, які посвідчують особу (фото, прізвище, ім'я та по батькові особи, дата народження, номери документів тощо). Методи, що використовуються для розпізнавання відомостей з документів, відрізняються у залежності від типу документу. Тому процесу аналізу документу передуює важливий крок – класифікація зображень цих документів. Подальша ефективність автоматизованої обробки зображення документу безпосередньо залежить від коректності визначення його типу. Оскільки обробка зображення документу є автоматизованим процесом, то і процес класифікації зображень документів теж має виконуватись автоматизовано. Це визначає мету даної роботи.

Об'єкт дослідження – процес класифікації зображень документів що посвідчують особу для систем ідентифікації.

Предмет дослідження – моделі, методи, алгоритми та засоби для класифікації зображень документів що посвідчують особу.

Мета кваліфікаційної роботи бакалавра – розробка та програмна реалізація методу нейромережевої класифікації зображень документів що посвідчують особу.

Завдання кваліфікаційної роботи бакалавра – Провести аналіз предметної області класифікації зображень документів, визначити особливості застосування згорткових нейронних мереж для задач класифікації при роботі з зображеннями документів. Виконати аналіз існуючих рішень щодо подібних задач. Розробити метод нейромережевої класифікації зображень документів. Розробити архітектуру нейронної мережі для класифікації зображень документів. Спроекувати структуру інформаційної системи нейромережевої класифікації зображень документів й структуру відповідної бази даних. Спроекувати та створити застосунок за розробленою технологією, виконати його тестування.

Розділ 1 Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області класифікації зображень документів

Документ що посвідчує особу – офіційний документ, який використовується для ідентифікації та підтвердження особистості конкретної людини.

Класифікація зображень документів є важливою задачею в галузі обробки даних. Ця задача широко розповсюджена в різних галузях, таких як фінанси, наука, технології, медицина.

Для того, щоб більш детально зрозуміти предметну область, потрібно проаналізувати наступні аспекти:

- типи зображень документів;
- методи обробки зображень;
- застосування класифікації зображень;
- типи класифікації;
- якість класифікації.

Для того щоб класифікувати зображення можна використовувати різноманітні методи обробки. До найпопулярніших входить сегментація, машинне навчання, розпізнавання образів [1].

Класифікація зображень документів може використовуватись в різних сферах. В кожній із цих галузей будуть свої вимоги до класифікації, тому необхідно розуміти контекст конкретної задачі.

Класифікація зображень може бути поділена на два типи бінарний (поділ на два класи) та багато класова класифікація (включає поділ більше ніж на два класи) [2]. Кожен тип класифікації має свої вимоги та властивості до методів обробки.

Для того, щоб оцінити якість класифікації потрібно врахувати такі аспекти як точність, швидкість обробки, кількість затрачених ресурсів на обробку. Для покращення якості можуть використовуватись різні методи (підготовка якісних даних, підбір оптимальних параметрів алгоритмів тощо).

Під час класифікації зображень можуть виникати різні помилки та проблеми. До найпоширеніших проблем можна віднести, низька якість вхідних даних, велика кількість даних, розбіжності в форматах зображення.

Найбільш поширений метод класифікації зображень документів є методи глибинного навчання такі як нейронні мережі. Нейронна мережа є одним з напрямків штучного інтелекту, який має за мету створити комп'ютерну модель аналітичних механізмів, що застосовуються в людському мозку [3].

Однією з типових задач, які може вирішувати нейронна мережа, є класифікація, передбачення та розпізнавання даних. Важливою особливістю нейронних мереж є їх здатність до самонавчання та розвитку, використовуючи помилки, які виникають під час навчання.

Оскільки робота із документами тісно пов'язана із повсякденним життям людей, відповідно і кількість документів що посвідчують особу вже є досить велика кількість.

Список документів, що посвідчують особу та підтверджують громадянство України:

- паспорт громадянина України;
- паспорт громадянина України для виїзду за кордон;
- дипломатичний паспорт України;
- службовий паспорт України;
- тимчасове посвідчення громадянина України.

Паспорт громадянина України – документ, який посвідчує особу та підтверджує громадянство України [4]. Наразі в Україні паспорт є у двох зразках (Рисунок 1.1 та 1.2).



Рисунок 1.1 – Паспорт громадянина України зразка 2016 року [4]



Рисунок 1.2 – Паспорт громадянина України зразка 1994 року [4]

Паспорт громадянина України для виїзду за кордон – документ, що посвідчує особу, також підтверджує її громадянство, та дає право цій особі на виїзд та в'їзд в Україну (Рисунок 1.3) [5].

Рисунок 1.4 – Дипломатичний паспорт України [6]

Службовий паспорт України – документ, що дозволяє власнику виїзд, в'їзд в Україну, та посвідчує особу (Рисунок 1.5). Паспорт оформляється і видається у порядку, встановленому Президентом України [7].



Рисунок 1.5 – Службовий паспорт України [7]

Тимчасове посвідчення громадянина України – документ, який видається особам, які досягли шістнадцятирічного віку, набули громадянства України та взяли зобов'язання припинити іноземне громадянство протягом двох років з моменту набуття громадянства України (Рисунок 1.6) [8].



Рисунок 1.6 – Тимчасове посвідчення громадянина України [8]

Це лише невелика кількість всіх документів які посвідчують особу. Також варто згадати документи які посвідчують особу та підтверджують їй спеціальний статус [9]. До таких документів відносяться наступні:

- посвідчення водія;
- посвідчення особи без громадянства для виїзду за кордон;
- посвідчення на постійне проживання;
- посвідчення на тимчасове проживання;
- картка мігранта;
- посвідчення біженця;
- проїзний документ біженця;
- посвідчення особи, яка потребує додаткового захисту.

Основні класи на які будуть класифікуватись документи:

- паспорт;
- водійське посвідчення;
- посвідчення особи;
- віза.

Отже, проаналізувавши предметну область можна зрозуміти, що класифікація зображень документів – це досить важлива задача в сучасному світі. Проте при розробці можна зіткнутись із певними проблемами але застосування сучасних методів дозволить досягнути високої якості класифікації.

1.2 Особливості застосування згорткових нейронних мереж для задач класифікації

Задача класифікації документів загалом є рутинною задачею і ручне виконання цієї задачі є трудомістким процесом. Для автоматизованої класифікації документів, що посвідчують особу використовуються різного роду способи та методи.

Нейронні мережі – один з видів машинного навчання. НМ є системою з'єднаних і взаємодіючих між собою простих процесів (штучних нейронів) [10].

Згорткові нейронні мережі (CNN) – один з найбільш ефективних методів для класифікації зображень. Нейронна мережа досягла такої ефективності через можливість автоматично визначати характеристики зображень. Звідси і виходить висока точність класифікації [11].

Згорткові НМ можуть враховувати характеристики об'єктів такі як колір чи форма, що дозволяє використовувати її для класифікації зображень.

Згорткові нейронні мережі є ефективними для роботи із зображеннями, так як вони можуть виявити залежність між пікселями, які знаходяться поруч.

Згорткові НМ можуть автоматично знаходити візуальні ознаки в зображеннях, що дозволяє досить ефективно розпізнавати об'єкти. Це лише маленька частина того, що можуть правильно навчені нейронні мережі.

Згорткові нейронні мережі використовуються для класифікації і розпізнавання об'єктів, осіб на фотографіях або образів по фото чи відео [12].

Згорткова нейронна мережа складається із різних шарів. Чим більше шарів тим складніше буде НМ і відповідно ефективність буде також вища.

Перший шар в згортковій НМ є основним «будівельним шаром», в якому проводяться всі основні обчислення. Мережа також може мати додаткові згорткові шари, що зробить її структуру ієрархічною (Рисунок 1.7) [13].

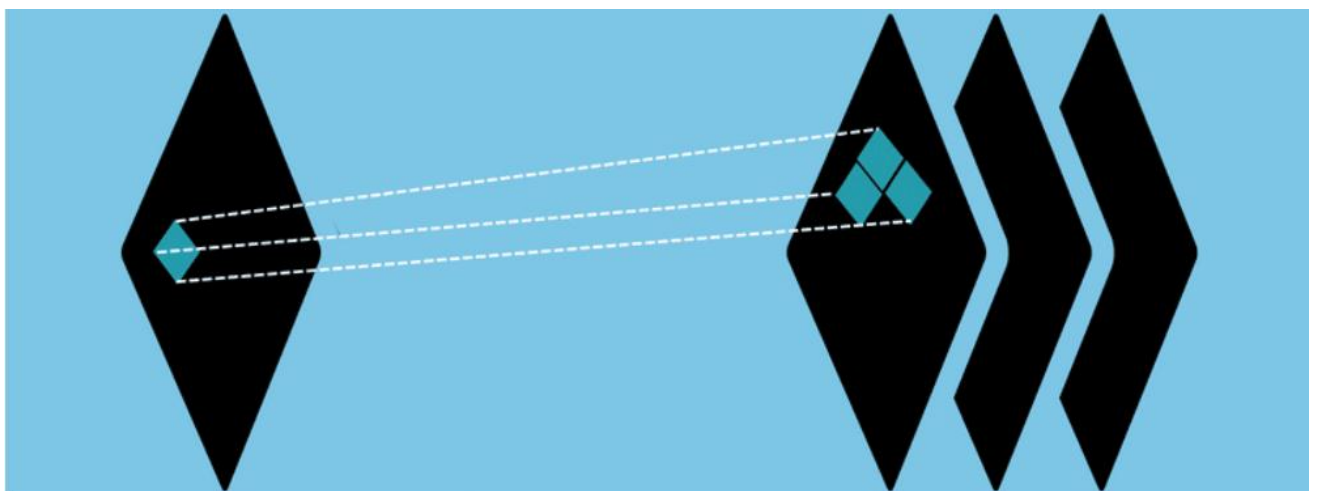


Рисунок 1.7 – Додаткові згорткові шари [13]

Згорткова нейронна мережа – це алгоритм глибокого навчання, який може сприймати вхідне зображення, призначати важливість (вагові значення та зміщення) різним аспектам/об'єктам на зображенні та мати можливість відрізнити один від іншого.

Серед переваг згорткової нейронної мережі можна виділити невелику кількість вагів які потрібно налаштовувати, стійкість до спотворень, універсальність в розподілені об'єктів [14].

Серед недоліків можна виділити велика кількість змінних параметрів, велика кількість параметрів які впливають на результат, але вибираються емпірично [15].

Отже, згорткові нейронні мережі є потужним інструментом для задач класифікації зображень, проте при розробці потрібно враховувати чимало аспектів. Доцільним є використання ЗНМ для реалізації методу нейромережевої класифікації зображень документів що посвідчують особу.

1.3 Аналіз існуючих рішень для подібних задач

Оскільки робота із документами в повсякденному житті зустрічається досить часто, люди намагаються автоматизувати цей процес. Один із етапів цієї автоматизації є класифікація зображень. Для вирішення цієї задачі на сьогоднішній час розроблено багато програмного забезпечення, далі розглянуто деяке з них.

Перший приклад – це служба Amazon Textract, розроблена відомою компанією Amazon. Amazon Textract – сервіс, який забезпечує розпізнавання тексту та інформації з документів, включаючи посвідчення особи. Використовуючи навчені моделі машинного навчання, Textract може витягати текст, фотографії, підписи та інші дані з документів, таких як паспорти, посвідчення особи тощо (Рисунок 1.8) [16].

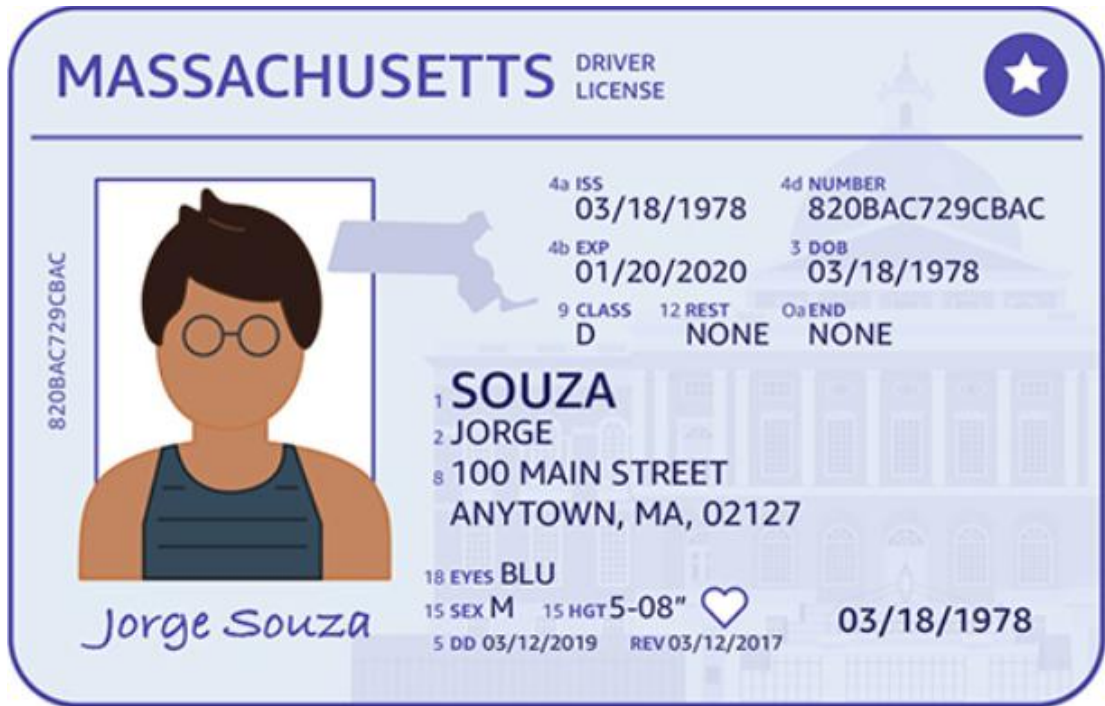


Рисунок 1.8 – Вхідні дані для Amazon Textract [16]

Результатом у цьому випадку є лише пари ключів і значень. Analyze ID також повертає оцінку достовірності та нормалізовані значення, якщо вони доступні.

```

-----
FIRST_NAME           JORGE
LAST_NAME            SOUZA
MIDDLE_NAME
SUFFIX
CITY_IN_ADDRESS      ANYTOWN
ZIP_CODE_IN_ADDRESS  02127
STATE_IN_ADDRESS     MA
STATE_NAME           MASSACHUSETTS
DOCUMENT_NUMBER      820BAC729CBAC
EXPIRATION_DATE      01/20/2020
DATE_OF_BIRTH        03/18/1978
DATE_OF_ISSUE
ID_TYPE              DRIVER LICENSE FRONT
ENDORSEMENTS         NONE
VETERAN
RESTRICTIONS         NONE
CLASS                D
ADDRESS              100 MAIN STREET
COUNTY
PLACE_OF_BIRTH
-----

```

Рисунок 1.9 – Результат роботи Amazon Textract[]

Серед переваг Amazon Textract можна виділити висока точність, зручне API, підтримка різних типів документів, розподілена архітектура.

Серед недоліків одразу виділяється висока вартість користування та залежність результату від якості вхідних даних.

Ще одне програмне забезпечення, яке часто використовується для класифікації документів – OpenCV. OpenCV – відкрите програмне забезпечення з комп'ютерного зору, яке може використовуватися для класифікації зображень документів, що посвідчують особу [17].

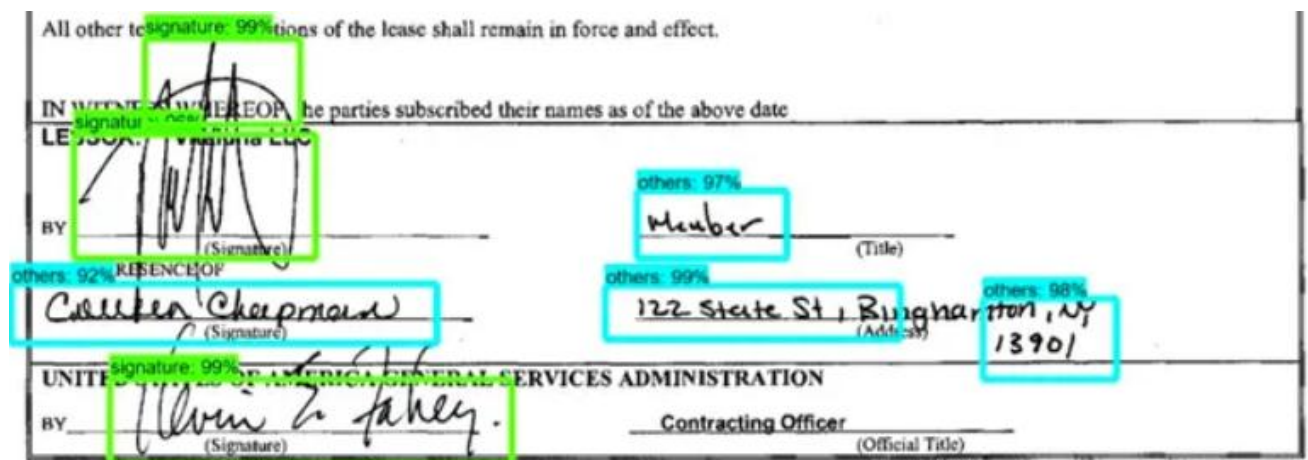


Рисунок 1.10 – Вхідні дані для opencv [17]

Вихідні дані можна встановити вибірково, та записати в файл чи базу даних тощо (Рисунок 1.11).

	Height	Width	AspectRatio	WhitePixelDensity	BlackPixelDensity	DocType
0	1434	3654	2.548117	4484933	754903	Handwritten
1	124	1818	14.661290	174925	50507	Handwritten
2	182	2637	14.489011	377017	102917	Handwritten
3	110	755	6.863636	65120	17930	Handwritten
4	328	1911	5.826220	488694	138114	Handwritten

Рисунок 1.11 – Результат роботи opencv із рукописним документом [17]

Серед переваг одразу виділяється багатофункціональність, підтримка багатьох мов програмування, відкрите програмне забезпечення. Основний недолік – складність в використанні для новачків.

Ще одним популярним рішенням для автоматизації процесів обробки документів є Kofax.

Kofax – це комплексне рішення для автоматизації обробки документів, яке включає функції класифікації різних типів документів, включаючи документи, що посвідчують особу [18]. Воно використовує технології розпізнавання тексту, обробки зображень та навчання з наглядом для автоматичної обробки документів та екстракції даних з них (Рисунок 1.12).

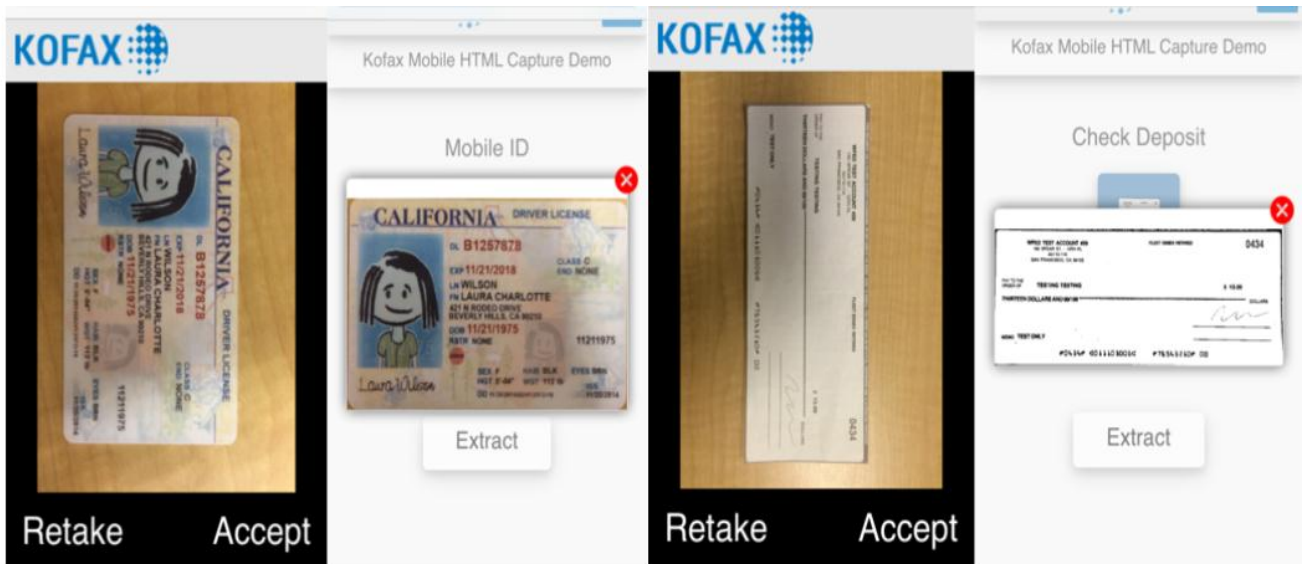


Рисунок 1.12 – Результат роботи Kofax [18]

Kofax може працювати як не лише із сканованими документами, а й і з рукописними, що є великим плюсом. Серед недоліків одразу помітно високу вартість, та складність налаштування.

Провівши аналіз існуючого програмного забезпечення, можна зазначити, що кожний з сервісів має як свої переваги так і недоліки. З основних недоліків можна виділити те, що наявність будь-яких других об'єктів на фото значно ускладнюють класифікацію зображень. Для реалізації методу нейромережевої

класифікації зображень документів що посвідчують особу варто врахувати переваги та недоліки проаналізованого програмного забезпечення.

1.4 Мета, задачі та вимоги до реалізації програмної системи

Метою роботи є розробка методу нейромережевої класифікації зображень документів що посвідчують особу та відповідної програмної системи що використовує метод нейромережевої класифікації зображень документів що посвідчують особу, для чого необхідно вирішити такі задачі:

- розробити метод нейромережевої класифікації зображень документів;
- розробити архітектуру згорткової нейронної мережі для класифікації зображень документів;
- спроектувати структуру інформаційної системи класифікації зображень документів, що використовує технологію нейромережевої класифікації зображень документів;
- виконати програмну реалізацію інформаційної системи;
- провести тестування розробленої інформаційної системи класифікації зображень документів.

Розділ 2 Метод нейромережевої класифікації зображень документів що посвідчують особу

2.1 Опис і кроки методу нейромережевої класифікації зображень документів

Метод нейромережевої класифікації зображень документів що посвідчують особу призначений для перетворення вхідних даних у вигляді навчальної множини зображень документів для навчання, тестової множини зображень документів для самоконтролю та робочого зображення документа для класифікації у вихідні дані у вигляді визначеного клас документу та оцінки коректності класифікації й містить кроки навчання й самотестування нейронної мережі навчальною множиною зображень документів, а також нейромережевої класифікації робочого зображення документа (Рисунок 2.1).



Рисунок 2.1 – Візуалізація роботи методу класифікації зображень

Вхідними даними методу нейромережевої класифікації зображень документів, що посвідчують особу є навчальна вибірка зображень документів що посвідчують особу, тестова множина зображень документів для самоконтролю та робоче зображення документа для класифікації (Рисунок 2.2).

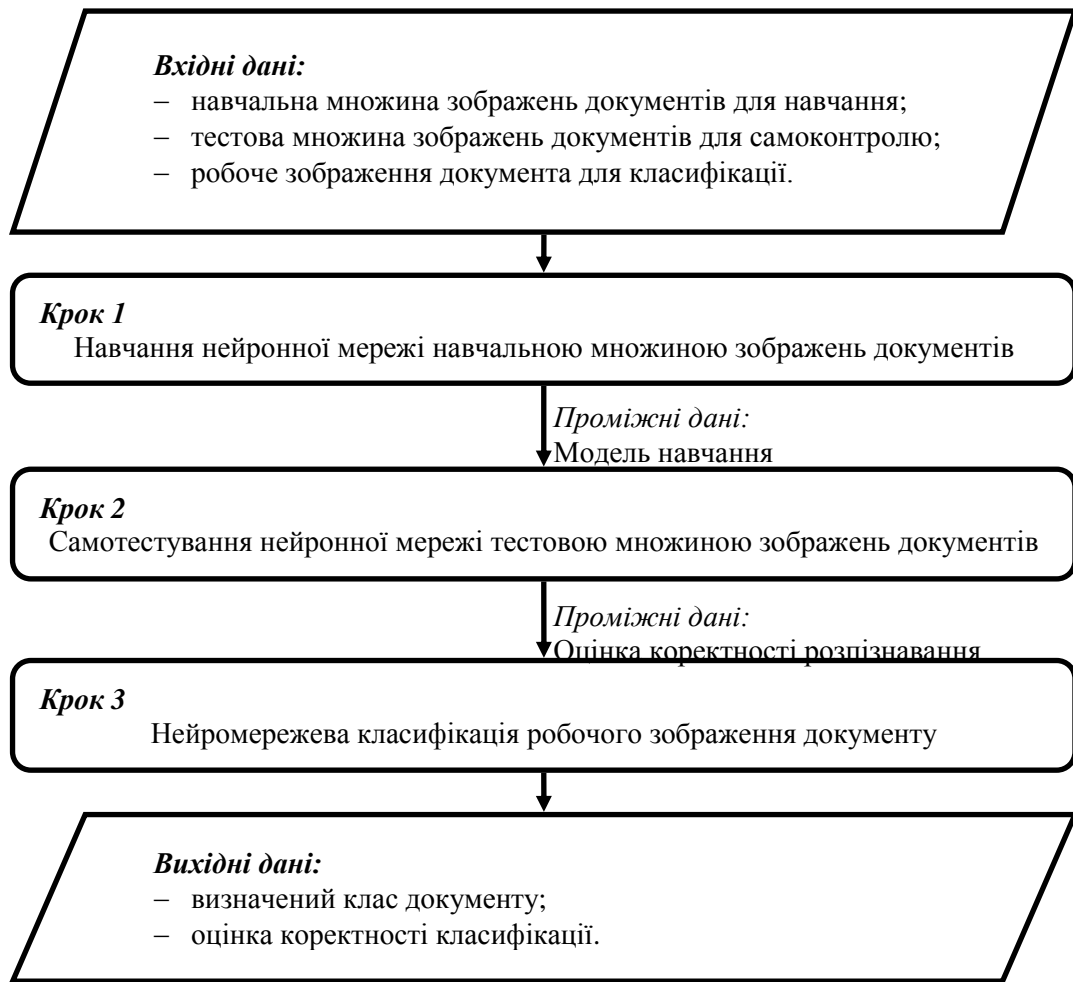


Рисунок 2.2 – Схема методу нейромережевої класифікації зображень документів що посвідчують особу

Процес класифікації зображень можна поділити на декілька етапів: навчання нейронної мережі навчальною множиною зображень документів, самотестування нейронної мережі тестовою множиною зображень документів та нейромережева класифікація робочого зображення документа.

Навчання нейронної мережі відбувається навчання згорткової нейронної мережі за допомогою вибірки зображень, що було обрано в складі вхідних даних.

Самотестування нейронної мережі відбувається наступним чином:

- із навчальної вибірки автоматично вибирається 20% для тестування;
- оцінюється коректність навченої моделі;
- якщо відсоток похибок більше 20% пропонується перенавчити модель із більшою кількістю зображень.

Нейромережева класифікація робочого зображення документа полягає у тому, що користувач обирає фото документа та завантажує у вже навчену модель. Після чого відбувається процес розпізнавання фото.

Вихідні дані методу нейромережевої класифікації зображень документів що посвідчують особу складають дані, які користувач отримує як результат, а саме прогнозований клас завантаженого документа, а також оцінку коректності класифікації.

Отже, в даному розділі було розроблено метод нейромережевої класифікації зображень документів що посвідчують особу. Метод призначений для перетворення вхідних даних у вигляді навчальної множини зображень документів для навчання, тестової множини зображень документів для самоконтролю та робочого зображення документа для класифікації. Вихідні дані подаються у вигляді визначеного клас документа та оцінки коректності класифікації й містить кроки навчання й самотестування нейронної мережі навчальною множиною зображень документів, а також нейромережевої класифікації робочого зображення документа.

2.2 Особливості використання методу нейромережевої класифікації зображень документів

2.2.1 Навчання нейронної мережі навчальною множиною зображень документів

Нейронні мережі на даний момент є одним із ключових інструментів, який використовується у машинному навчанні. НМ здатні вивчати та розуміти складні залежності та патерни у вхідних даних та використовуються для вирішення широкого спектру завдань.

Одним з базових етапів використання нейронної мережі є налаштування параметрів навчання нейронної мережі. Серед важливих параметрів можна виділити [19]:

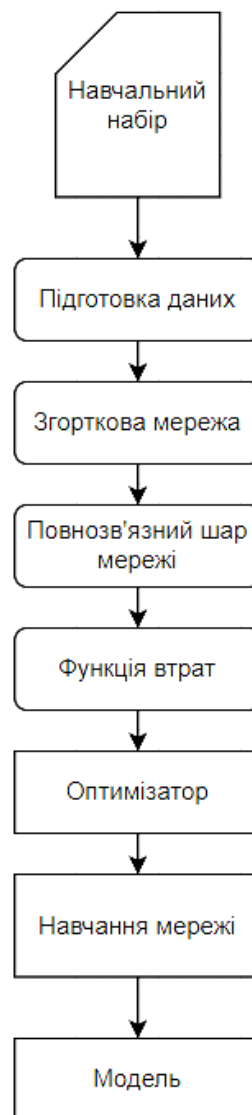
- кількість епох;

- швидкість навчання;
- оптимізатор.

Оптимізатор – алгоритм який використовується для покращення ефективності навчання згорткових нейронних мереж [20].

Навчання класифікатора в згортковій нейронній мережі (CNN) полягає у встановленні вагів та зміщень мережі з метою зменшення помилки класифікації на тренувальному наборі даних та отримання точної класифікації на тестових даних.

На Рисунку 2.3 зображена схема навчання класифікатора.



Рисунку 2.3 – Схема навчання класифікатора

В цій схемі навчальний набір подається у згорткову нейронну мережу після підготовки даних. Після проходження крізь мережу вихідні дані піддаються обробці і проходять через повнозв'язний шар. Наступним кроком є виклик функції втрат, яка допомагає визначити наскільки добре працює модель. Оптимізатор використовує цю інформацію, щоб покращити точність мережі шляхом корекції вагів. Після навчання мережі отримується модель, яка може бути використана для класифікації нових даних.

2.2.2 Самоперевірка нейронної мережі тестовою множиною зображень документів

Наступним кроком у методі нейронмережевої класифікації зображень документів що посвідчують особу буде тестування моделі на нових зображеннях. Це допоможе оцінити загальну ефективність моделі та визначити, чи можна застосовувати її на практиці (Рисунок 2.4).



Рисунок 2.4 – Схема тестування згорткової нейронної мережі

Тестування згорткової нейронної мережі складається із декількох етапів, а саме:

- підготовка тестового набору даних для тестування можна використовувати окремий набір зображень або партію зображень які були використані для навчання мережі;
- підготовка згорткової нейронної мережі включає завантаження моделі з файлу та визначення параметрів мережі;
- подача тестових даних у мережу такі як зображення, подаються у вхідний шар, після чого вони проходять через різні шари(згортки, пулінг).
- результат згорткових та пулінгових шарі обробляються, та визначається оцінка для кінцевого вихідного шару;
- отримані результати порівнюються з правильними відповідями, та оцінюється точність мережі. Часто використовується метрика точності класифікування, така як точність, чутливість, специфічність.
- після тестування потрібно проаналізувати результати, та визначити які класи були класифіковані правильно а які неправильно. Це допоможе виявити слабкі місця в мережі та покращити її точність.

Загалом, побудова згорткової нейронної мережі для обробки зображень може зайняти багато часу та зусиль, але це може стати ефективним інструментом для розв'язання багатьох завдань в області комп'ютерного зору та обробки зображень, в тому числі і для розробки методу нейромережевої класифікації зображень документів що посвідчують особу.

2.2.3 Нейромережева класифікація робочого зображення документу

Одним із напрямків, де використовуються нейронні мережі є класифікація документів. Такий підхід дозволяє комп'ютеру «навчитись» визначати, розпізнавати до якої категорії належить той чи інший документ. Далі наведено опис процесу класифікації зображень документів що посвідчують особу.

Класифікатор – алгоритм або модель машинного навчання, який навчається розпізнавати, класифікувати об’єкти за певними властивостями [21].

Зазвичай класифікатор працює на основі певних властивостей або ознак об’єктів, які він повинен класифікувати (Рисунок 2.5). Наприклад, класифікатор може класифікувати вид транспорту на основі їхньої форми, розміру тощо.



Рисунок 2.5 – Схема роботи класифікатора

Для початку вхідна інформація проходить через згорткову нейронну мережу, після чого відфільтровані дані піддаються підготовці до класифікації. Після цього дані піддаються фінальній обробці перед класифікації. Наступним кроком є виклик функції активації. На виході отримуємо прогнозований клас для вихідного зображення.

Отже, для реалізації методу нейромережевої класифікації зображень документів що посвідчують особу необхідним є знайти та підготувати зразки

робочих вхідних даних, які можна використовувати для навчання згорткової нейронної мережі. Переважно для навчання згорткової нейронної мережі використовують зображення форматів JPEG або PNG.

2.3 Розробка архітектури нейронної мережі для класифікації зображень документів що посвідчують особу

2.3.1 Використання бібліотеки TensorFlow для класифікації зображень документів

Оскільки останнім часом штучний інтелект зробив великий крок вперед, почали розробляти та випускатись бібліотеки, які спрощують роботу із нейронними мережами. Одна із таких бібліотек – TensorFlow.js.

TensorFlow.js – це бібліотека машинного навчання з відкритим вихідним кодом, яка дозволяє виконувати навчання моделей машинного навчання на браузері або на Node.js [22].

TensorFlow має декілька класів та методів, які дозволяють працювати із згортковими нейронними мережами. Основні з них:

- «tf.layers.conv2d» – клас, який дозволяє створити шар згортки, який буде виконувати операції із вхідними даними. Має перелік параметрів, серед них розмір ядра згортки, кількість фільтрів, крок згортки, функцію активації;

- «tf.layers.maxPooling2d» – клас, який дозволяє створити шар пулінгу, який зменшує розмір ознак, отриманих із попереднього шару згортки;

- «f.layers.flatten» – клас, який дозволяє перетворювати двовимірні масиви в n-вимірний масив ознак;

- «tf.layers.dense» – клас, який дозволяє створити пов'язаний шар, який використовується для класифікації або регресії;

- «tf.train.AdamOptimizer» – клас, який відповідає за функцію оптимізатор в нейронній мережі.

Основна схема роботи TensorFlow із згортковою нейронною мережею складається із наступних кроків:

- підготовка даних – підготовка тренувальних даних та тестових даних, їх нормалізація та перетворення в тензор;
- створення моделі – ініціалізація нейронної мережі методом «tf.sequential» та додавання різних типів шарів;
- компіляція моделі – крок, на якому встановлюються початкові параметри навчання, такі як оптимізатор тощо;
- навчання моделі – відбувається за допомогою тестових даних з використанням методу «model.fit()» який приймає вхідні дані, кількість епох як вхідні параметри;
- оцінка точності моделі – перевірка точності моделі на тестових даних з використанням методу «model.evaluate()»;
- передбачення класу зображення – використовується вже навчена модель для визначення класу зображень з нових даних.

Кроки можуть змінюватись в залежності від конкретної задачі, проте загальна схема роботи TensorFlow із згортковою нейронною мережею залишається незмінною. За допомогою TensorFlow.js можна реалізувати наступну архітектуру згорткової нейронної мережі, схематичне зображення якої подано на рисунку 2.6.



Рисунок 2.6 – Архітектура згорткової нейронної мережі

Вхідний шар – цей шар буде приймати зображення документа. Розмір зображення може бути збільшеним або зменшеним за допомогою згорток або пулінгу.

Перший згортковий шар – це шар складається з кількох шарів згортки, які застосовуються до вхідного зображення. Кожен із цих шарів згортки використовується для фільтрування та відповідає за виявлення конкретної ознаки на зображенні.

Шар пулінгу – наступним шаром після першого згорткового є шар пулінгу, який відповідає за розмір вхідного зображення та зниження кількості параметрів моделі.

Другий згортковий шар – шар подібний до першого, проте він виконує ті ж дії але вже на вже після зміни розміру зображення та з більш високим рівнем абстракції.

Ще один шар пулінгу використовується для того щоб зменшити розмір зображення та кількість параметрів.

Повнозв'язний шар – це шар який використовується після проходження останнього шару пулінгу, вихідні дані перетворюються у одновимірний вектор та передається до повнозв'язного шару. Повнозв'язний шар об'єднує всі ознаки та здійснює класифікацію.

Вихідний шар – це останній шар визначає кількість класів класифікації та використовує функцію активації, яка в свою чергу визначає ймовірність приналежності до кожного класу.

Отже, робота з згортковими нейронними мережами в TensorFlow.js дозволяє створювати потужні засоби для обробки зображень в вебсередовищі. За допомогою TensorFlow.js можна навчати згорткові нейронні мережі, які здатні класифікувати, візуалізувати та обробляти зображення, що робить його ідеальним інструментом для розробки інтерактивних вебдодатків, сервісів та аналізу зображень.

2.3.2 Налаштування гіперпараметрів моделі TensorFlow

Важливим етапом у розробці згорткової нейронної мережі – є правильне налаштування гіперпараметрів моделі. Гіперпараметри – це параметри, які

встановлюються перед початком тренування моделі, та визначають поведінку моделі під час тренування [23].

Розмірність вікон згортки – цей параметр визначає розмір вікна згортки, яке під час обробки рухається по зображенню. Чим більше вікно згортки тим більше регіональної інформації воно надає, проте потребує більше обчислювальних ресурсів;

Кількість фільтрів – параметр, який визначає кількість фільтрів на кожному шарі згортки. Тобто кожен фільтр виділяє окрему ознаку зображення (кут, рівність кольорів тощо). Більша кількість фільтрів може допомогти збільшити точність моделі;

Функція активації – параметр, який визначає функцію, яка використовується для активації значень у вихідному тензорі після проходження через згортковий шар;

Функція втрат – параметр, який визначає функцію, яка використовується для обчислення втрат під час тренування моделі;

Кількість шарів – параметр, який визначає кількість згорткових шарів, що будуть використовуватись в мережі. Чим більша кількість шарів, тим вища точність моделі. Проте потрібно зауважити, що велика кількість шарів може призвести до перенавчання моделі.

Розмір пакета – параметр, який визначає кількість зображень, які будуть одночасно обробляться моделлю під час тренування. Більші значення можуть збільшити швидкість навчання, проте також буде більше використовуватись оперативної пам'яті.

Кількість епох – параметр, який визначає кількість повторень навчального циклу для всього датасету. Чим більше епох тим модель має краще здатність узагальнювати риси, проте це також може призвести до перенавчання.

Швидкість навчання – параметр, який визначає, якою мірою ваги моделі оновлюються під час тренування. Чим більші значення тим більша точність, але водночас можуть виникати проблеми «перестрілки» і гальмування збільшення точності.

Отже, варто зазначити, що кожен із цих параметр є досить важливим та значною мірою впливає на результативність моделі. Тому налаштування цих параметрів є досить важливою задачею при розробці згорткової нейронної мережі для реалізації методу нейромережевої класифікації зображень документів що посвідчують особу

2.3.3 Операції згорткової нейронної мережі TensorFlow для класифікування зображень

Для того щоб спростити та пришвидшити розробку методу нейромережевої класифікації зображень документів що посвідчують особу було прийнято рішення використовувати бібліотеку TensorFlow.

Діаграма послідовності класифікації документів згортковою нейронною мережею для може включати наступні кроки:

1. Завантаження документа;
2. Попередня обробка зображення;
3. Передача зображення через згортковий шар (редагування зображення);
4. Застосування функцій активації;
5. Застосування підсумовуючого шару;
6. Повторення кроків, для виявлення ознак вищих рівнів;
7. Повнозв'язний шар для класифікування документа на певний клас.

На Рисунку 2.7 зображено діаграму послідовності для методу нейромережевої класифікації зображень документів що посвідчують особу

В цілому є два згорткових шари, кожен з яких використовується для виявлення все більш і більш абстрактних ознак. Потім йде повнозв'язний шар, який робить прогноз на основі отриманого вектору ознак із згорткових шарів.

Функція активації використовується для не-лінійності між шарами, що дозволить моделі робити більш складні прогнози.

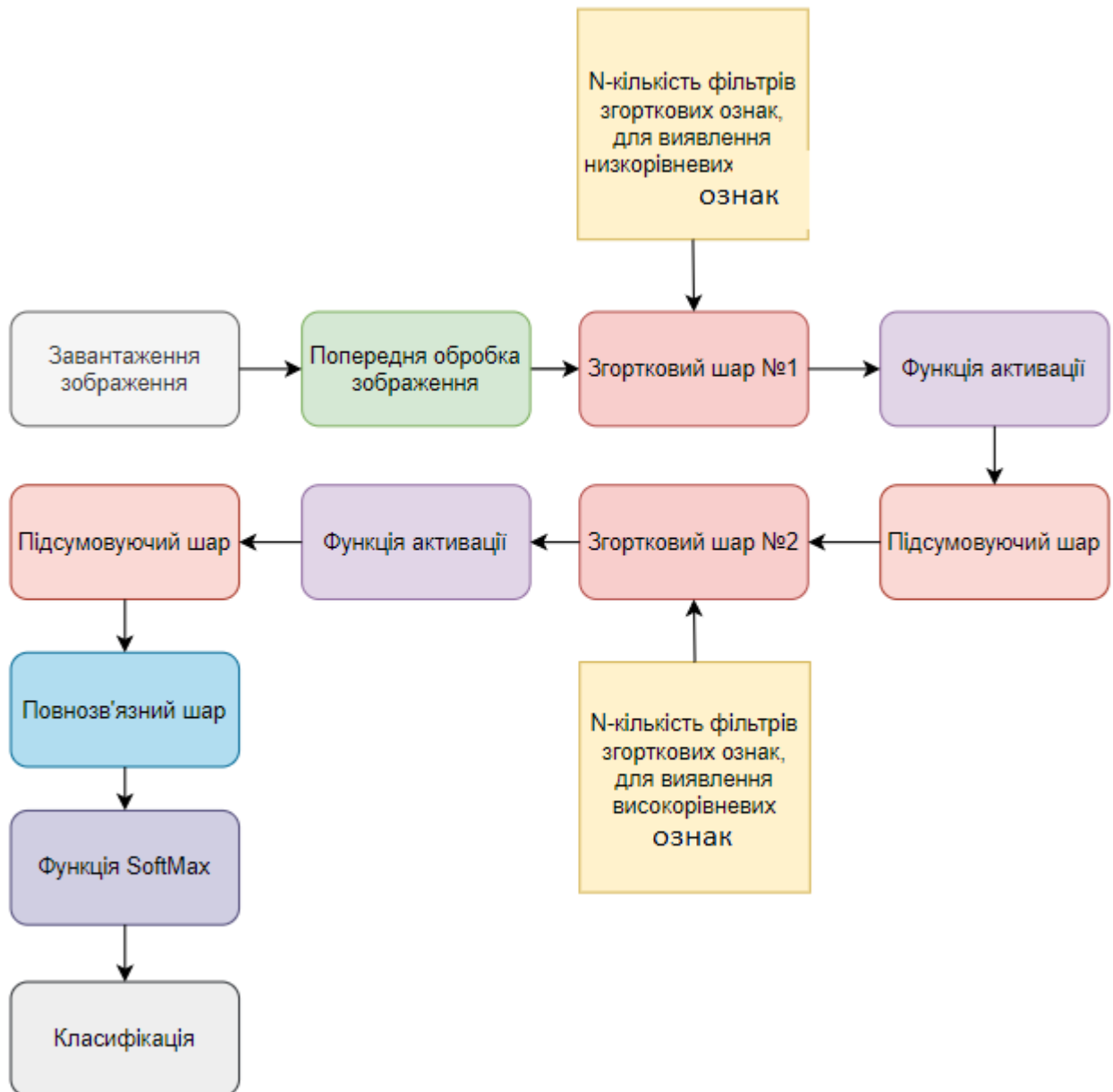


Рисунок 2.7 – Діаграма послідовності операцій згорткової нейронної мережі за класифікування зображень

Підсумовуючий шар скорочує вимірність зображення з кожного згорткового шару. Тобто цей шар може зменшити розмірність зображення та зберегти найважливіші виявлені ознаки. Підсумовуючий шар поділяє зображення на невеликі області та замінює її одним значенням. Це значення зазвичай представляє максимальне (MaxPooling) або середнє (AVGPooling) значення в цій області.

Функція SoftMax знаходить ймовірності для кожного з класів, як можна передбачити.

Кінцевий шар – класифікація, визначає, до якого класу належить вхідне зображення.

Діаграма активності в згортковій нейронній мережі для класифікації документів може включати наступні кроки:

1. Вхідні зображення розміром 224x224x3;
2. Створення згорткового шару (Conv2D) із фільтрами;
3. Створення шару максимального пулінгу (MaxPooling2D);
4. Повторення максимального пулінгу із збільшенням фільтрів;
5. Отримання результату та передача до повнозв'язного шару;
6. Отримання результату класифікації.

На Рисунку 2.8 зображено приклад діаграми активності згорткової нейронної мережі для класифікації документів.

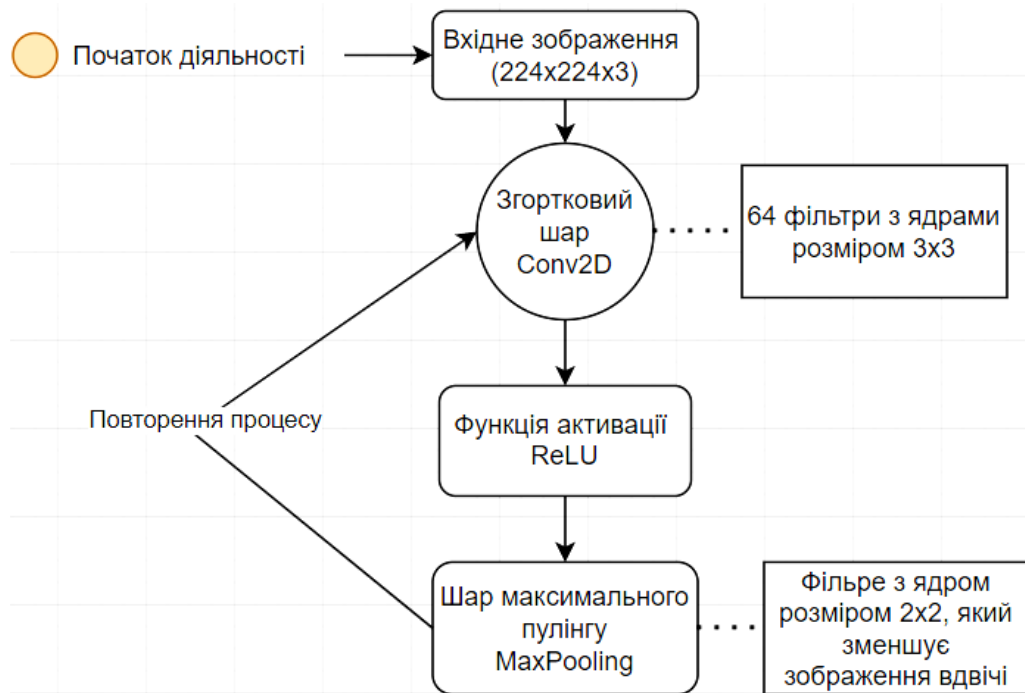


Рисунок 2.8 – Діаграма активності згорткової нейронної мережі

Із діаграми можна зрозуміти, що вхідні зображення мають формат 224x224x3, тобто мають розмірність 224 на 224 пікселів та є кольоровими.

Першим шаром є згортковий шар (Conv2D), що містить в собі 64 фільтри з ядрами розміром 3x3, а також функцію активації ReLU.

Після згорткового шару використовується шар максимального пулінгу (MaxPooling2D), який містить в собі фільтр 2×2 , що зменшує зображення вдвічі.

Після цього можна ще повторити процес декілька раз, допоки не буде досягнуто бажаного результату.

Фільтри(ядра) є важливим елементом згорткової нейронної мережі. Вони є невеликими матрицями ваг, які проходять по вхідному зображенню, виконуючи операцію згортки, завдяки чому вилучають важливі ознаки.

До прикладу, якщо на вхід завантажити зображення 32×32 , в згортковому шарі з трьома фільтрами розміром 3×3 , кожен фільтр буде мати свою матрицю ваг розміром 3×3 , де кожен елемент відображає значення ваги пікселя (Рисунок 2.9).

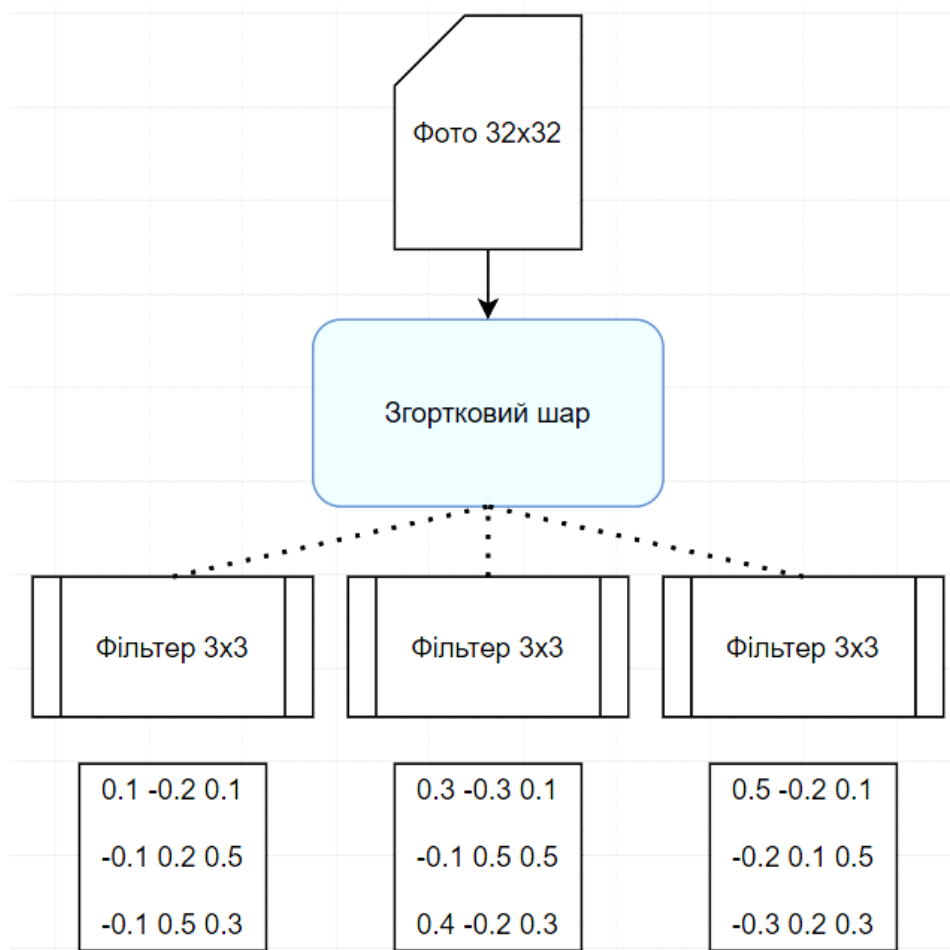


Рисунок 2.9 – Матриці ваг фільтрів

Кожен із цих фільтрів рухається по зображенню, та застосовується до кожного 3x3 блоку пікселів. Потім відповідний блок пікселів перемножується на відповідну матрицю ваг фільтра, після чого отриманні значення сумуються, утворюючи один піксель у вихідному зображенні.

Отже, для того щоб розробити ефективний метод для класифікації зображень документів що посвідчують особу, потрібно врахувати багато аспектів, та продумати архітектуру нейронної мережі. Проте TensorFlow дозволяє значно спростити цей етап розробки.

2.3.4 Архітектура нейронної мережі

Згорткові шари забезпечують виявлення різних ознак у зображенні, починаючи від простих орієнтаційних ребер, до більш складних шаблонів, таких як форми обличчя або текст. Загальна схема використання нейронної мережі для класифікації зображення подана на рисунку 2.10.

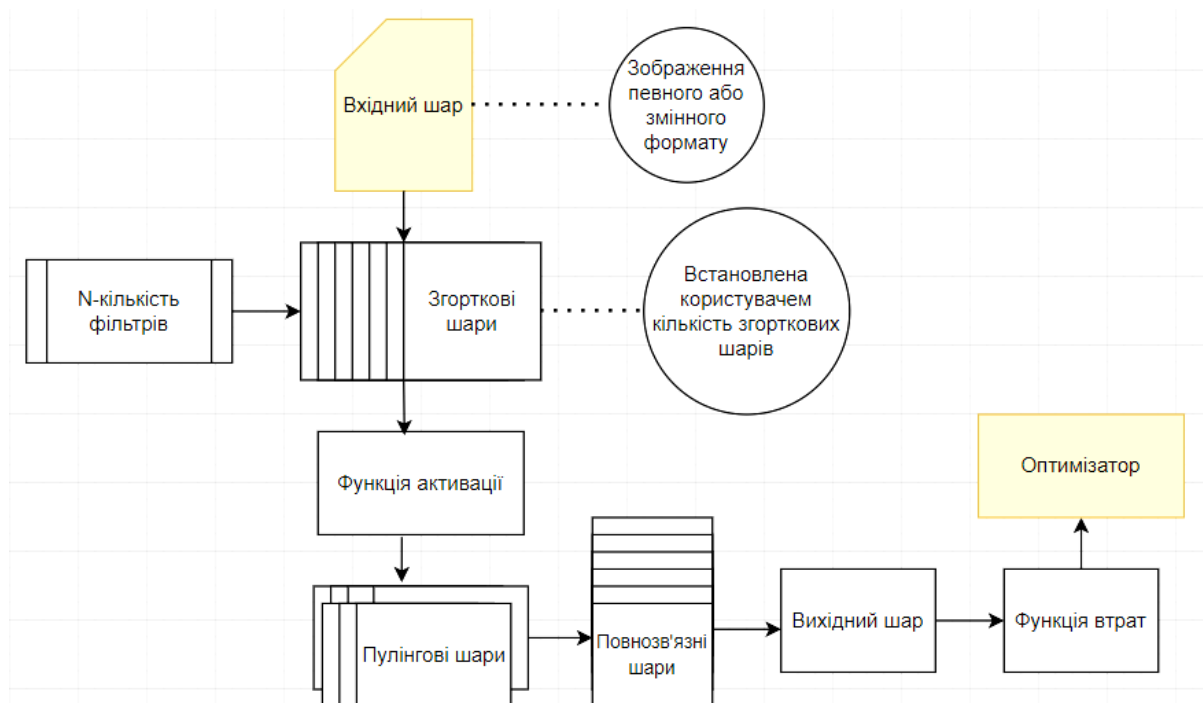


Рисунок 2.10 – Схема використання нейронної мережі для класифікації зображення

Вхідними даними є зображення певного документа. Розмірність зображення може бути одразу визначеним, а може бути змінним, тобто користувач сам встановлює розмірність на адмін-панелі.

Згорткові шари – спеціальні шари, які використовують фільтра або ядра, для виявлення різного роду візуальних ознак. Кожен згортковий шар, має мати певну кількість фільтрів, ядер які будуть застосовуватись до зображення. До прикладу, на Рисунку 2.11 візуально зображено роботу згорткових шарів.



Рисунок 2.11 – Візуалізація роботи згорткового шару

Функція активації це функція яка викликається після проходження кожного згорткового шару може бути викликана функція активації. Ця функція дає змогу додати нелінійність до мережі.

Пулінгові шари – шари, які використовуються для зменшення розміру вихідних зображень та виділення найважливіших ознак. Зазвичай використовується спеціальний метод «максимум». В цьому методі регіони зображення замінюються на одне значення – максимальне в цьому регіоні.

Повнозв'язні шари – шари після певної кількості згорткових та пулінгових шарів використовуються повнозв'язні шари, які зєднують всі важливі ознаки, виявленні у попередньо пройдених шарах, з різними класами. Це відбувається задля того щоб здійснити сотаточну класифікацію. Зазвичай виконується декілька повнозв'язних шарів із функціями активації.

При реалізації вихідного шару, розробляється певна кількість нейронів, відповідну кількості класів, на які потрібно класифікувати зображення. Зазвичай використовуються спеціальні функції «SoftMax», щоб отримати ймовірність для кожного класу.

Функція втрат – функція, для тренування мережі. Ця функція визначає різницю між прогнозованим значенням мережі та правильним мітками.

Оптимізатор використовується для налаштування вагових коефіцієнтів згорткової нейронної мережі під час тренування. Це робиться для того, щоб мінімізувати функцію втрат.

В цілому така реалізація методу може використовуватись як базова. Для деяких зображень, можуть змінюватись деякі деталі в реалізації, такі як, кількість шарів, розмірність фільтрів, кількість нейронів тощо.

Отже, в даному підрозділі була спроектована архітектура нейронної мережі для методу класифікації документів що посвідчують особу. Також було описано основні функції такі як оптимізатор, функції втрат.

2.4 Підготовка зразків робочих вхідних даних для класифікатора

Для навчання тестування та оцінки розробленого методу нейромережевої класифікації зображень документів що посвідчують особу необхідно знайти відповідний якісний набір даних, адже від нього залежить якість роботи методу.

Датасет із документами українського зразку в відкритому доступі не знайдено, тому в роботі було використано один з існуючих відкритих датасетів із зображеннями закордонних документів.

Для навчання нейронної мережі потрібно спочатку завантажити зображення з джерела. У якості джерела зображень будуть використовуватись датасети.

Датасет – це набір даних, який використовується для навчання, тестування та оцінки алгоритмів машинного навчання. Датасет містить в собі дані про об'єкти, що мають бути класифікованими або передбаченими. Для того щоб зібрати датасет потрібно для початку знайти дані про об'єкти які будуть класифікуватись. Це може бути будь-який тип даних, серед найпопулярніших:

- аудіо;
- відео;
- текст;
- числові дані;

У випадку де потрібно класифікувати документи що посвідчують особу, найкраще підходить датасет із зображеннями.

Наступним кроком стане обробка зображення. Тобто, якщо виконується задача класифікації, може знадобитись привести зображення до певного формату, розміру тощо.

Потрібно зауважити, що для ефективного навчання моделі, датасет має бути достатньо великим та різноманітним. Лише тоді результат класифікування буде мати високу точність.

При роботі із датасетом важливо документів для навчання та тренування моделі класифікації зображень, що посвідчують особу, є кілька важливих моментів, які слід врахувати.

Розмітка даних є важливим кроком при підборі датасету. Датасет, повинен бути належним чином різноманітним з відповідними мітками класу, наприклад, «паспорт», «водійське посвідчення», «id-картка» тощо.

Розподіл даних є досить важливим аспектом в підборі датасету. Забезпечення приблизно однакової кількості зображень для кожного класу. Це допоможе уникнути перекоосу в моделі навчання.

Перекриття даних використовується для того, щоб покращити надійність моделі та уникнути перенавчання можна використати метод перекриття даних. Цей метод включає збільшення розмаїтості даних шляхом застосування трансформації до існуючих зображень, таких як зміщення, обертання, масштабування, розмивання тощо.

Перевірка якості даних – обов'язковий крок перед початком навчання моделі. Потрібно перевірити датасет на наявність неправильного розміру зображень, наявність дефектів, шумів, артефактів тощо. Важливо прибрати аномалії, або непридатні для навчання, тестування зображення із датасету.

Після того як датасет готовий, його потрібно розділити на валідаційний, тренувальний та тестовий. Тренувальний датасет буде використовуватись при навчанні моделі. Тестовий датасет буде використовуватись для оцінки точності моделі на незалежних даних. Валідаційний датасет використовується для налаштування гіперпараметрів моделі. Зазвичай використовується відношення 75-80% для тренувальної підвибірки, 5-10% для тестової підвибірки, 5-10% для валідаційної підвибірки. Такий спосіб розподілення зображень допоможе більш точно оцінити продуктивність моделі на невидимих раніше даних, та уникнути перенавчання.

Перед подачею даних на вхід моделі хорошим рішенням буде проведення нормалізації. Наприклад, масштабування значень пікселів в діапазоні $[0,1]$, або стандартизацію даних. Якщо датасет має дисбаланс у кількості зображень різних класів, можна застосувати стратегії для збалансування даних, такі як випадкова вибірка, дублювання тощо. Також можливо використати змінення ваг для класів в яких менше всього зображень.

Досить важливим моментом є використання перевірочної підвибірки для налаштування гіперпараметрів моделі (розмір шарів, швидкість навчання, кількість епох тренування тощо). Це допоможе покращити продуктивність моделі та уникнути перенавчання.

Далі потрібно перетворити зображення в тензор. Тензор – багатовимірний масив, що зберігає дані у вигляді числових значень [24]. Тензор також може

бути представлений як n -вимірний масив де n -кількість вимірів, де кожний вимір містить певну кількість значень.

Для зображення тензор може бути заданий як (batch_size, height, width, channels), де:

- batch_size – кількість зображень у пакеті, який передбачається за одну ітерацію проходу мережі;
- height – висота зображення в пікселях;
- width – ширина зображення в пікселях;
- channels – кількість каналів зображень (для чорно-білого 1 канал, для кольорового зображення – 3 або більше).

Отже, якщо у датасеті буде 32 зображення розміром 224 на 224 пікселів з трьома каналами (кольорове зображення), то тензор буде мати наступний вигляд (32, 224, 224,3).

Для забезпечення правильного формату та діапазону значень тензора необхідно виконати обробку даних. Один з можливих підходів – нормалізація значень тензора до діапазону від 0 до 1. Це означає, що мінімальне значення тензора буде відображено як 0, а максимальне – як 1. Цей підхід дозволяє забезпечити однаковий масштаб значень в тензорі, що покращує процес навчання та передбачення в мережі.

Обраний датасет VID надає загальнодоступний набір іноземних ідентифікаційних зображень документів. Даний датасет був створений для вирішення трьох найбільш поширених проблем в сфері комп'ютерного зору.

Датасет містить якісні зображення однакового розміру, що досить важливо при навчання нейронної мережі. Датасет містить в собі:

- лицьова та зворотна сторона водійських прав;
- лицьова та зворотна сторона паспорту;
- лицьова та зворотна сторона реєстру фізичних осіб.

Всі дані з документів розмиті, щоб не порушувати авторські права (Рисунок 2.12).

REPUBLICA FEDERATIVA DO BRASIL
 MINISTÉRIO DAS CIDADES
 DEPARTAMENTO NACIONAL DE TRÁNSITO
 CARTEIRA NACIONAL DE HABILITAÇÃO

INTERPRINT LTDA.

VÁLIDA EM TODO O TERRITÓRIO NACIONAL
 315694779

PROIBIDO PLASTIFICAR
 867946670

NOME
 BINI SPADOTO TELMA

DOC. IDENTIDADE / ORG. EMISSOR UF
 342237482 SJS GO

CPF
 111.060.240-52

DATA NASCIMENTO
 14/08/1977

FILIAÇÃO
 APUA SANTINELLI RUBANO
 DUARTE
 DERIGGI FAZENDEIRO GERALDI

PERMISSÃO ACC CAT. HAB.
 E

Nº REGISTRO
 92536413061

VALIDADE
 27/02/1971

1º HABILITAÇÃO
 14/02/1973

OBSERVAÇÕES

ASSINATURA DO PORTADOR

LOCAL
 ITUAÇU-BA

DATA EMISSÃO
 09/05/2007

MAICO SEYEV PES
 55915257670
 GO955536202

ASSINATURA DO EMISSOR

DETRAN - MG (MINAS GERAIS)

Рисунок 2.12 – Зразок зображення документу що посвідчує особу з датасету VID

Також датасет вирішує проблему з сегментацією зображення, оскільки до кожного фото надається сегментована версія фото (Рисунок 2.13).



Рисунок 2.13 – Сигментована версія фото в датасеті VID

Отже, даний датасет VID досить добре підходить для навчання нейронної мережі, оскільки містить велику кількість високоякісних зображень, із правильно підібраними властивостями, що значно спрощує використання та тестування згорткової нейронної мережі.

2.5 Проектування структури інформаційної системи класифікації зображень документів

Для створення ефективної інформаційної системи в якій користувач зможе взаємодіяти із датасетом та працювати із нейромережевим методом для класифікації зображень документів що посвідчують особу, потрібно розбити велику інформаційну систему на менші підсистеми.

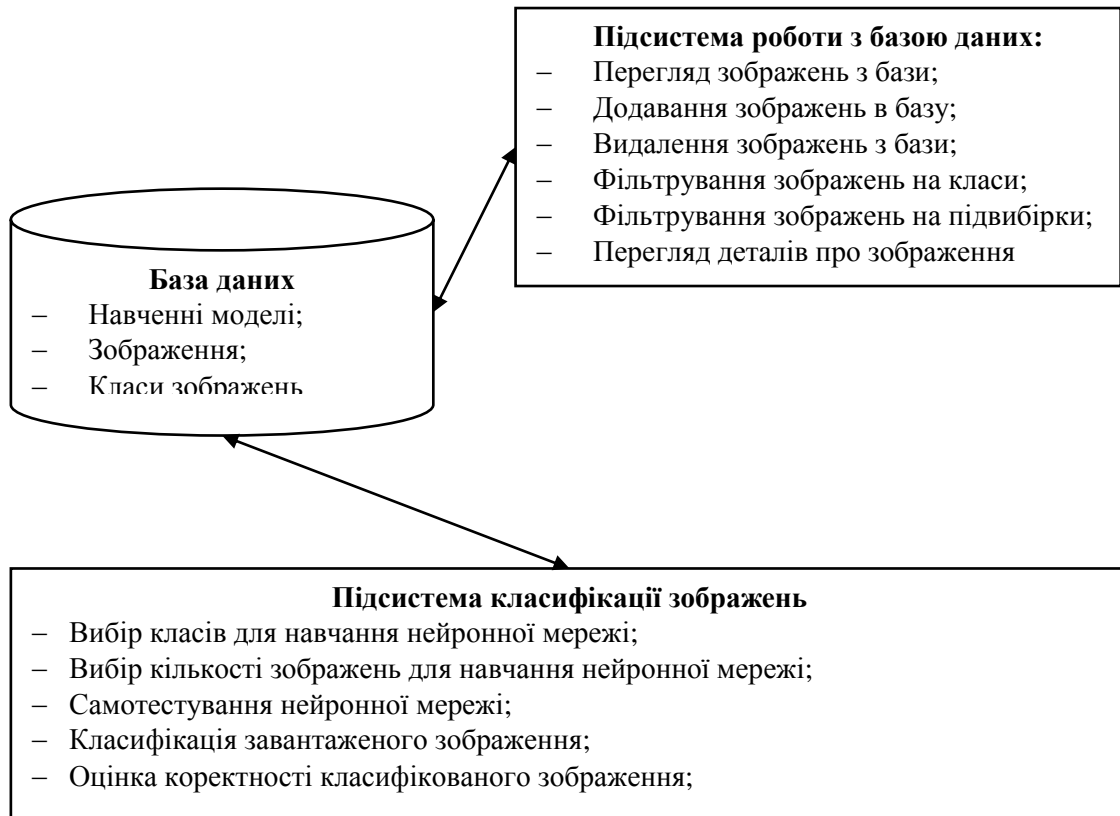


Рисунок 2.14 – Підсистеми інформаційної системи класифікації зображень документів

Підсистема роботи з базою даних:

- користувач може переглядати всю базу зображень (перегляд зображень);
- користувач може додавати нові зображення до існуючої бази зображень(додавання зображень);
- користувач має змогу видаляти зображення із існуючої бази зображень(видалення зображень);
- користувач має змогу фільтрувати зображення на існуючі в базі класи(фільтрування зображень на класи);
- користувач має змогу фільтрувати зображення на підвибірки (тестувальна, навчальна);
- користувач має змогу переглядати властивості зображення, наприклад розмір, тощо(перегляд деталей зображення).

Підсистема класифікації зображення:

- вибір класів для навчання нейронної мережі, тобто користувач має змогу вибрати класи для навчання нейронної мережі;
- користувач має змогу обрати кількість зображень кожного класу які будуть використані для навчання нейронної мережі(вибір кількості зображень для навчання нейронної мережі);
- користувач має змогу оцінити якість навченої моделі завдяки методу самотестування (самотестування нейронної мережі);
- користувач має змогу завантажити зображення в навчену модель та отримати прогнозований клас (класифікація зображення);
- користувач має змогу оцінити якісь роботи згорткової нейронної мережі (оцінка коректності класифікації).

Отже, було розглянуто підсистеми, які формують інформаційну систему класифікації зображень документів, яка використовує розроблений метод для класифікації зображень документів що посвідчують особу.

2.6 Даталогічна модель бази даних інформаційної системи класифікації зображень документів

Розробка коректної структури бази даних є досить важливим аспектом при розробці інформаційної системи, яка використовує метод для класифікації зображень за допомогою згорткової нейронної мережі.

Для розробки інформаційної системи яка буде включати метод нейромережевої класифікації зображень була розроблена наступна даталогічна модель (Рисунок 2.15).

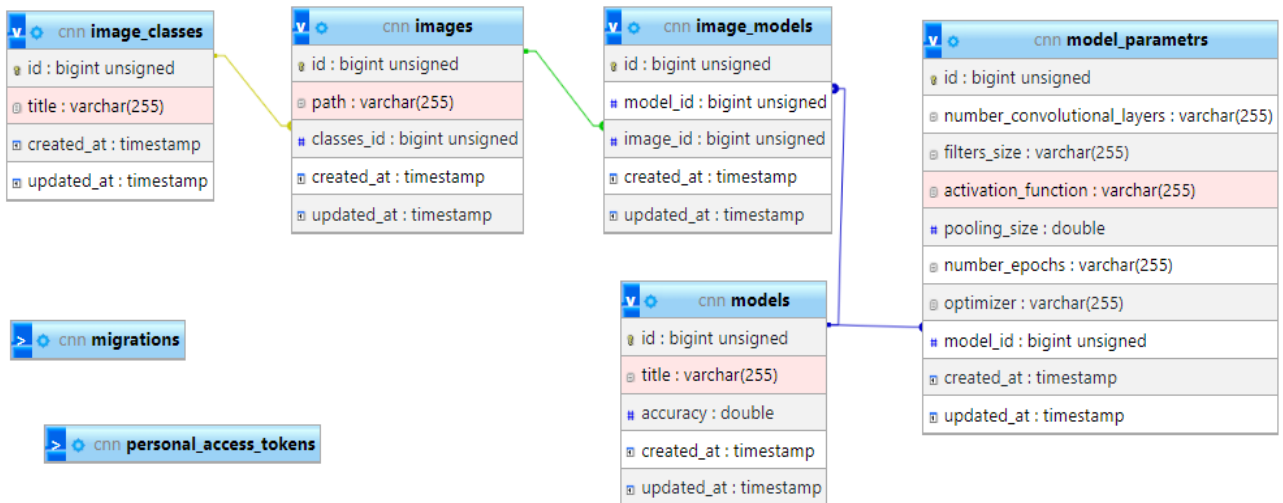


Рисунок 2.15 – Датологічна модель бази даних інформаційної системи

Датологічна модель включає наступні таблиці: images, image_clases.

Для того щоб зберігати зображення, була створена таблиця «images» (Таблиця 2.1).

Таблиця 2.1 – Атрибути таблиці «images»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Унікальний ключ
2.	Title	varchar(255)	Заголовок зображення
3.	Path	varchar(255)	Шлях до зображення
4.	FK_Clases	int	Клас зображення
5.	created_at	timestamp	Дата створення зображення
6.	updated_at	timestamp	Дата оновлення зображення

Щоб визначити клас зображення (паспорт, водійське посвідчення тощо) була створена таблиця «image_clases» (Таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «image_clases»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Унікальний ключ
2.	Title	varchar(255)	Назва класу
3.	created_at	timestamp	Дата створення класу
4.	updated_at	timestamp	Дата оновлення класу

Для того щоб зберегти вибірку зображень, яка використана для навчання нейромережі була створена таблиця «image_models» (Таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «image_models»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Унікальний ключ
2.	model_id	int	Посилання на модель
3.	image_id	int	Посилання на зображення

Для того щоб користувач зміг зберегти модель в базі даних була створена таблиця «models» (Таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «models»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Унікальний ключ
2.	title	varchar(255)	Назва моделі
3.	accuracy	double	Точність класифікації

Для того щоб користувач зміг гучно налаштувати параметри моделі була створена таблиця «model_parametrs» (Таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці «model-parametrs»

№ п/п	Назва	Тип даних	Опис
1.	ID	int	Унікальний ключ
2.	number_convolutional_layers	int	Кількість згорткових шарів
3.	filters_size	varchar(255)	Розмірність фільтрів
4.	activation_function	varchar(255)	Функція активації
5.	pooling_size	varchar(255)	Розмір пулінгу
6.	optimizer	varchar(255)	Функція оптимізації

Дана даталогічна модель БД інформаційної системи класифікації зображень документів дозволить зручно маніпулювати вибірками зображень. Також завдяки даній даталогічній моделі, користувач зможе досить гнучко налаштувати параметри методу для класифікації зображень документів що посвідчують особу.

2.7 Висновки до розділу 2

За допомогою методу для нейромережевої класифікації зображень документів можна автоматизувати процес класифікації зображень на певні категорії або класи. Для цього необхідно зібрати датасет зображень з об'єктами, які будуть класифікуватись, створити нейронну мережу, спроектувати її архітектуру в залежності від задачі, активізувати нейронну мережу на навчання та тестування її.

Після успішного тестування нейромережу можна використовувати для класифікації зображень, якщо тести не пройдено – необхідно доопрацювати мережу.

В якості нейромереджжі було використано TensorFlow.js, бібліотеку для машинного навчання, яка дозволяє реалізувати згорткову нейронну мережу з використанням кількох згорткових та пулінгових шарів, а також повнозв'язного

та вихідного шарів для класифікації зображень. Було спроектовано архітектуру де перші два згорткових шари використовуються для виявлення конкретних ознак на зображенні, а шари пулінгу – для зменшення розміру зображення та кількості параметрів моделі.

Повнозв'язний шар об'єднує всі ознаки та здійснює класифікацію, а вихідний шар визначає кількість класів класифікації та використовує функцію активації, щоб визначити ймовірність приналежності до кожного класу.

Також у даному розділі було спроектовано базу даних інформаційної системи класифікації зображень документів, що включає метод для класифікації зображень документів що посвідчують особу.

Отже, в даному розділі було спроектовано архітектуру нейронної мережі та описано метод нейромережевої класифікації документів що посвідчують особу.

Розділ 3 Програмна реалізація інформаційної системи класифікації зображень документів

3.1 Вибір засобів розробки інформаційної системи класифікації зображень документів

На даний момент існує велика кількість засобів розробки інформаційних систем, проте не всі вони підходять для розробки згорткових нейронних мереж.

Перше, що користувач бачить після того як відкриє програму це – інтерфейс. Для розробки структури інтерфейсу було обрано мову розмітки HTML. HTML – це мова розмітки, що використовується для створення та організації веб-сторінок. HTML використовує теги для визначення структури та вмісту сторінки [25].

Для того щоб розробити приємний інтерфейс, тобто надання стилів для тегів було обрано каскадну таблицю стилів – CSS. CSS – це мова стилів, яка використовується для оформлення та стилізації веб-сторінок, створених з використанням HTML. CSS дозволяє змінювати зовнішній вигляд елементів сторінки, таких як кольори, шрифти, розміри, позиціонування та інші аспекти.

Для того щоб розробити метод нейромережевої класифікації документів була обрана бібліотека TensorFlow.js, яка працює із мовою програмування JavaScript. JavaScript – це програмна мова, що використовується для розробки веб-додатків та динамічного взаємодії з користувачем на веб-сторінках. Вона дозволяє створювати інтерактивні елементи, обробляти події, виконувати асинхронні запити до сервера та маніпулювати вмістом сторінки.

Для серверної частини програмного продукту було прийняте рішення використовувати фреймворк Laravel. Laravel – це веб-фреймворк з відкритим кодом для розробки веб-додатків на мові PHP [26]. Він надає зручні та потужні інструменти для швидкої розробки веб-додатків, включаючи готові компоненти, систему маршрутизації, управління базами даних, аутентифікацію та багато іншого. Laravel пропонує елегантну синтаксис і спрощує процес розробки, допомагаючи програмістам писати чистий, зрозумілий та підтримуваний код.

Він є одним з найпопулярніших фреймворків PHP і забезпечує високу продуктивність та масштабованість для розробки веб-додатків різних розмірів та складності.

Такий набір технологій дозволить розробити згорткову нейронну мережу, а також розробити приємний та інтуїтивно зрозумілий для нового користувача інтерфейс інформаційної системи класифікації зображень документів.

3.2 Структура модулів інформаційної системи класифікації зображень документів та їх взаємозв'язок

Оскільки Laravel, використовує MVC патерн, що дозволить розробити зручну структуру проекту, що в майбутньому дозволить уникнути проблем із масштабованістю інформаційної системи класифікації зображень документів.

Для коректної роботи згорткової нейронної мережі інформаційної системи класифікації зображень документів були створені наступні класи:

- CropAndResizeImage – клас, який обрізає зображення до квадратної форми, та змінює його розмір;
- getClasses – клас який визначає функцію getClasses, яка приймає масив рядків classes як вхідний параметр і повертає об'єкт, який містить відповідність між мітками і їхніми індексами;
- getDefaultDownloadHandler – клас, який визначає функції, пов'язані із генерацію рядків для завантаження файлів залежно від переданих класів;
- loadPretrainedModel – клас, який використовує моделі TensorFlow.js для завантаження попередньо навчених моделей і повертає модель, готову до використання;
- prepareData – клас, який включає декілька методів, пов'язаних із обробкою даних та міток за допомогою TensorFlow.js;
- train – код в цьому класі включає метод train, який використовує TensorFlow.js для тренування моделі машинного навчання. Основні кроки тренування включають визначення моделі, компіляцію моделі з оптимізатором

та втратою, а також тренування моделі з використанням набору даних `xs` і міток `ys`;

- `translateImages` – клас, який включає в себе декілька методів пов'язаних із завантаженням та перетворенням зображення для використання у моделі машинного навчання за допомогою `TensorFlow.js`;

- `types` – цей клас визначає набір інтерфейсів і типів даних, які використовуються в програмі для роботи з даними, моделями та параметрами.

Отже, в даному підрозділі описані класи, які використовує інформаційної системи класифікації зображень документів з використанням методу класифікування зображень документів що посвідчують особу. Даний набір класів реалізує важливі моменти інформаційної системи класифікації зображень документів, зокрема робота із розмірністю зображень, завантаженням зображень тощо.

3.3 Особливості реалізації інформаційної системи

На початку розробки інформаційної системи класифікації зображень документів потрібно визначити клас `MLClassifier`, який є обгорткою навколо `TensorFlow.js`, та надає простий інтерфейс для тренування, прогнозування та оцінки моделі класифікації.

- `PretrainedModel` – модель, яка використовується як попередньо навчена модель, для класифікації зображень. Ця модель завантажується за допомогою функції «`loadPretrainedModel`» та зберігається в «`pretrainedModel`» для подальшого використання.

- `Model` – модель, яка буде тренуватися на власних даних. По замовчуванню це порожня модель, але після тренування буде містити навчені ваги та архітектуру.

- `Callbacks` – це масив функцій зворотнього виклику, які будуть виконані після завантаження попередньо навченої моделі. Вони додаються до масиву `callbacks`, а потім виконуються в методі `init`.

– `Args` – це об'єкт, що містить аргументи, передані при створенні екземпляра `MLClassifier`. Він зберігається в `args` для подальшого використання.

– `Data` – це об'єкт, що містить дані, необхідні для тренування та оцінки моделі. Початково він містить порожній об'єкт `classes`, а пізніше заповнюється даними тренування та оцінки у вигляді тензорів `xs` та `ys`.

Важливою частиною реалізації згорткової нейронної мережі інформаційної системи класифікації зображень документів, є створення функціоналу для роботи із зображеннями. Для реалізації цього функціоналу була написана функція «`cropAndResizeImage`».

`CropAndResizeImage`, отримує триразовий тензор `img` (зображення) та розміри `dims` у вигляді масиву `[number, number]`. Вона повертає обрізане та змінене за розміром зображення.

– Використовуючи `tf.tidy`, створюється блок, щоб забезпечити чистоту ресурсів TensorFlow під час обчислень.

– Використовуючи `tf.image.resizeBilinear`, зображення змінюється за розміром `dims` з методом білінійної інтерполяції. Зображення обрізається за допомогою функції `crop`, описаної вище.

– Зображення розширюється по першому розміру за допомогою `expandDims`, щоб отримати один батч зображень.

– Значення пікселів перетворюються на числа з плаваючою точкою та нормалізуються. Спочатку вони діляться на 127, а потім віднімаються 1 за допомогою `tf.scalar`. Повертається обрізане, змінене за розміром та нормалізоване зображення.

– Функція `cropAndResizeImage` експортується як основна функція модуля, що дозволяє її використовувати у інших частинах програми.

Коли викликається функція `cropAndResizeImage` з тензором зображення та розмірами, вона обрізає зображення до квадратної форми, змінює його розмір до заданих розмірів `dims`, нормалізує значення пікселів та повертає оброблене зображення у формі тензору.

Оскільки в інформаційній системі класифікації зображень документів велика кількість роботи відбувається із зображеннями, потрібно реалізувати досить багато функцій. Схема роботи користувача інформаційної системи класифікації зображень документів із датасетом зображена на рисунку 3.2.

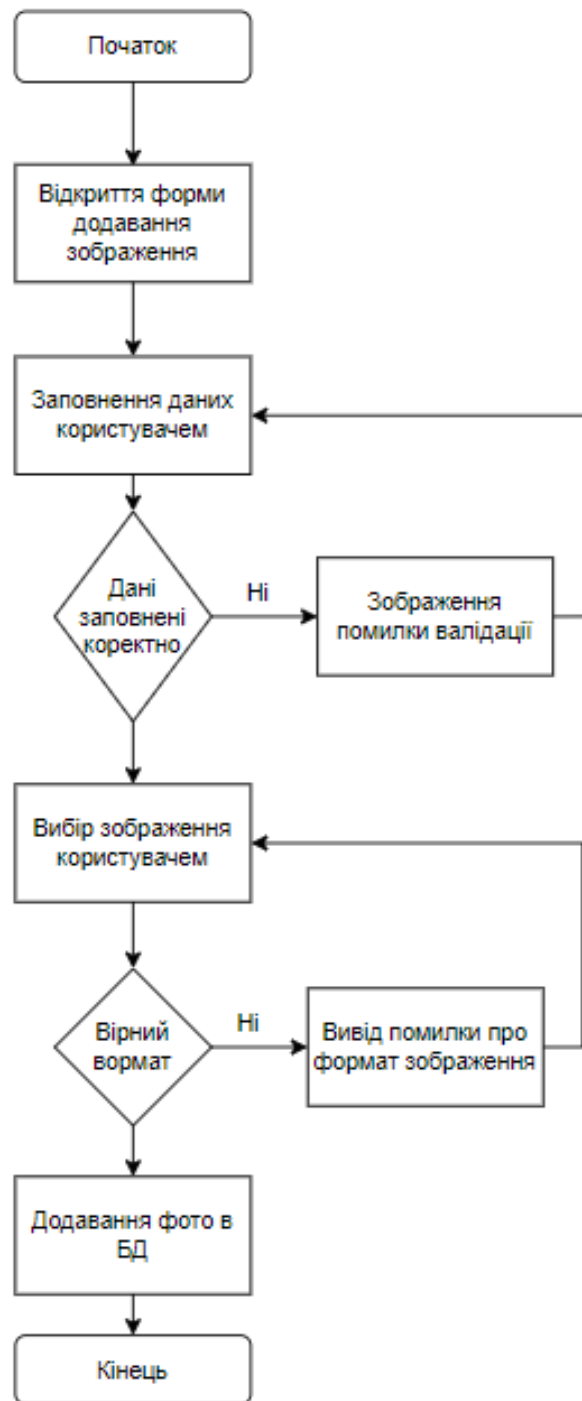


Рисунок 3.1 – Схема роботи користувача із датасетом

Для початку користувачеві інформаційної системи класифікації зображень документів потрібно відкрити форму додавання зображення. Наступним кроком буде заповнення даних, назва фото, клас. Далі потрібно вибрати фото, після чого натиснути кнопку додати. Після цього запусаються функції валідації і якщо все вірно, то фото буде додане в БД.

Оскільки в інформаційній системі передбачено автоматичне тестування, то схема використання цієї функції зображена на рисунку 3.1.

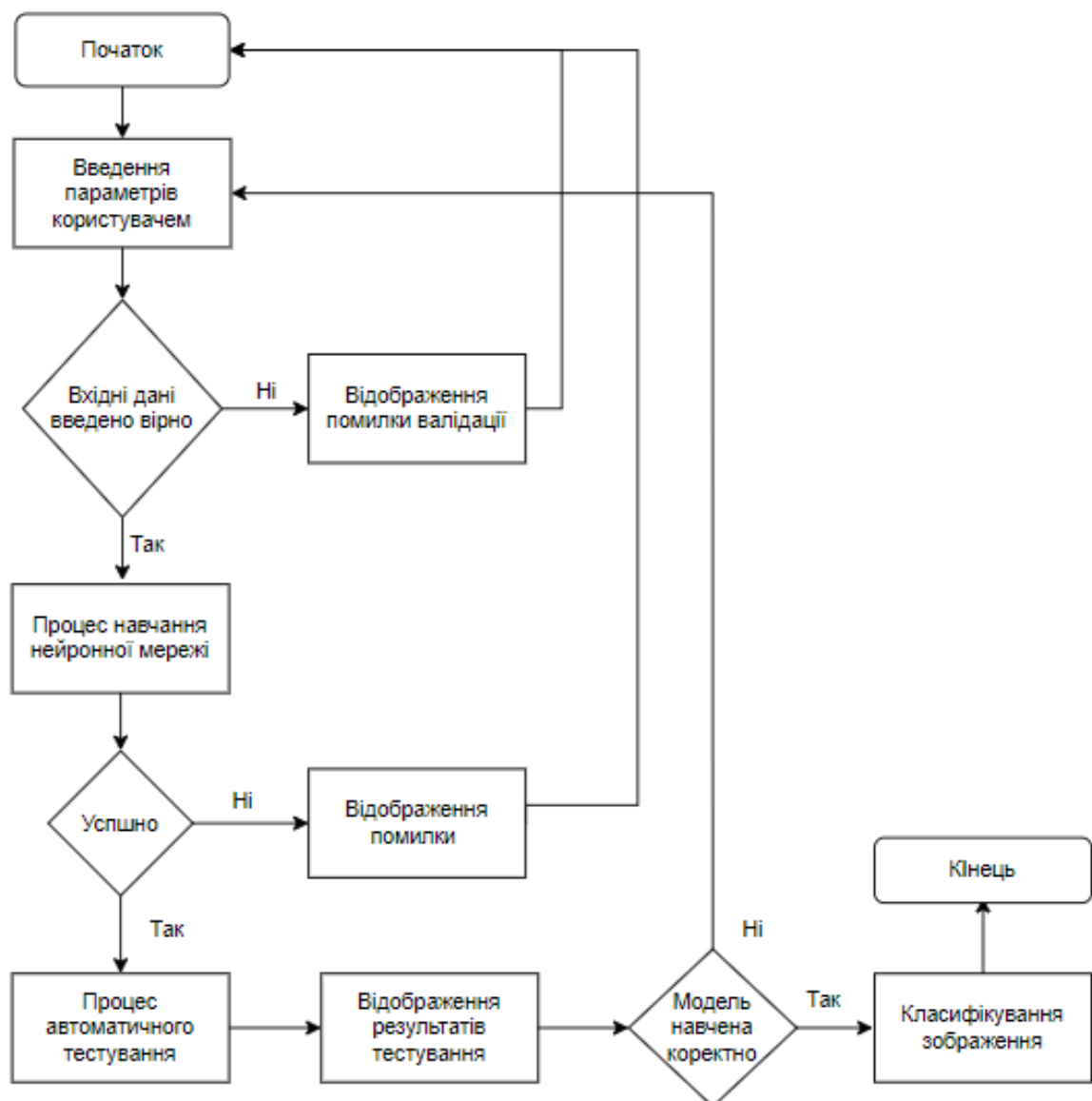


Рисунок 3.2 – Схема роботи методу класифікування зображення

Для початку користувачу потрібно зайти на сторінку класифікатора. Далі на формі налаштування нейронної мережі користувач вибирає кількість фото та класи які навчають нейронну мережу. Якщо дані пройдуть функції валідації мережа автоматично розпочне своє тестування. Після цього дані будуть виведені користувачеві, якщо користувач вирішить що результати тестування незадовільні, він зможе довчити модель. Якщо ж користувача все влаштовує, він може приступати до класифікування фото.

Отже в даному підрозділі розписані базові класи інформаційної системи класифікації зображень документів для роботи із TensorFlow.js. Також було побудовано дві схеми, які зображують додаткові функції інформаційної системи класифікації зображень документів, яка використовує метод для нейромережового класифікування зображень документів що посвідчують особу.

3.4 Опис функціональних можливостей інформаційної системи класифікації зображень документів

3.4.1 Інструкція користувача

Будь-яка інформаційна система має свій інтерфейс, для того щоб користувачу було зручно потрібно розробити приємний на вигляд та зрозумілий інтерфейс.

Після відкриття інформаційної системи класифікації зображень документів, користувач попадає на головну сторінку де розташована навігація по інформаційній системі (Рисунок 3.3)



Рисунок 3.3 – Навігація по інформаційній системі класифікації зображень документів

Натиснувши кнопку «Датасет», користувач інформаційної системи класифікації зображень документів попадає на сторінку з всіма зображеннями датасету. На цій сторінці користувач має змогу редагувати та фільтрувати зображення (Рисунок 3.4).

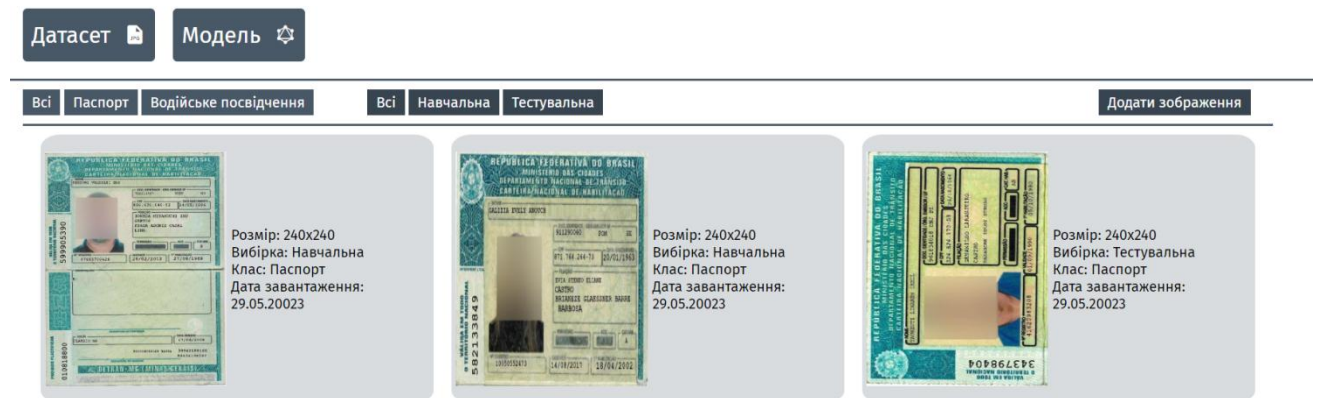


Рисунок 3.4 – Сторінка «датасет» інформаційної системи класифікації зображень документів

Натиснувши кнопку «модель» користувач інформаційної системи класифікації зображень документів попаде на сторінку класифікатора, де йому буде доступно декілька форм (Рисунок 3.5).

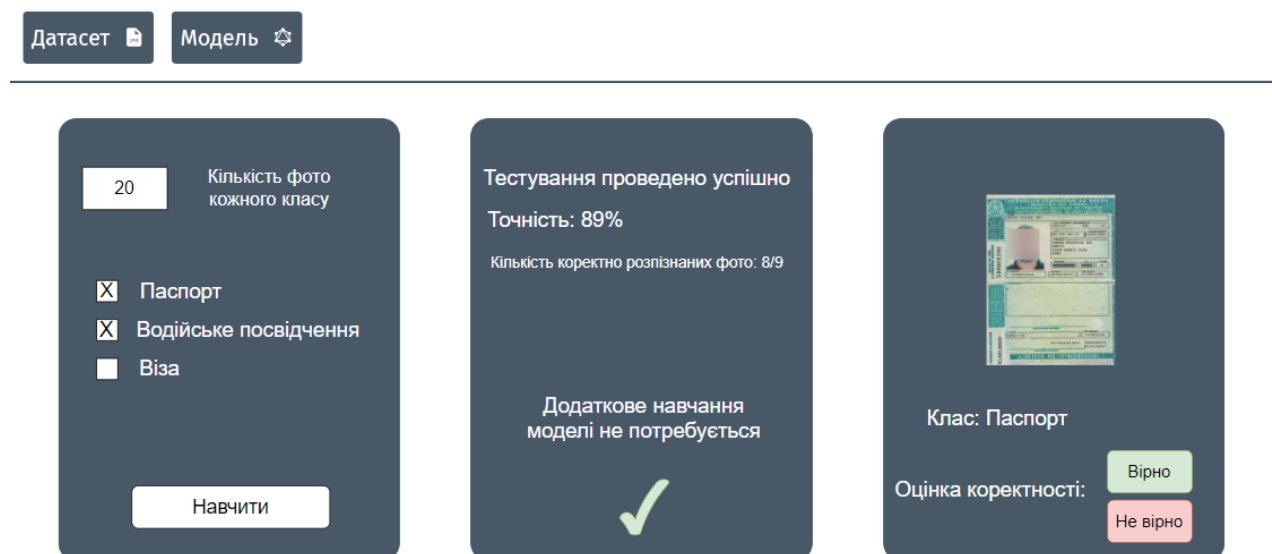


Рисунок 3.5 – Сторінка «Модель» інформаційної системи класифікації зображень документів

На даній сторінці, користувач інформаційної системи класифікації зображень документів має змогу навчити нейронну мережу із певними параметрами, оцінити точність завдяки самотестуванню, та розпізнати зображення документа.

Для того щоб навчити нейронну мережу, користувачу інформаційної системи класифікації зображень документів доступна форма, на якій він може обрати класи, та кількість фото для навчання (Рисунок 3.6).

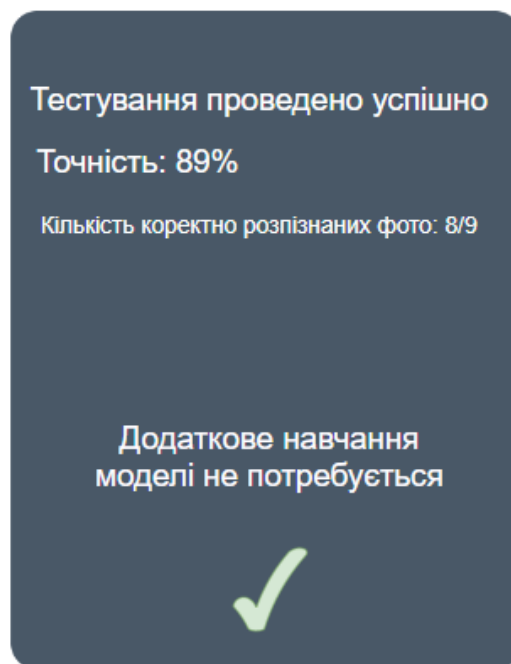


Рисунок 3.6 – Форма для навчання згорткової нейронної мережі

Після навчання нейронної мережі, відбувається автоматичне тестування коректності. Для виводу результатів була створена окрема форма (Рисунок 3.7).

Після класифікації зображення, користувачу інформаційної системи класифікації зображень документів пропонується оцінити чи вірно мережа класифікувала зображення. Якщо користувач натисне кнопку «Вірно» зображення автоматично буде додано до датасету із класифікованим класом.

Якщо користувач натисне «Не вірно» зображення буде класифіковано заново (Рисунок 3.8).

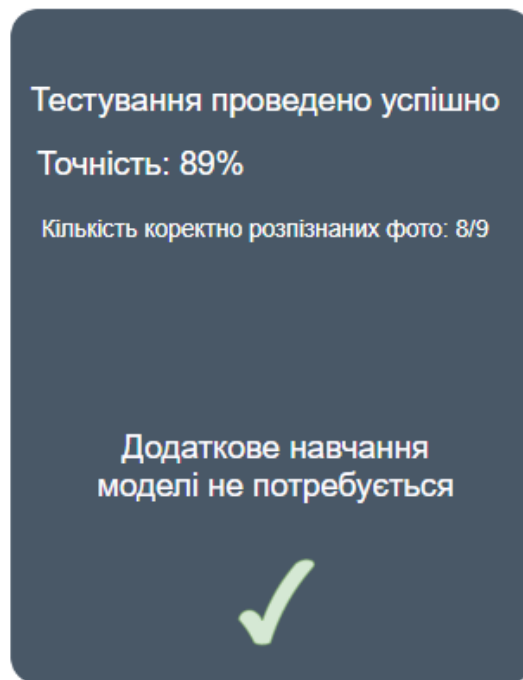


Рисунок 3.7 – Форма інформаційної системи класифікації зображень документів для виводу результатів автоматичного тестування

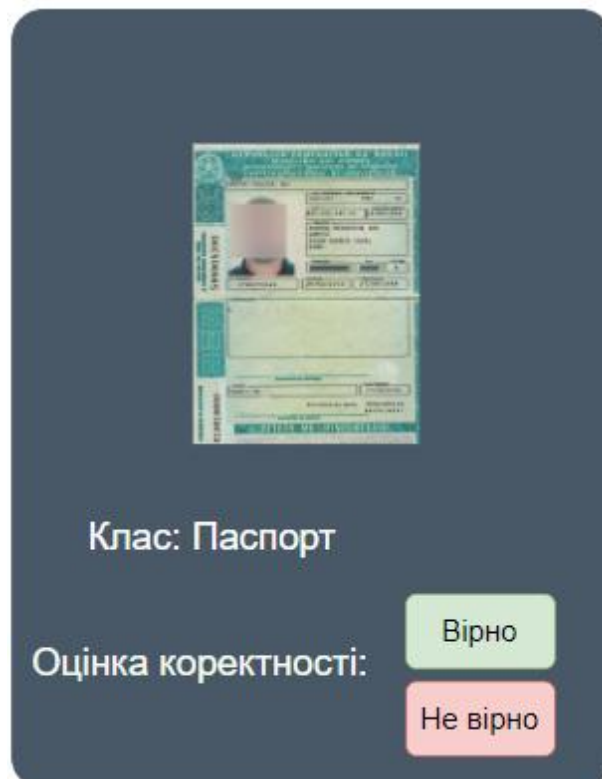


Рисунок 3.8 – Форма інформаційної системи класифікації зображень документів для оцінки коректності класифікування

Отже розроблена інструкція інформаційної системи класифікації зображень документів, яка зображує коротко можливі дії користувача в інформаційній системі.

3.4.2 Тестування інформаційної системи класифікації зображень документів

Тестування в розробці відіграє важливу роль, оскільки дозволяє запобігти появленню неочікуваних помилок в роботі програми. При розробці інформаційної системи класифікації зображень документів необхідно протестувати основні підсистеми, а саме роботи з базою даних та підсистеми класифікації.

Перше, що зможе зробити користувач відкривши інформаційну систему – додати фото до датасету (Рисунок 3.9). Тест-кейс для перевірки даної функції інформаційної системи класифікації зображень документів наведено у таблиці 3.1.

Таблиця 3.1 – Тест-кейс AA001

Тест-кейс ID: AA0001	Пріоритет: 1	Створено: 29.05.2022, Похитун А.В
Назва: Перевірка функції додавання фото до датасету		
Вхідні дані:		
<ol style="list-style-type: none"> 1. Назва: passport 2. Клас: Паспорт 3. Вибірка: Тестувальна 4. Фото паспорта із датасету 		
Передумова: Клас паспорт має існувати в базі даних		
Кроки		Очікуваний результат
<ol style="list-style-type: none"> 1. Відкрити інформаційну систему 2. Натиснути кнопку «Додати фото» 3. Ввести вхідні дані 4. Натиснути кнопку зберегти 		<p>Сторінка перезавантажилась.</p> <p>Фото та дані відобразились в системі</p>

Результат виконання тест-кейсу: пройдено успішно

Датасет

Модель

Всі


Паспорт

Водійське посвідчення


Всі

Навчальна

Тестувальна



Розмір: 240x240
Вибірка: Навчальна
Клас: Водійське посвідчення
Дата завантаження: 23.05.20023



Розмір: 240x240
Вибірка: Тестувальна
Клас: Паспорт
Дата завантаження: 29.05.20023

Рисунок 3.9 – Додавання фото до датасету

Також при роботі із зображеннями, користувачу інформаційної системи класифікації зображень документів доступні різного роду фільтри, що допоможе швидше знайти потрібне зображення (Рисунок 3.10). Тест кейс цієї функції розписано в таблиці 3.2.

Таблиця 3.2 – Тест-кейс AA002

Тест-кейс ID: AA0002	Пріоритет: 2	Створено: 29.05.2022, Похитун А.В
Назва: Перевірка функції сортування датасету		
Вхідні дані: -		
Передумова: Фото із класом «паспорт» має бути додане в базу даних		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> 1. Відкрити інформаційну систему 2. Натиснути кнопку «Датасет» 3. Натиснути кнопку «Паспорт» 	<p>Сторінка перезавантажилась.</p> <p>Фото відсортувались</p>	
Результат виконання тест-кейсу: пройдено успішно		

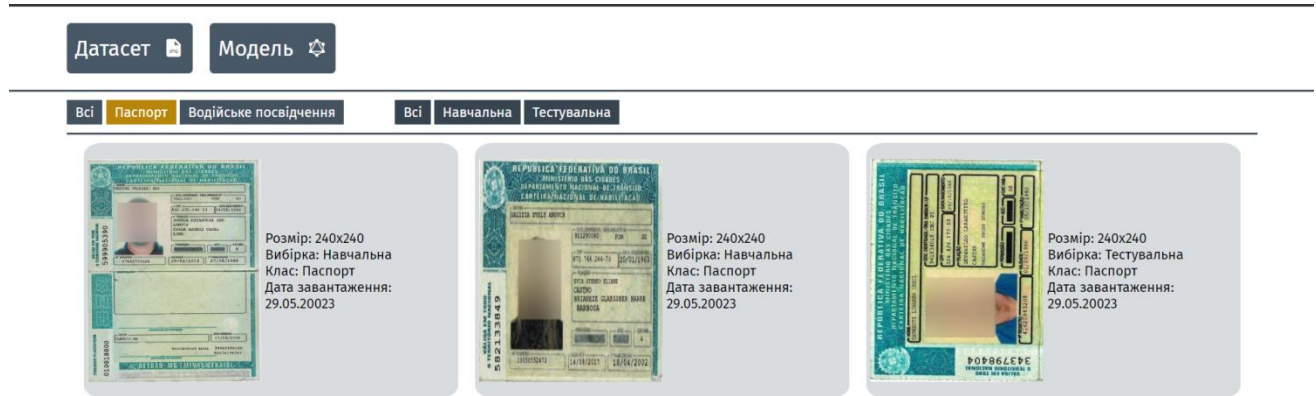


Рисунок 3.10 – Результат сортування датасету

Функція самотестування моделі є досить важливою оскільки вона дає змогу оцінити середню коректність класифікації. Тест-кейс для перевірки даної функції наведено у таблиці 3.3 (Рисунок 3.11).

Таблиця 3.3 – Тест-кейс AA003

Тест-кейс ID: AA0003	Пріоритет: 1	Створено: 29.05.2022, Похитун А.В
Назва: Перевірка функції самотестування		
Кроки	Очікуваний результат	
<ol style="list-style-type: none"> 1. Відкрити інформаційну систему 2. Натиснути кнопку «Модель» 3. Натиснути кнопку «Навчити» 	Вивід результату самотестування	
Результат виконання тест-кейсу: пройдено успішно		

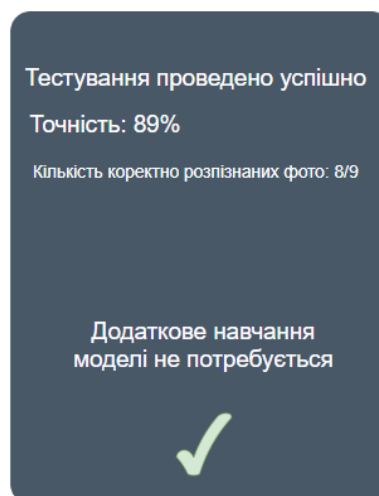


Рисунок 3.11 – Результат самотестування

Отже, тестування інформаційної системи класифікації зображень документів показало, що основний функціонал виконується коректно, помилок не виявлено, а метод нейромережевої класифікації зображень документів що посвідчують особу є практично застосовним.

3.5 Дослідження ефективності застосування методу нейромережевої класифікації зображень документів що посвідчують особу

Під час розробки згорткової нейронної мережі, потрібно врахувати що затрата часу для її навчання може бути досить великою проблемою. Для того щоб оцінити швидкість навчання було створено декілька вибірок зображень, та виконано навчання нейронної мережі на цих зображеннях (Рисунок 3.12).

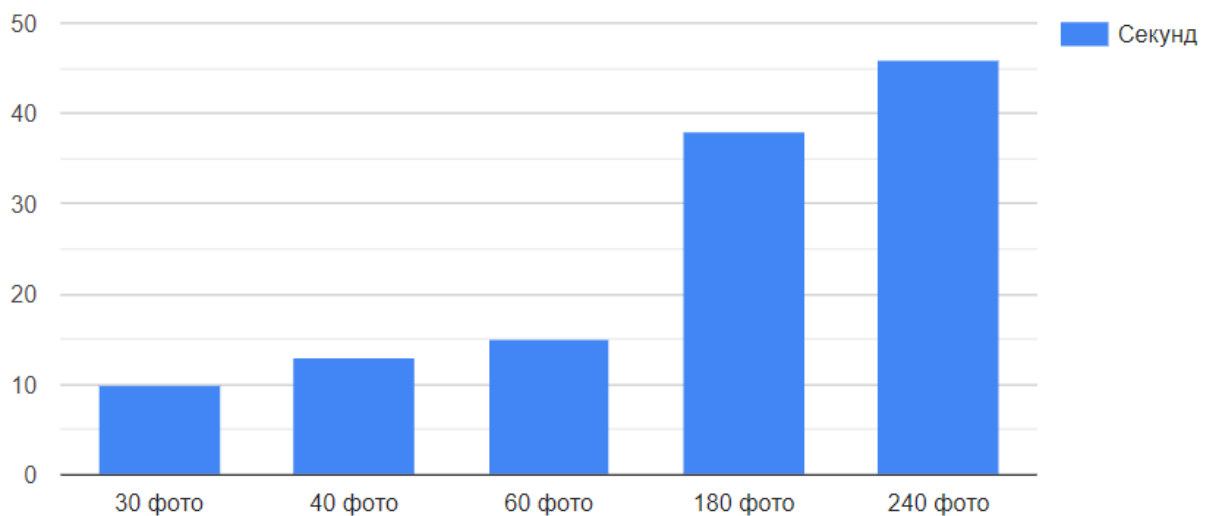


Рисунок 3.12 – Швидкість навчання нейронної мережі інформаційної системи класифікації зображень документів

Варто зазначити, що навчання нейронної мережі відбувалось на персональному комп'ютері з наступними характеристиками:

- Оперативна пам'ять: 8 гб;
- Процесор: Intel Core i5 8th Gen (4 ядра 8 потоків);

– Відеокарта: 1050TI (4 гб відеопам'яті).

Потрібно взяти до уваги, що при навчанні нейронної мережі проходило протягом 20 епох (Епоха позначає один цикл протягом усього набору тренувань). Більш детальніший результат наведено в Таблиці 3.4

Таблиця 3.4 – Результати тестування швидкості навчання нейронної мережі інформаційної системи класифікації зображень документів

Загальна кількість зображень	Швидкість навчання (секунди)
30	10
40	13
60	15
180	38
240	47

Провівши дослідження, можна зробити висновок, що нейронна мережа обробляє одне фото з швидкістю $\approx 3-4$ секунди, проте на більшій кількості зображень швидкість навчання знижується до $\approx 5-6$ секунд.

Для того, щоб визначити приблизну оцінку коректності було проведено дослідження із різною розмірністю датасетів (Рисунок 3.13).

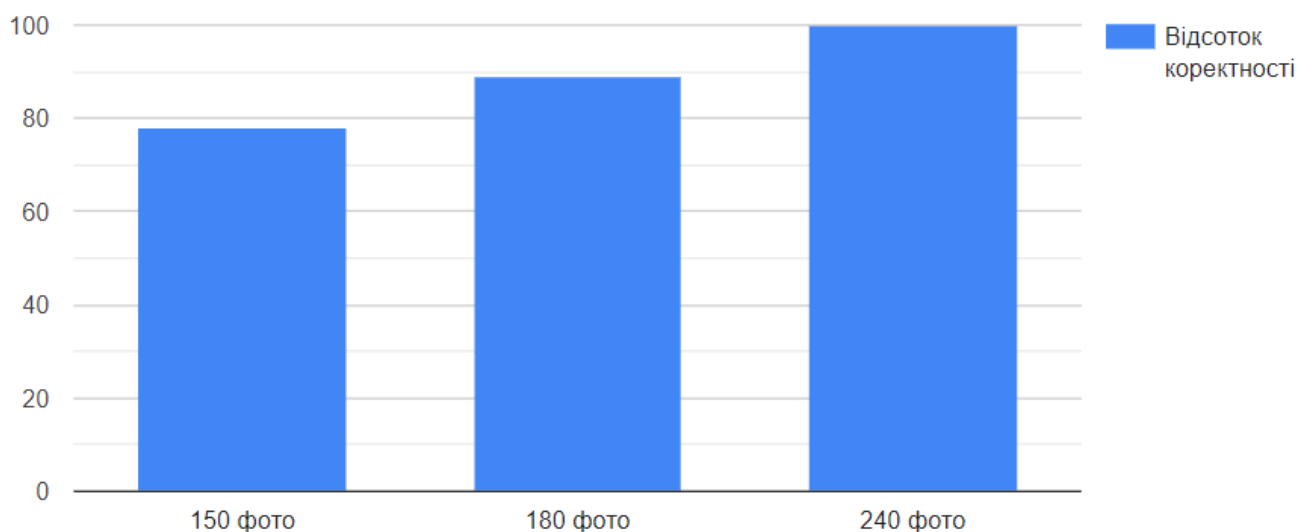


Рисунок 3.13 – Оцінка коректності

Більш детальний результат зображено в Таблиці 3.5.

Таблиця 3.5 – Оцінка коректності

Кількість фото	Коректність (%)	Коректність(відношення)
150	78%	6/9
180	89%	8/9
240	100%	9/9

Отже, дослідження показало інформаційної системи класифікації зображень документів, що кількість зображень напряму впливає на швидкість навчання одного фото згорткової нейронної мережі. Також було проведене дослідження яке показало, що для коректної оцінки зображення документа, нейронну мережу потрібно навчати достатньою великою кількістю зображень, а саме не менше 60 на кожен клас (в даному випадку).

3.6 Висновки до розділу 3

В цілому, в даному розділі було проаналізовано існуючі технології та метод розробки інформаційних систем, обрано декілька мов програмування які дозволяють ефективно розробляти подібні інформаційні системи.

В даному розділі було обрані засоби розробки, які допоможуть ефективно розробити прикладне застосування методу для нейромережевої класифікації зображень документів що посвідчують особу в вигляді інформаційної системи класифікації зображень документів.

Також було розписано структуру, класи інформаційної системи, яка використовує метод класифікації зображень що посвідчують особу.

Базовий функціонал інформаційної системи класифікації зображень документів був протестований та налагоджений.

Даний розділ містить інформацію про програмну структуру інформаційної системи класифікації зображень документів.

Було проведено декілька досліджень, які допомогли визначити ефективність та швидкість навчання згорткової нейронної мережі інформаційної системи класифікації зображень документів, для дослідження ефективності методу нейромережевої класифікації зображень документів що посвідчують особу.

Висновки

Кваліфікаційна робота бакалавра присвячена розробці метода нейромережевої класифікації зображень документів що посвідчують особу та відповідної інформаційної системи, яка дозволяє за зображеннями документа визначати його вид за допомогою згорткової нейромережі.

В ході виконання кваліфікаційної роботи бакалавра був проведений аналіз предметної області класифікації зображень документів. Визначено основні типи документів які посвідчують особу для класифікації.

Було знайдено та досліджено існуюче подібне програмне забезпечення, виявлено переваги та недоліки кожного із них.

Окремим результатом виконання кваліфікаційної роботи бакалавра є розроблена інформаційна система яка використовує метод нейромережевої класифікації документів. Для створення інформаційної системи класифікації зображень документів використовувалась мова програмування JavaScript із бібліотекою TensorFlow. Дизайн інформаційної системи класифікації зображень документів розроблений на HTML та CSS із шаблонізатором Blade який є вбудованим в Laravel. Для інтерактивності на сторінках використовувалась ще одна бібліотека – JQuery.

Також в ході кваліфікаційної роботи було обрано датасет з зображеннями закордонних документами що посвідчують особу для їх використання в роботі інформаційної системи класифікації зображень документів.

Окремою частиною роботи стало автоматичне тестування навченої моделі нейромережі. Це допомагає користувачу інформаційної системи класифікації зображень документів зрозуміти наскільки коректно ця модель зможе класифікувати зображення документа що посвідчує особу.

Проведене дослідження ефективності застосування методу нейромережевої класифікації зображень документів що посвідчують особу показало, що на швидкість навчання впливає кількість зображень в датасеті.

Встановлено, що для коректної оцінки зображення документа, нейронну мережу потрібно навчати не менше ніж 60-ма зображеннями на кожен клас.

Інформаційну систему класифікації зображень документів можна масштабувати та покращувати в майбутньому, наприклад наступним:

- розробка системи ролей та користувачів;
- більш гнучкіше налаштування моделі;
- розробка адаптивного інтерфейсу;
- можливість скачати лістинг навченої моделі.

Проект створений в дослідницьких цілях, тому конкурентоспроможності вже готовим комерційним програмним продуктам не складає, хоча розроблений метод нейромережевої класифікації зображень документів що посвідчують особу в подальшому може бути використаний у прикладних програмних системах.

Перелік посилань

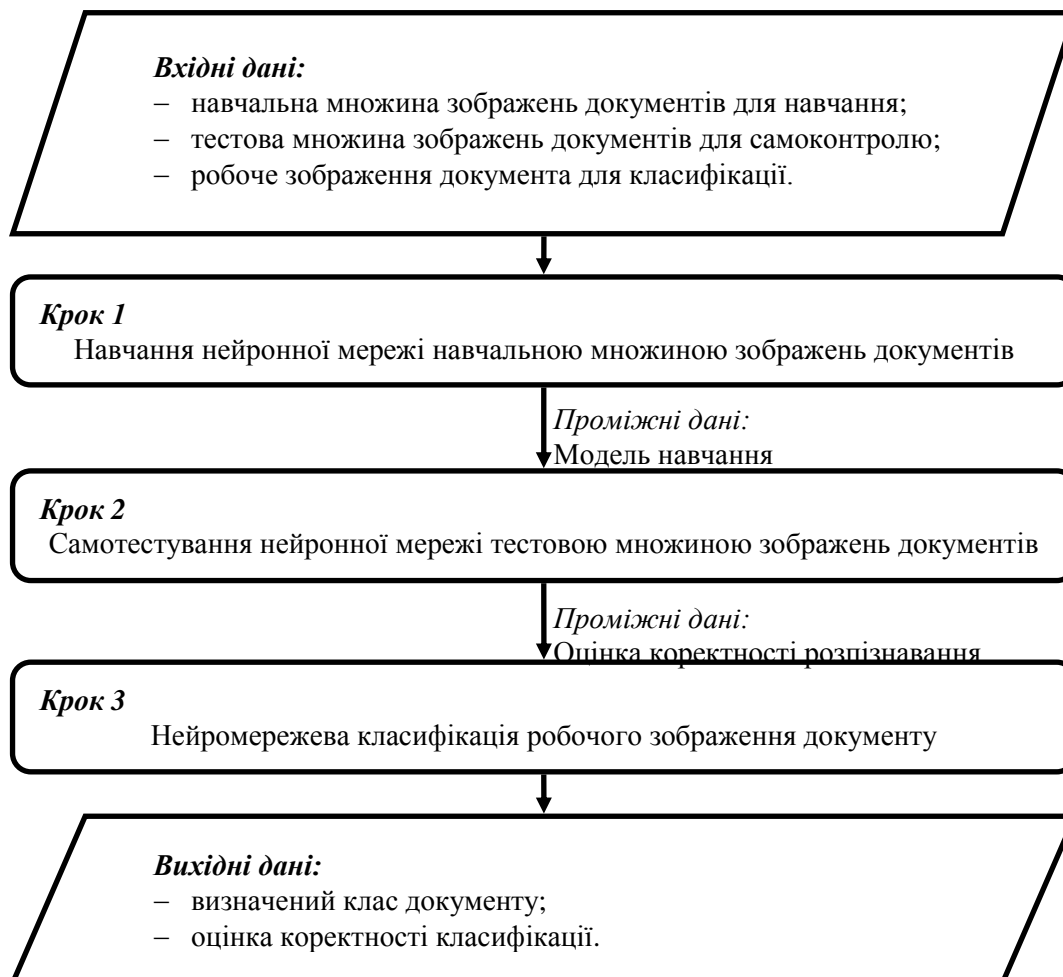
1. Conf. Методи обробки зображень. URL: <https://conf.ztu.edu.ua/wp-content/uploads/2019/12/74-1.pdf>
2. Code. Класифікація зображень за допомогою ЗНМ. URL: <https://codeguida.com/post/1400>
3. Livingfo. Що таке нейронні мережі, та як вони працюють. URL: <https://livingfo.com/shcho-take-nejronni-merezhi-ta-iak-vony-pratsiuiut/>.
4. Dmsu. Паспорт громадянина України. URL: <https://dmsu.gov.ua/poslugi/pasport-gromadyanina-ukrajni.html>.
5. Guide. Паспорт громадянина України для виїзду за кордон. URL: <https://guide.dii.gov.ua/view/oformlennia-i-vydacha-pasporta-hromadianyna-ukrainy-dlia-vyizdu-za-kordon-z-bezkontaktnym-elektronnym-nosiiem-2c06c141-616d-4e09-8661-a293b6b0ee25>.
6. Ligazakon. Дипломатичний паспорт громадянина України. URL: https://ips.ligazakon.net/document/U515_92
7. Zakon.rada. Службовий паспорт громадянина України. URL: <https://zakon.rada.gov.ua/laws/show/264/2015#Text>
8. Kmu. Тимчасове посвідчення громадянина України. URL: <https://www.k-mu.gov.ua/npas/1863207>
9. Legalclinics. Перелік документів, що надають особливий статус громадянину. URL: <https://legalclinics.in.ua/consult/consultation-26-06-2020-5/>
10. Арепс. Як працюють нейронні мережі. URL: <http://areps.kpi.ua/neural-networks/en>
11. Codeguida. Нейронні мережі, шлях до глибинного навчання. URL: <https://codeguida.com/post/739>.
12. Evergreens. Згортоква нейронна мережі, застосування. URL: <https://evergreens.com.ua/ua/articles/cnn.html>.

13. Openarchive. Згорткові нейронні мережі як основ машинного навчання. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/a112beea-43f7-42d3-b63b-b7b40c46324e/content>.
14. Engati. Convolutional neural network. URL: <https://www.engati.com/glossary/convolutional-neural-network>.
15. Aspiringyouths. Disadvantages CNN. URL: <https://aspiringyouths.com/advantages-disadvantages/convolutional-neural-network-cnn/>.
16. Amazon. Amazon Textextract. URL: <https://aws.amazon.com/ru/textract/features/>.
17. OpenCV. URL: <https://opencv.org/>.
18. Kofax. URL: <https://www.kofax.com/>.
19. Towardsdatascience. Understanding and calculating the number of parameters in CNN: URL: <https://towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d>
20. Sciencedirect. Optimization CNNs. URL: <https://www.sciencedirect.com/science/article/pii/S1319157820304845>.
21. C3. Classifier in Cnns. URL: <https://c3.ai/glossary/data-science/classifier/>.
22. TensorFlow. URL: <https://www.tensorflow.org/?hl=ua>.
23. Analyticsvidhya. Tuning the hyperparameters in CNN. URL: <https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/>.
24. Openarchive. Алгоритм підвищення ефективності згорткових нейронних мереж. URL: <https://openarchive.nure.ua/items/5b3a92e2-c8c4-45a9-9e01-9e1d72850d41>.
25. W3schoolsua. Підручник HTML. URL: <https://w3schoolsua.github.io/html/index.htm-l#gsc.tab=0>.
26. Laravel, офіційний сайт. URL: <https://laravel.com/>.

ДОДАТКИ

Додаток А

Схема методу нейромережевої класифікації зображень документів що посвідчують особу

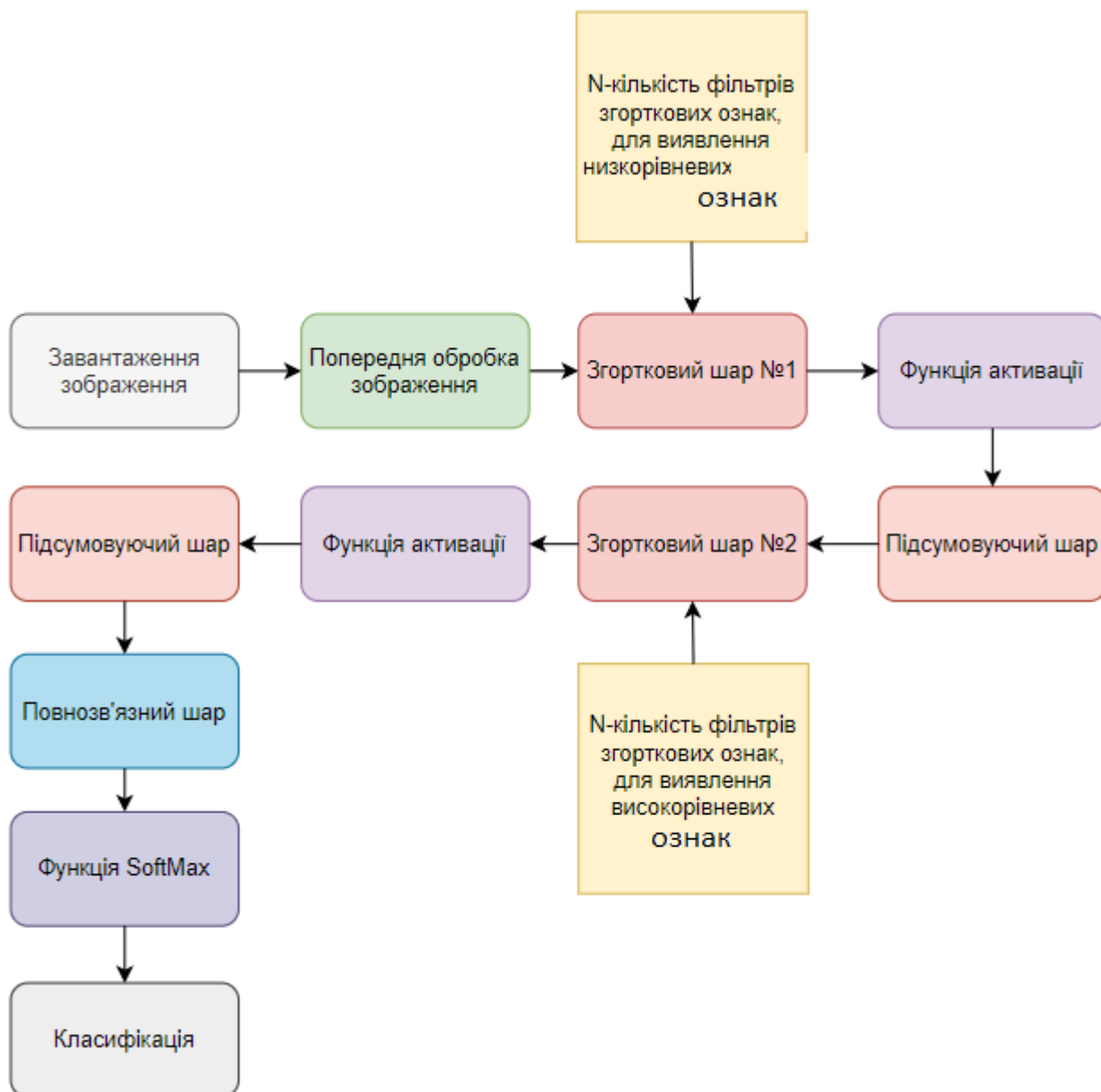


Додаток Б
Схема навчання класифікатора



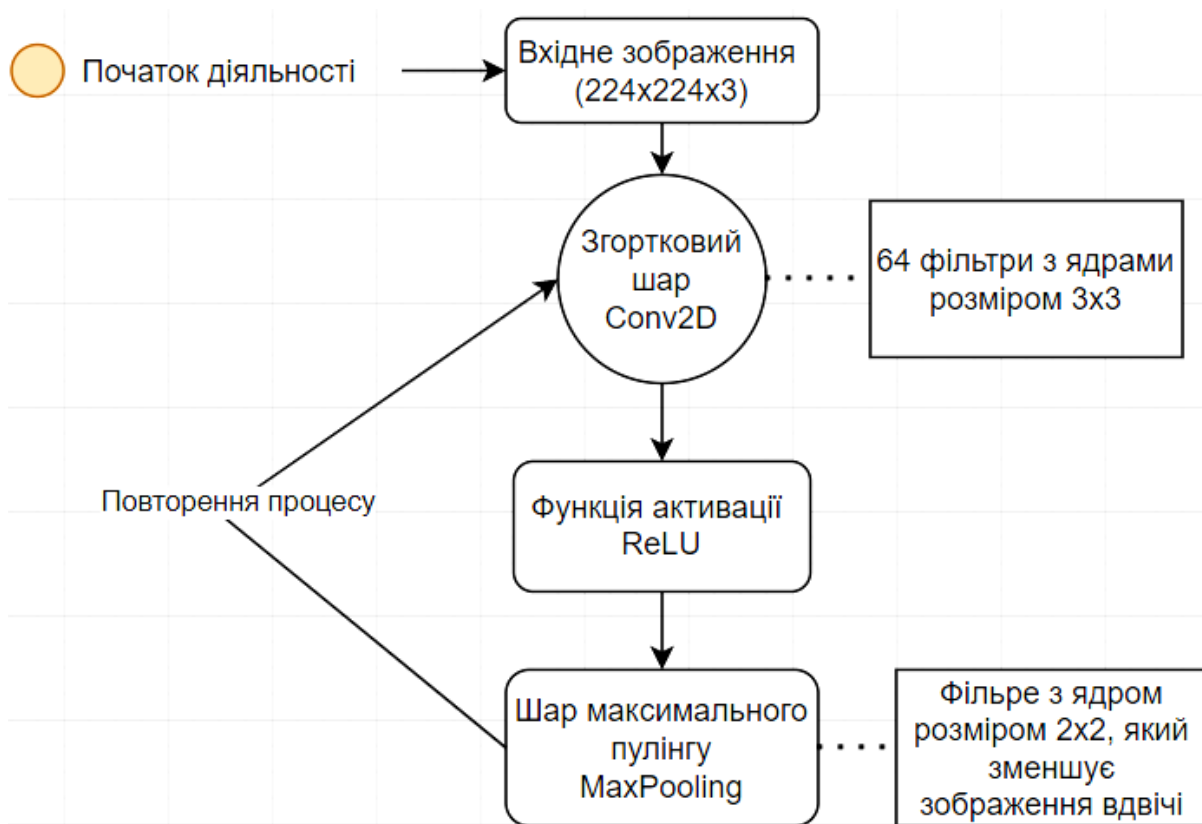
Додаток В

Діаграма послідовності операцій згорткової нейронної мережі за класифікування зображень



Додаток Г

Діаграма активності згорткової нейронної мережі



Додаток Д

Програмний код

Лістинг файлу cropAndResizeImage.js

```
import * as tf from '@tensorflow/tfjs';

const crop = (img: tf.Tensor3D) => {
  const size = Math.min(img.shape[0], img.shape[1]);
  const centerHeight = img.shape[0] / 2;
  const beginHeight = centerHeight - (size / 2);
  const centerWidth = img.shape[1] / 2;
  const beginWidth = centerWidth - (size / 2);
  return img.slice([beginHeight, beginWidth, 0], [size, size, 3]);
}

// convert pixel data into a tensor
const cropAndResizeImage = async (img: tf.Tensor3D, dims: [number, number]):
Promise<tf.Tensor3D> => {
  return tf.tidy(() => {
    const croppedImage = crop(tf.image.resizeBilinear(img, dims));
    return
croppedImage.expandDims(0).toFloat().div(tf.scalar(127)).sub(tf.scalar(1));
  });
};

export default cropAndResizeImage;
```

Лістинг файлу prepareData.js

```
import * as tf from '@tensorflow/tfjs';

import {
  IClasses,
} from './types';

const oneHot = (labelIndex: number, classLength: number) => tf.tidy(() =>
tf.oneHot(tf.tensor1d([labelIndex]).toInt(), classLength));

// const turnTensorArrayIntoTensor = (tensors: tf.Tensor[]) =>
tensors.reduce((data?: tf.Tensor, tensor: tf.Tensor) => tf.tidy(() => {
//   if (data === undefined) {
//     return tf.keep(tensor);
//   }

//   const newData = tf.keep(data.concat(tensor, 0));
//   data.dispose();
//   return newData;
// }), undefined);

export const addData = (tensors: tf.Tensor[]): tf.Tensor => {
  const data = tf.keep(tensors[0]);
  return tensors.slice(1).reduce((data: tf.Tensor, tensor: tf.Tensor) =>
tf.tidy(() => {
    const newData = tf.keep(data.concat(tensor, 0));
    data.dispose();
    return newData;
  }), data);
};
```

```

};

export const addLabels = (labels: string[], classes: IClasses): tf.Tensor2D |
undefined => {
  const classLength = Object.keys(classes).length;
  if (classLength <= 1) {
    throw new Error('You must provide more than 1 class for training');
  }

  return labels.reduce((data: tf.Tensor2D | undefined, label: string) => {
    const labelIndex = classes[label];
    const y = oneHot(labelIndex, classLength);

    return tf.tidy(() => {
      if (data === undefined) {
        return tf.keep(y);
      }

      const old = data;
      const ys = tf.keep(old.concat(y, 0));

      old.dispose();
      y.dispose();

      return ys;
    });
  }, undefined);
};

```

Лістинг файлу index.js

```

import * as tf from '@tensorflow/tfjs';
import cropAndResizeImage from './cropAndResizeImage';
import getClasses from './getClasses';
import train from './train';
import translateImages, {
  IImageData,
} from './translateImages';
import loadPretrainedModel from './loadPretrainedModel';
// import log, { resetLog } from './log';
import {
  addData,
  addLabels,
} from './prepareData';
import getDefaultDownloadHandler from './getDefaultDownloadHandler';

import {
  IParams,
  // IConfigurationParams,
  IData,
  ICollectedData,
  IArgs,
  // DataType,
} from './types';

// export { DataType } from './types';

class MLClassifier {
  // private pretrainedModel: typeof tf.model;
  // private pretrainedModel: any;
  private pretrainedModel: tf.Model;
  // private model: tf.Sequential;
  private model: any;

```

```

private callbacks: Function[] = [];
private args: IArgs;
private data: IData = {
  classes: {},
};

constructor(args: IArgs) {
  this.args = args;
  this.init();
}

private callbackFn = (fn: string, type: string, ...args: any[]) => {
  const key = `${fn}${type.substring(0, 1).toUpperCase()}${type.substring(1)}`;
  if (this.args[key] && typeof this.args[key] === 'function') {
    this.args[key](...args);
  }
}

private init = async () => {
  this.callbackFn('onLoad', 'start');
  this.pretrainedModel = await loadPretrainedModel(this.args.pretrainedModel);

  this.callbacks.map(callback => callback());

  this.callbackFn('onLoad', 'complete');

  // Warmup the model
  const dims = await this.getInputDims();
  tf.tidy(() => {
    this.pretrainedModel.predict(tf.zeros([1, ...dims, 3]));
  });
}

private loaded = async () => new Promise(resolve => {
  if (this.pretrainedModel) {
    return resolve();
  }

  this.callbacks.push(() => {
    resolve();
  });
});

// private cropAndActivateImage = async (image: tf.Tensor3D) => {
private cropAndActivateImage = async (image: any) => {
  await this.loaded();
  // const {
  //   inputLayers,
  // } = this.pretrainedModel;
  // const {
  //   batchInputShape,
  // } = inputLayers[0];
  const dims = await this.getInputDims();
  await tf.nextFrame();
  const processedImage = await cropAndResizeImage(image, dims);
  await tf.nextFrame();
  const pred = this.pretrainedModel.predict(processedImage);
  return pred;
}

private getInputDims = async (): Promise<[number, number]> => {
  await this.loaded();

  const {
    inputLayers,

```

```

    } = this.pretrainedModel;

    const {
      batchInputShape,
    } = inputLayers[0];

    return [
      batchInputShape[1],
      batchInputShape[2],
    ];
  }

private getData = async (dataType: string): Promise<ICollectedData> => {
  if (dataType !== 'train' && dataType !== 'eval') {
    throw new Error(`Datatype ${dataType} unsupported`);
  }

  return {
    xs: this.data[dataType].xs,
    ys: this.data[dataType].ys,
    classes: this.data.classes,
  };
}

public getModel = () => this.model;

public addData = async (origImages: Array<tf.Tensor | IImageData |
HTMLImageElement | string>, origLabels: string[], dataType: string = 'train') =>
{
  this.callbackFn('onAddData', 'start', origImages, origLabels, dataType);
  if (!origImages) {
    throw new Error('You must supply images');
  }
  if (!origLabels) {
    throw new Error('You must supply labels');
  }

  const dims = await this.getInputDims();
  const {
    images,
    errors,
    labels,
  } = await translateImages(origImages, dims, origLabels);
  if (images.length !== labels.length) {
    throw new Error('Class mismatch between labels and images');
  }

  if (dataType === 'train' || dataType === 'eval') {
    const activatedImages: tf.Tensor[] = [];
    for (let i = 0; i < images.length; i++) {
      const image = images[i];
      // TODO: Debug this any type
      const activatedImage: any = await this.cropAndActivateImage(image);
      activatedImages.push(activatedImage);
      await tf.nextFrame();
    }

    this.data.classes = getClasses(labels);
    const xs = addData(activatedImages);
    await tf.nextFrame();
    const ys = addLabels(labels, this.data.classes);
    await tf.nextFrame();
    this.data[dataType] = {
      xs,
      ys,
    };
  }
}

```

```

    };
  }

  await tf.nextFrame();
  this.callbackFn('onAddData', 'complete', origImages, labels, dataType,
errors);
}

public clearData = async (dataType?: string) => {
  this.callbackFn('onClearData', 'start', dataType);
  if (dataType) {
    this.data[dataType] = { };
  }

  this.data['train'] = {};
  this.data['eval'] = {};

  this.callbackFn('onClearData', 'complete', dataType);
}

public train = async (params: IParams = {}) => {
  this.callbackFn('onTrain', 'start', params);
  await this.loaded();
  const data = await this.getData('train');

  if (!data.xs) {
    throw new Error('You must add some training examples');
  }
  if (!data.ys) {
    throw new Error('You must add some training labels');
  }

  const classes = Object.keys(data.classes).length;
  if (classes <= 1) {
    throw new Error('You must train with more than one class');
  }
  const {
    model,
    history,
  } = await train(this.pretrainedModel, data, classes, params, this.args);

  this.model = model;
  this.callbackFn('onTrain', 'complete', params, history);
  return history;
}

public predict = async (origImage: tf.Tensor | HTMLImageElement | string,
label?: string) => {
  try {
    this.callbackFn('onPredict', 'start', origImage);
    await this.loaded();
    if (!this.model) {
      throw new Error('You must call train prior to calling predict');
    }
    const dims = await this.getInputDims();
    const {
      images,
      errors,
    } = await translateImages([origImage], dims);
    if (errors && errors.length && !images[0]) {
      throw errors[0].error;
    }
    const data = images[0];
    const img = await this.cropAndActivateImage(data);
    // TODO: Do these images need to be activated?
  }
}

```

```

const predictedClass = tf.tidy(() => {
  const predictions = this.model.predict(img);
  // TODO: address this
  return (predictions as tf.Tensor).as1D().argMax();
});

const classId = (await predictedClass.data())[0];
predictedClass.dispose();
const prediction = Object.entries(this.data.classes).reduce((obj, [
  key,
  val,
]) => ({
  ...obj,
  [val]: key,
}), {})[classId];
this.callbackFn('onPredict', 'complete', origImage, label, prediction);
return prediction;
} catch(err) {
  console.error(err, origImage, label);
  // throw new Error(err);
}
}

public evaluate = async (params: IParams = {}) => {
  this.callbackFn('onEvaluate', 'start', params);
  await this.loaded();
  if (!this.model) {
    throw new Error('You must call train prior to calling predict');
  }
  const data = await this.getData('eval');

  if (!data.xs || !data.ys) {
    throw new Error('You must add some evaluation examples');
  }

  const evaluation = await this.model.evaluate(data.xs, data.ys, params);
  this.callbackFn('onEvaluate', 'complete', params, evaluation);
  return evaluation;
}

// handlerOrURL?: tf.io.IOHandler | string;
public save = async(handlerOrURL?: string, params: IParams = {}) => {
  this.callbackFn('onSave', 'start', handlerOrURL, params);
  await this.loaded();
  if (!this.model) {
    throw new Error('You must call train prior to calling save');
  }

  const savedModel = await this.model.save(handlerOrURL ||
getDefaultDownloadHandler(this.data.classes), params);
  this.callbackFn('onSave', 'complete', handlerOrURL, params, savedModel);
  return savedModel;
}
}

export default MLClassifier;

export { PRETRAINED_MODELS_KEYS as PRETRAINED_MODELS } from
'./loadPretrainedModel';

```

Лїстїнг файлу loadPretrainedModel.js

```

import * as tf from '@tensorflow/tfjs';

export const PRETRAINED_MODELS_KEYS = {
  MOBILENET: 'mobilenet_v1_0.25_224',
}

export const PRETRAINED_MODELS = {
  [PRETRAINED_MODELS_KEYS.MOBILENET]: {
    url: 'https://storage.googleapis.com/tfjs-
models/tfjs/mobilenet_v1_0.25_224/model.json',
    layer: 'conv_pw_13_relu',
  },
};

const loadPretrainedModel = async (pretrainedModel: string | tf.Model =
PRETRAINED_MODELS_KEYS.MOBILENET) => {
  if (pretrainedModel instanceof tf.Model) {
    return pretrainedModel;
  }

  if (!PRETRAINED_MODELS[pretrainedModel]) {
    throw new Error('You have supplied an invalid key for a pretrained model');
  }

  const config = PRETRAINED_MODELS[pretrainedModel];
  const model = await tf.loadModel(config.url);
  const layer = model.getLayer(config.layer);
  return tf.model({
    inputs: [model.inputs[0]],
    outputs: layer.output,
  });
};

export default loadPretrainedModel;

```

Додаток Е

Презентаційний матеріал

Кваліфікаційна робота Бакалавра

Метод нейромережевої класифікації документів що посвідчують особу



Виконав:
студент групи КН-19-2
А.В. Похитун

Керівник:
к.т.н., доцент кафедри
О.В. Мазурець

Актуальність

У ХХІ столітті документи є невід'ємною частиною кожної сучасної людини. Особливо велика кількість часу витрачається на роботу з документами у різних державних установах .

Для ідентифікації особи, автоматизованої авторизації та автоматизованого витягу персональних даних ефективним шляхом є розпізнавання візуальних відомостей, що містяться на офіційних документах, які посвідчують особу (фото, прізвище, ім'я та по батькові особи, дата народження, номери документів тощо). Методи, що використовуються для розпізнавання відомостей з документів, відрізняються у залежності від типу документу. Тому процесу аналізу документу передує важливий крок – класифікація зображень цих документів.

Оскільки обробка зображення документу є автоматизованим процесом, то і процес класифікації зображень документів теж має виконуватись автоматизовано. Це визначає мету даної роботи.

Мета та постановка задачі роботи

Метою кваліфікаційної роботи бакалавра є:

- розробити метод неймережевої класифікації зображень документів;
- спроектувати структуру інформаційної системи класифікації зображень документів, що використовує технологію неймережевої класифікації зображень документів;
- виконати програмну реалізацію інформаційної системи;
- провести тестування розробленої інформаційної системи класифікації зображень документів.
- розробити архітектуру згорткової нейронної мережі для класифікації зображень документів;

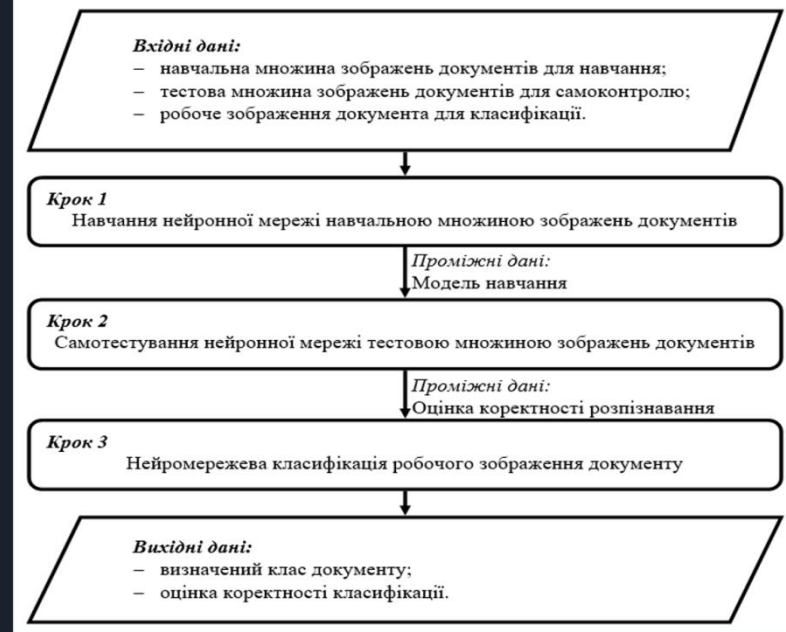


Схема методу неймережевої класифікації зображень документів що посвідчують особу

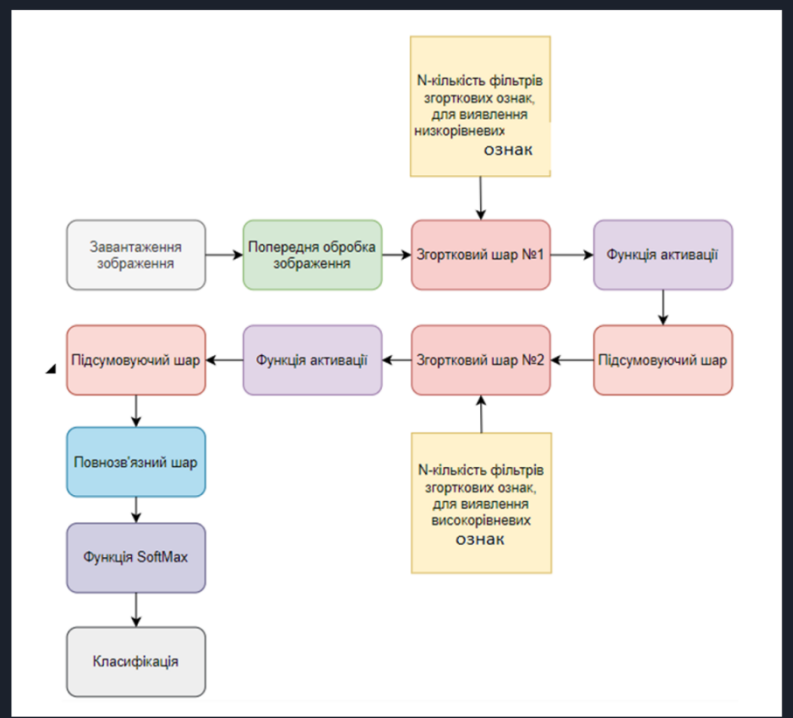
Метод неймережевої класифікації

Метод неймережевої класифікації зображень документів що посвідчують особу призначений для перетворення вхідних даних у вигляді навчальної множини зображень документів для навчання.

Неймережева класифікація робочого зображення документа полягає у тому, що користувач обирає фото документа та завантажує у вже навчену модель. Після чого відбувається процес розпізнавання фото.



Діаграма послідовності операцій згорткової нейронної мережі за класифікування зображень

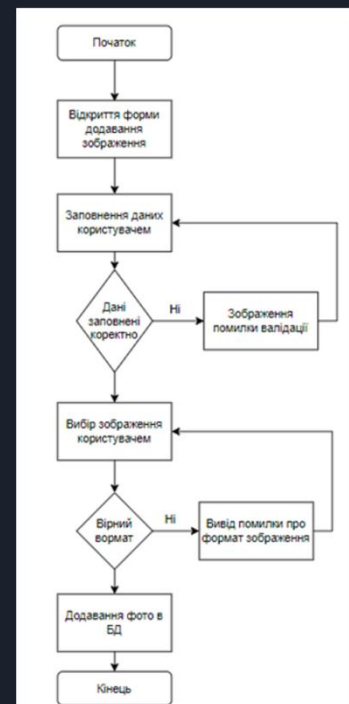


Датасет

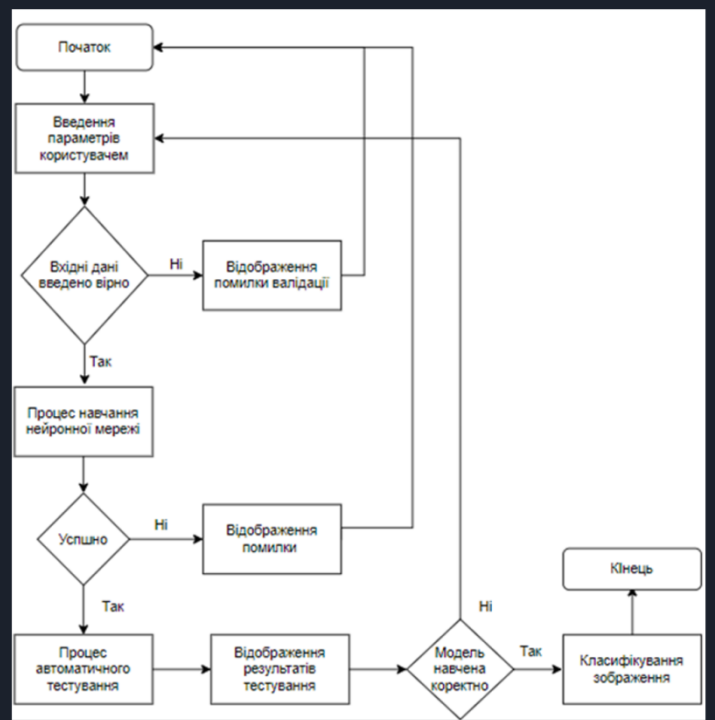
Датасет із документами українського зразку в відкритому доступі не знайдено, тому в роботі було використано один з існуючих відкритих датасетів із зображеннями закордонних документів.



Для початку користувачу потрібно відкрити форму додавання зображення. Наступним кроком буде заповнення даних, назва фото, клас. Далі потрібно вибрати фото, після чого натиснути кнопку додати. Після цього запустяться функції валідації і якщо все вірно, то фото буде додане в БД.



Для початку користувачу потрібно зайти на сторінку класифікатора. Далі на формі налаштування нейронної мережі користувач вибирає кількість фото та класи які навчать нейронну мережу. Якщо дані пройдуть функції валідації мережа автоматично розпочне своє тестування. Після цього дані будуть виведені користувачеві, якщо користувач вирішить що результати тестування незадовільні, він зможе додати модель. Якщо ж користувача все влаштовує, він може приступати до класифікування фото.



Програмна реалізація

Навчання нейронної мережі

20 Кількість фото кожного класу

Паспорт
 Водійське посвідчення
 Віза

Навчити

Автоматична система тестування

Тестування проведено успішно
 Точність: 89%
 Кількість коректно розпізнаних фото: 8/9

Додаткове навчання моделі не потребується



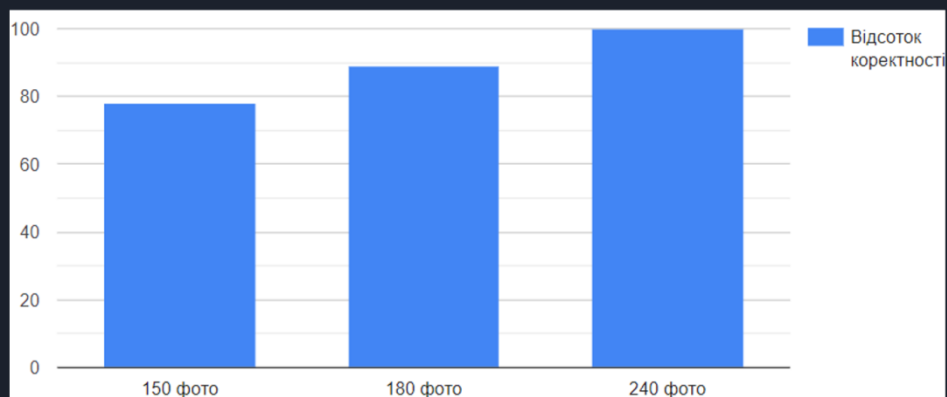
Класифікатор

Клас: Паспорт

Оцінка коректності: Вірно Не вірно

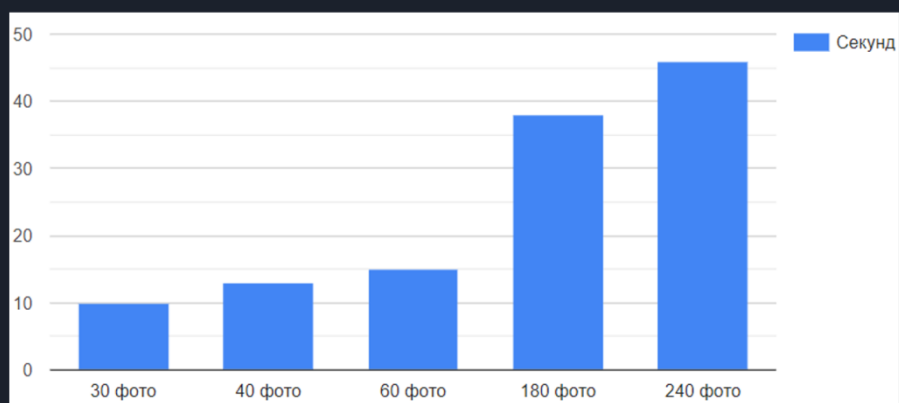
Дослідження коректності класифікації

К-сть	%	Сп.
150	79%	6/9
180	89%	7/9
240	100%	9/9



Дослідження швидкості навчання

К-ть	Шв.
30	10
40	13
60	15
180	38
240	47



Висновок

Під час виконання кваліфікаційної роботи бакалавра було розроблено метод нейромережевої класифікації зображень документів що посвідчують особу, який призначений для перетворення вхідних даних у вигляді навчальної множини зображень документів для навчання, тестової множини зображень документів для самоконтролю та робочого зображення документа для класифікації у вихідні дані у вигляді визначеного клас документа та оцінки коректності класифікації й містить кроки навчання й самотестування нейронної мережі навчальною множиною зображень документів, а також нейромережевої класифікації робочого зображення документа.

Ім'я користувача:
Кафедра КН

ID перевірки:
1015421729

Дата перевірки:
05.06.2023 09:32:15 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
05.06.2023 09:39:54 EEST

ID користувача:
100005671

Назва документа: КН-19-2 Похитун

Кількість сторінок: 65 Кількість слів: 9139 Кількість символів: 71977 Розмір файлу: 2.40 MB ID файлу: 1015084184

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

6.13% Схожість

Найбільша схожість: 1.84% з джерелом з Бібліотеки (ID файлу: 1015072628)

4.54% Джерела з Інтернету

313

Сторінка 67

3.72% Джерела з Бібліотеки

90

Сторінка 69

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2

Підозріле форматування

16
сторінок

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 2.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 12%**

ID: 114685 Назва: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Додано в БД: 2023-06-05 Автора: А.В. Похитун Керівники: О.В. Мазурець Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	58425	909	2362 (4%)	34 (4%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод нейромережевої класифікації зображень документів що посвідчують особу

Автор: студент групи КН-19-2 Похитун Андрій Вікторович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доцент кафедри КН Мазурець О.В.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі Похитун А.В., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали статей.

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

- за системою Anti-Plagiarism: 2%;

- за системою Unichек: 6,13 %

Керівник роботи

Олександр МАЗУРЕЦЬ

Гарант ОП

Олександр МАЗУРЕЦЬ

Завідувач кафедри КН

Олександр БАРМАК



**ВІДГУК НАУКОВОГО КЕРІВНИКА
на кваліфікаційну роботу бакалавра**

студента *гр. КН-19-2 Похитуна Андрія Вікторовича*

за темою Метод нейромережевої класифікації зображень документів що посвідчують особу

1. Актуальність теми

Актуальним завданням, що розглядається в даній роботі, є розробка та реалізація методу нейромережевої класифікації зображень документів що посвідчують особу. Застосування методу нейромережевої класифікації зображень документів що посвідчують особу відкриває нові можливості для покращення швидкості та точності ідентифікації типів документів.

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

За описом предметної області об'єктом дослідження є процес класифікації зображень документів що посвідчують особу для систем ідентифікації. Предметом дослідження є моделі, методи, алгоритми та засоби для класифікації зображень документів що посвідчують особу. Метою роботи є розробка та програмна реалізація методу нейромережевої класифікації зображень документів що посвідчують особу. Для вирішення цього завдання були застосовані математичні моделі, методи та алгоритми, спрямовані на вирішення теоретичних і практичних проблем, які виникають при розробці інформаційних технологій. Таким чином, кваліфікаційна робота бакалавра відповідає стандарту бакалавра у галузі комп'ютерних наук (спеціальність 122).

3. Професійні та особистісні якості бакалавра

Під час написання пояснювальної записки та розробки прикладного програмного забезпечення, студент Похитун Андрій Вікторович продемонстрував високий рівень володіння професійними навичками спеціальності "Комп'ютерні науки". Студент успішно впорався з поставленими завданнями своєї кваліфікаційної роботи бакалавра і продемонстрував високу професійну компетентність.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Студент Похитун Андрій Вікторович виконував усі поставлені завдання самостійно. Його робота є його особистим надбанням.

5. Ступінь оволодіння методами дослідження

Студент Похитун Андрій Вікторович продемонстрував вміння аналізувати та розв'язувати складні завдання, використовуючи сучасні методи та підходи. Він також виявив гнучкість та комплексний підхід до роботи, здатність до самостійного вирішення проблем.

6. Повнота та якість розкриття теми роботи

Студент повністю та якісно розкрив тему роботи, детально розглянув предметну область досліджуваної теми.

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

Матеріал в роботі викладено логічно, послідовно та аргументовано. Літературна грамотність тексту роботи має високий рівень.

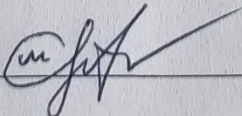
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Практичне застосування методу нейромережевої класифікації зображень документів, що посвідчують особу, полягає у швидкій та точній автоматичній ідентифікації типу документів, для спрощення процесу перевірки особистості та покращення систем безпеки.

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

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

Керівник



к.т.н., доцент каф. КН Олександр МАЗУРЕЦЬ



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-19-2 Похитуна Андрія Вікторовича

за темою: Метод нейромережевої класифікації зображень документів що посвідчують особу

1. Актуальність обраної теми

Актуальність теми "Метод нейромережевої класифікації зображень документів, що посвідчують особу" полягає у забезпеченні швидкої та надійної автоматизованої ідентифікації типу документів, що має велике значення для поліпшення систем безпеки, впровадження ефективних механізмів перевірки особистості та захисту від шахрайства.

2. Повнота розкриття мети та завдань роботи

Мета та завдання роботи за темою "Метод нейромережевої класифікації зображень документів, що посвідчують особу" розкриті повністю.

3. Зміст кожного розділу роботи

У розділі 1 студент детально розглянув предметну область, що пов'язана з досліджуваною темою. Описав особливості застосування згортоквих нейронних мереж та проаналізував існуючі подібні рішення. У розділі 2 описано кроки методу нейромережевої класифікації зображень документів та його особливості: навчання, класифікацію, тестування. Наведено та описано архітектуру нейронної мережі. Розглянуто особливості налаштування гіперпараметрів згорткової нейронної мережі. Спроектовано структуру інформаційної системи, підготовано зразки для навчання та тестування нейронної мережі. У розділі 3 детально описано програмну реалізацію інформаційної системи та наведено структуру модулів інформаційної системи. Проведено дослідження ефективності застосування методу нейромережевої класифікації зображень документів що посвідчують особу.

4. Оцінка розробленої інформаційної системи, її практична цінність

Дана інформаційна система має практичну цінність, адже дозволить пришвидшити процес класифікації зображень документів що посвідчують особу – це підвищить якість надання послуг у різних сферах діяльності.

5. Якість оформлення кваліфікаційної роботи бакалавра

Кваліфікаційна робота студента Похитуна Андрія Вікторовича оформлена якісно та згідно вимог нормоконтролю.

6. Недоліки кваліфікаційної роботи бакалавра

В ряді випадків задля покращення компоновки рисунки розміщено раніше ніж згадування на них зустрічається в тексті. Скорочення в переліку скорочень не відсортовано за алфавітом. Втім наведене не впливає на одержані результати.

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

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

Рецензент

Яшике С.М., доц. кер. ІІІ