

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

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації та управління

До захисту допущено:

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ____ ” _____ 2021 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 126 «Інформаційні системи та технології»**

на тему: «Інформаційна система розпізнавання числової
інформації»

Виконала:

студентка IV курсу, групи ІС-72

Кравець Олександра Олегівна

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник

доц., к.т.н., доц. Сперкач Майя Олегівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

**Консультант з
графічної
документації**

доц., к.т.н., Новінський Валерій Петрович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

Рецензент

доц., к.т.н., доц. Писаренко Андрій

Володимирович

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

_____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студентка _____
(підпис)

Київ – 2021 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2021 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Кравець Олександрі Олегівні
(прізвище, ім'я, по батькові)

1. Тема проєкту «*Система розпізнавання числової інформації*»,

керівник проєкту Сперкач Майя Олегівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “ 11 ” 05 2021 р. № 1139

2. Термін подання студентом проєкту “04”червня 2021 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури масивів з інформацією

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використання

2. Схема структурна класів програмного забезпечення

2. Схема структурна послідовності програмного забезпечення

4. Схема структурна компонентів програмного забезпечення

5. Креслення вигляду екранних форм

6. Консультанти розділів проєкту

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

7. Дата видачі завдання «7» квітня 2021 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення рекомендованої літератури	10.04.2021	
2.	Аналіз існуючих методів розв'язання задачі	14.04.2021	
3.	Постановка та формалізація задачі	16.04.2021	
4.	Розробка інформаційного забезпечення	20.04.2021	
5.	Алгоритмізація задачі	24.04.2021	
6.	Обґрунтування використовуваних технічних засобів	28.04.2021	
7.	Розробка програмного забезпечення	29.04.2021	
8.	Налагодження програми	09.05.2021	
9.	Виконання графічних документів	10.05.2021	
10.	Оформлення пояснювальної записки	12.05.2021	
11.	Подання ДП на попередній захист	14.05.2021	
12.	Подання ДП на основний захист	04.06.2021	
13.	Подання ДП рецензенту	07.06.2021	

Студент

Олександра КРАВЕЦЬ

Керівник

Майя СПЕРКАЧ

[illegible]

Пояснювальна записка до дипломного проєкту

на тему: *«Інформаційна система розпізнавання числової
інформації»*

Київ – 2021 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 22 рисунки, 4 таблиці, 1 додаток та 37 джерел.

Дипломний проєкт присвячений вирішенню задачі розпізнавання, класифікації та подальшої обробки та збереження рукописної числової інформації.

Метою створення системи є спрощення процесу перевірки контрольних робіт студентів або учнів (де в якості відповідей маємо рукописні числові значення) за рахунок автоматизації цього процесу шляхом розпізнавання написаних студентами (учнями) відповідей у відповідному бланку для відповідей за допомогою моделей машинного навчання.

У розділі загальних положень встановлено мету, цілі та задачі розробки, визначено функціональні границі системи та побудована структурна схема варіантів використання, проаналізовано існуючі аналоги та встановлено відмінність від них системи, що проектується.

У розділі інформаційного забезпечення надано детальний опис вхідних та вихідних даних, описано структуру масиву з інформацією, який використовується у даній системі.

Розділ математичного забезпечення присвячений опису змістовної та математичної постановки задачі, аналізу існуючих методів розв'язання задачі даного дипломного проєкту та обґрунтування вибору одного з них з його подальшим детальним описом.

Розділ програмного забезпечення описує засоби розробки програмного продукту та етапи проектування його архітектури. Описано

					ДП 7233.00.000 ПЗ		
		Прізвище	Підпис	Дата			
Розроб.		Кравець О. О.			Система розпізнавання числової інформації	Літ.	Лист
Перевірив.		Сперкач М. О.					Листів
							2
Н. кон.		Новінський В.П.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72	
Затв.		Сперкач М. О.					

специфікацію функцій та звіти, які генеруються в ході запуску програми.

У технологічному розділі визначено мету проведення випробувань програмного продукту та описано їх результати.

ОБРОБКА ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ,
РУКОПИСНІ ЧИСЛА, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

ABSTRACT

The structure and the scope of the work. Explanatory note of the diploma project consists of five sections, contains 22 drawings, 5 tables, 1 application and 37 sources.

The diploma project is devoted to solving the problem of recognition, classification, further processing and saving of handwritten numerical information.

The system purpose is simplifying the process of checking the student's tests (where the answers are handwritten numerical values) by automating this process by recognizing the answers written by students in the appropriate form for answers using machine learning models.

In the general terms section the developmt purpose and are established, functional borders of system are defined and the structural scheme of variants of use is constructed, the existing analogues are analyzed and the difference from them of the projected system is established.

The information support section provides a detailed description of the input and output data, as well as the structure description of the information array, that is used in this system is provided.

The mathematical support section is devoted to the meaningful and mathematical formulation of the problem, analysis of the existing methods for solving the problem of this thesis project and justification of choosing one of them with its subsequent detailed description.

The software support section describes the software development tools and the stages of designing its architecture. The specification of functions and reports generated during program startup is described.

The technology section defines the purpose of testing the software product and describes their results.

KEYWORDS: IMAGE PROCESSING, MACHINE LEARNING, NEURAL NETWORKS, HANDWRITTEN DIGITS, IMAGE CLASSIFICATION

					ДП 7233.00.000 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	7
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА	10
1.1.1 <i>Опис процесу діяльності</i>	11
1.1.2 <i>Опис функціональної моделі</i>	11
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	12
1.3 ПОСТАНОВКА ЗАДАЧІ	13
1.3.1 <i>Призначення розробки</i>	13
1.3.2 <i>Цілі та задачі розробки</i>	13
Висновок до розділу	14
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	16
2.1 ВХІДНІ ДАНІ	16
2.2 ВИХІДНІ ДАНІ	16
2.3 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ	17
Висновок до розділу	19
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	21
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	21
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ	21
3.3 ОБГРУНТУВАННЯ МЕТОДІВ РОЗВ'ЯЗАННЯ	27
3.4 ОПИС МЕТОДУ РОЗВ'ЯЗАННЯ	31
Висновок до розділу	37
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	38
4.1 ЗАСОБИ РОЗРОБКИ	38
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	40
4.2.1 <i>Загальні вимоги</i>	40
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	41
4.3.1 <i>Діаграма класів</i>	41
4.3.2 <i>Діаграма послідовності</i>	42
4.3.3 <i>Діаграма компонентів</i>	43
4.3.4 <i>Специфікація функцій</i>	44

4.4	Опис звітів.....	48
	Висновок до розділу	51
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	52
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	52
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	55
5.2.1	Мета випробувань.....	55
5.2.2	Загальні положення.....	56
5.2.3	Результати випробувань.....	57
	Висновок до розділу	60
	ЗАГАЛЬНІ ВИСНОВКИ	61
	ПЕРЕЛІК ПОСИЛАНЬ	63
	ДОДАТОК А.....	81

ВСТУП

Люди можуть бачити і візуально відчувати навколишній світ, використовуючи свої очі і мозок. Комп'ютерний зір працює над тим, щоб комп'ютери мали можливість бачити та оброблювати зображення таким же чином, як це робить людський зір. На даний момент вже є розробленою низка алгоритмів в області комп'ютерного зору для розпізнавання зображень.

Класифікація об'єктів між собою є для нас досить легким завданням, але виявляється доволі складним для машин, і тому класифікація зображень є важливим завданням у галузі комп'ютерного зору. Цей процес являє собою віднесення зображень до одного із низки заздалегідь визначених класів.

Потенційно може існувати n кількість класів, за якими є можливість класифікувати певне зображення. Ручна перевірка та класифікація зображень може бути виснажливим завданням, особливо коли їх кількість величезна (наприклад, 20 000), і тому буде дуже корисно, якщо цей буде автоматизований за допомогою комп'ютерного зору.

Прикладом використання класифікації зображень у реальному світі можуть слугувати досягнення у галузі автономного водіння також. Є можливим створення моделі класифікації зображень, яка розпізнає різні об'єкти, такі як інші транспортні засоби, пішоходи, світлофори та покажчики на дорозі.

Дана задача вирішуються завдяки застосуванню методів машинного навчання.

Машинне навчання, що сприяє вирішенню цієї проблеми - являє собою підгалузь штучного інтелекту, яка імітує процес навчання, який відбувається у людей. Ідея полягає в тому, щоб надати програмі необхідний набір даних, на основі яких, програма б визначала би свої правила, за якими потім буде сама працювати.

Мета машинного навчання полягає в тому, щоб відчувати,

					ДП 7233.00.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

запам'ятовувати, вчитися та розпізнавати, як людина. Одним з найбільш затребуваних методів машинного навчання є глибоке навчання. Ідея глибокого навчання виникла ще в 1950-х роках із визначенням поняття персептрону. Персептрон був першою машиною, яка мала здатність відчувати і вчитися. Тоді у 1980-х рр. була визначена багат шарова структура персептрону з обмеженою кількістю прихованих шарів. Однак можливості навчання персептрону були дуже обмеженими. Таким чином, його не враховували роками до висунення пропозиції про нейронні мережі з безліччю прихованих шарів у 2000-х. Коннекціонізм, запроваджений різними рівнями репрезентації, покращив можливості навчання в архітектурі. Власне, головним аспектом глибокого навчання є цей багаторівневий феномен репрезентації. Він походить від теорії, згідно з якою мозок обробляє інформацію у декілька шарів. Оскільки алгоритми глибокого навчання вдосконалювались, через що зростала схожість архітектури з біологічними нейронними мережами.

Даний дипломний проєкт присвячений процесу розпізнаванню числової інформації написаної від руки (у вигляді відповідей студента або учня), яку буде отримано з відповідного бланку для відповідей, її подальша обробка і збереження вже в печатному вигляді.

Практичне значення одержаних результатів. Розроблено та натреновано модель класифікації зображень рукописних чисел, що представляють собою відповіді з бланку та реалізована подальша обробка отриманого списку чисел з метою його оцінення, ґрунтуючись на списку з правильними відповідями, якими повинен бути заповнений відповідний бланк. .

Публікації. Результати роботи були опубліковані у статті Кравець О.О., М.О. Сперкач Задача розпізнавання рукописної числової інформації / О.О Кравець, М.О. Сперкач // Матеріали VI всеукраїнської науково-практичної

					ДП 7233.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 22-23 квітня 2021 р.

					ДП 7233.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Розпізнавання чисел є темою досліджень протягом багатьох років завдяки своїм широким сценаріям застосування, включаючи обробку форм для відповідей, де сканується рукописний текст з заповненої форми; розпізнавання поштових індексів, що сприяє полегшенню процесу сортування поштових повідомлень; розпізнавання введених цифр рукописного вводу на смартфонах, обробка банківських чеків та розпізнавання числових записів (наприклад, рік або номер сторінки документу) цифрованих рукописних документів. У цих контекстах існуючі набори даних частіше за все базуються на відсканованих рукописних цифрах та літерах з різних алфавітів [1].

Завдяки великій та постійно зростаючій кількості рукописних документів існує великий попит на перетворення цих документів у цифрові копії записів, які є більш доступними через цифрові системи, такі як цифрові форми та бази даних. Для автоматичного переходу від рукописних чисел до їх цифрової версії багатозначне розпізнавання є неминучим та корисним [2].

Машинне навчання, що допомагає у вирішенні цієї проблеми - являє собою підгалузь штучного інтелекту, що імітує процес навчання у людей. Ідея тут полягає в тому, щоб надати програмі необхідний набір даних, на основі яких, програма б визначала би свої правила, за якими потім буде працювати [3].

Якщо говорити про обробку форм для відповідей, то можна виділити процес перевірки викладачем великої кількості робіт студентів, який завжди вимагає великої кількості часу та високого рівня уваги, тому питання створення застосунку, який би допомагав їм у цьому питанні є достатньо актуальним.

Саме через це, у даній дипломній роботі пропонується автоматизоване рішення задачі перевірки контрольних робіт студентів та учнів, відповіді яких подаються у вигляді числової інформації, написаною від руки.

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

1.1.1 Опис процесу діяльності

Дана робота буде присвячена процесу розпізнаванню числової інформації написаної від руки (у вигляді відповідей студента або учня), яку буде отримано з відповідного бланку для відповідей, її подальша обробка і збереження вже в печатному вигляді.

У процесі вирішенні поставленої задачі можна виділити такі етапи:

- підготовка вибірок даних з великою кількістю зображень рукописних чисел для тестування моделей машинного навчання та навчання нейронної мережі на підготовленій вибірці;
- виділення окремих фрагментів отриманого бланку, звідки буде зчитуватись інформація;
- класифікація зображень чисел, написаних від руки до певного числа;
- розрахунок оцінки, яку отримає людина, чий бланк було перевірено;

1.1.2 Опис функціональної моделі

Визначимо акторів системи та дії, які вони можуть виконувати в рамках системи, спроектувавши функціональну модель системи.

У системі розпізнавання числових символів функціонує лише один актор – користувач.

На структурній схемі варіантів використання, було розглянуто дії, що може виконати користувач системи.

Схема структурна варіантів використання наведена в графічному матеріалі.

Розглянемо детальніше деякі з варіантів використання:

- завантажити до системи фотографію заповненого бланку: користувач робить фотографію бланку з відповідями студента та завантажує її до системи, а сама система у цей час спочатку розпізнає сам бланк з відповідями, потім розпізнає та зчитує відповідні числові символи(у

- вигляді відповідей студента);
- завантажити до системи файл з відповідями: для того щоб системи мала можливість виставити бали студенту, вона повинна мати дані, з якими треба порівняти числові відповіді, які були отримані з бланку;
 - отримати оцінку одразу, без збереження її значення до файлу: користувач має можливість одразу після завантаження зображення та його обробки, побачити оцінку студента, без подальшого її збереження;
 - перевірити правильність сканування оцінок: користувач отримує список з відповідями з бланку на головному екрані.

1.2 Огляд наявних аналогів

Під час пошуку аналогів не було виявлено програмних рішень з функціональністю, яка б дозволяла перевіряти роботи учнів на основі розпізнавання їх відповідей, написаних від руки у вигляді числових значень, отриманих з завантаженого зображення заповненого бланку . Проте було знайдено аналоги зі схожою функціональністю, де можна сканувати рукописний текст, з завантаженого зображення.

Розглянемо деякі з них детальніше:

- ABBYY FineReader

Перевага :

Розпізнає рукописний текст на 192 мовах, при цьому для 48 мов підтримка перевірки орфографії.

- SimpleOCR

Перевага :

Підтримує інтеграцію з іншими програмами по розпізнаванню тексту.

- OCR Desktop

Перевага :

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Для розпізнавання текстів використовує нейромережу, якій для навчання надали 4 мільйони прикладів, завдяки чому точність розпізнавання висока.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням розробки є розпізнавання рукописної числової інформації, яку буде отримано у вигляді відповідей з бланку, та її подальше оцінювання.

1.3.2 Цілі та задачі розробки

Метою розробки є спрощення процесу перевірки контрольних робіт студентів або учнів (де в якості відповідей маємо числові значення) за рахунок автоматизації цього процесу шляхом розпізнавання написаних студентами (учнями) відповідей у відповідному бланку для відповідей за допомогою моделей машинного навчання.

Завдяки створеній системі користувачі, у вигляді людей, які перевіряють роботи, можуть суттєво зменшити час для їх перевірки.

Робота присвячена підтримці процесу обробки та класифікації зображень чисел, адже перед тим як зчитати необхідну інформацію, треба обробити зображення з бланку, прибравши всі сторонні предмети з зображення та виділивши окремі фрагменти зображення, звідки буде зчитуватись інформація.

Для досягнення мети необхідно вирішити такі задачі:

- підготовка вибірок даних для тестування моделей машинного навчання;
- навчання нейронної мережі на підготовленій вибірці;
- класифікація зображень чисел, написаних від руки до певного числа;

- обробка зображення бланку з відповідями (вирівнювання, видалення зайвих частин фото)
- виділення окремих фрагментів бланку, звідки буде зчитуватись інформація;
- оцінка отриманих відповідей;
- оцінка достовірності результатів.

Висновок до розділу

Підбиваючи підсумки після написання даного розділу дипломного проєкту було описано предметне середовище, яке розглядається, а саме: був зроблений короткий опис процесу розпізнавання символів та були наведені аргументи, чому саме цей процес становить собою темою досліджень протягом багатьох років. Також були описані основні та проаналізовані сфери застосування даного процесу, серед яких була обрана одна – обробка форм для відповідей. Після більш детального аналізу обраної сфери застосування, був виокремлений процес перевірки викладачем великої кількості робіт студентів, який завжди вимагає великої кількості часу та високого рівня уваги. Отже, у даній дипломній роботі було запропоновано автоматизоване рішення задачі перевірки контрольних робіт студентів та учнів, відповіді яких подаються у вигляді числової інформації, написаною від руки.

Були сформульовані призначення, мета та задачі розробки даної системи.

Було проведено огляд існуючих аналогів даного програмного забезпечення, після пошуку яких не було виявлено програмних рішень з функціональністю, яка б дозволяла перевіряти роботи учнів на основі розпізнавання їх відповідей, які написані від руки.

Проте було знайдено аналоги зі схожою функціональністю, де можна сканувати рукописний текст, з завантаженого зображення, до яких входять

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

Abby FineReader, який розпізнає рукописний текст на багатьох мовах, і на деяких з яких підтримує перевірку орфографії, наступним є Simple OCR, який підтримує інтеграцію з іншими подібними сервісами, та OCR Desktop, який має досить високу точність розпізнавання.

Були виявлені функціональні границі системи, після чого була створена функціональна модель системи, яка являє собою опис користувача система з його можливими діями у системі. Ця модель системи представлена у вигляді структурної схеми варіантів використання.

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані для розроблюваної інформаційної системи можна розділити на два типи:

- дані для правильного функціонування системи (набір даних для навчання нейронної мережі);
- дані, які надходять від користувача (зображення бланку для аналізу та файл з правильними відповідями для проведення перевірки).

Для процесу навчання нейронної мережі будемо використовувати набір даних MNIST. Сама назва «MNIST» - це аббревіатура, що з англійської розшифровується як Modified National Institute of Standards and Technology dataset (Модифікований набір даних Національного інституту стандартів і технологій).

Що стосується даних, що вводить користувач, то в системі присутні два типи для вхідних даних:

- файл з правильними відповідями, що будуть використовуватися для перевірки відповідей з бланку. Файл подається у форматі XLSX та розміром не більше 200 Мб;
- файл з зображення бланку с заповненими полями для відповідей, Файл подається у форматі JPG або PNG та розміром не більше 200 Мб.

2.2 Вихідні дані

Головним результатом роботи системи є отримана оцінка, яка буде розрахована як відсоткове відношення відповідей студента до правильних відповідей. Також до вихідних даних буде належати список оцінок з бланку для відповідей, який буде отриманий шляхом розпізнавання

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

рукописних чисел з відповідних частин бланку.

Масив таких результатів представляє собою дані, що будуть використовуватись користувачем для проведення остаточного оцінювання автора відповідей з бланку.

2.3 Структура масивів інформації

Набір даних MNIST - це найбільш популярний набір даних для всіх інформаційних ресурсів з теми машинного навчання та класифікації зображень. Набір містить 70 000 зображень рукописних чисел, з яких потім можна створити набори даних з 60 000 навчальних зображень та 10 000 тестових зображень [5].

Кожне зображення має роздільну здатність 28x28, а числові значення пікселів вказані у різних градаціях сірого. Кожне зображення представлене матрицею 28x28, кожен елемент матриці є цілим числом від 0 до 255 (пізніше буде приведено до варіанту від 0 до 1). Ярлик(label) кожного зображення - це передбачувана цифра рукописного зображення від 0 до 9 [5].

На рисунку 2.1 представлено приклади об'єктів, що входять до набору даних MNIST.

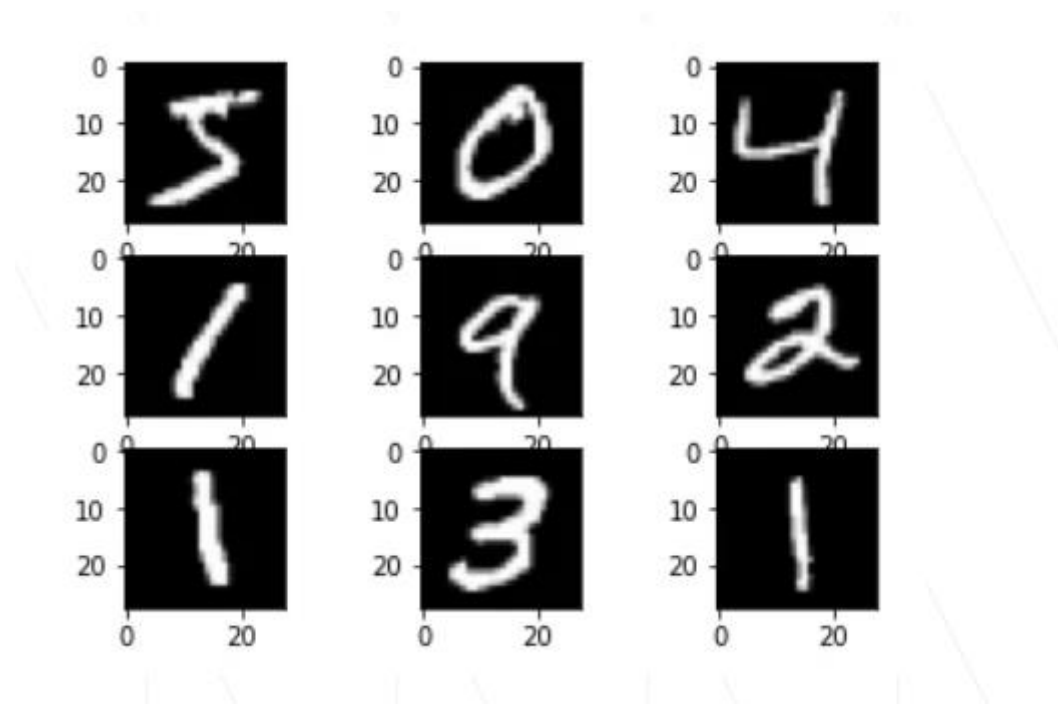


Рисунок 2.1 – Приклади об'єктів з набору даних MNIST

Описана вище структура набору даних MNIST представлена на рисунку 2.2.

```

Сумарна кількість зображень з набору даних MNIST: 70000
Кількість класів зображень з набору даних MNIST: 10
Структура даних для тренувального набору:
Структура масиву з матрицями зображень: (60000, 28, 28)
Структура масиву з ярликами(labels): (60000,)
Структура даних для тестувального набору:
Структура масиву з матрицями зображень: (10000, 28, 28)
Структура масиву з ярликами(labels): (10000,)
  
```

Рисунок 2.2 – Структура набору даних MNIST

Кількість зображень кожною цифри для тестувального набору приблизно однакова, що продемонстровано на діаграмі розподілу навчальної вибірки на рисунку 2.3.

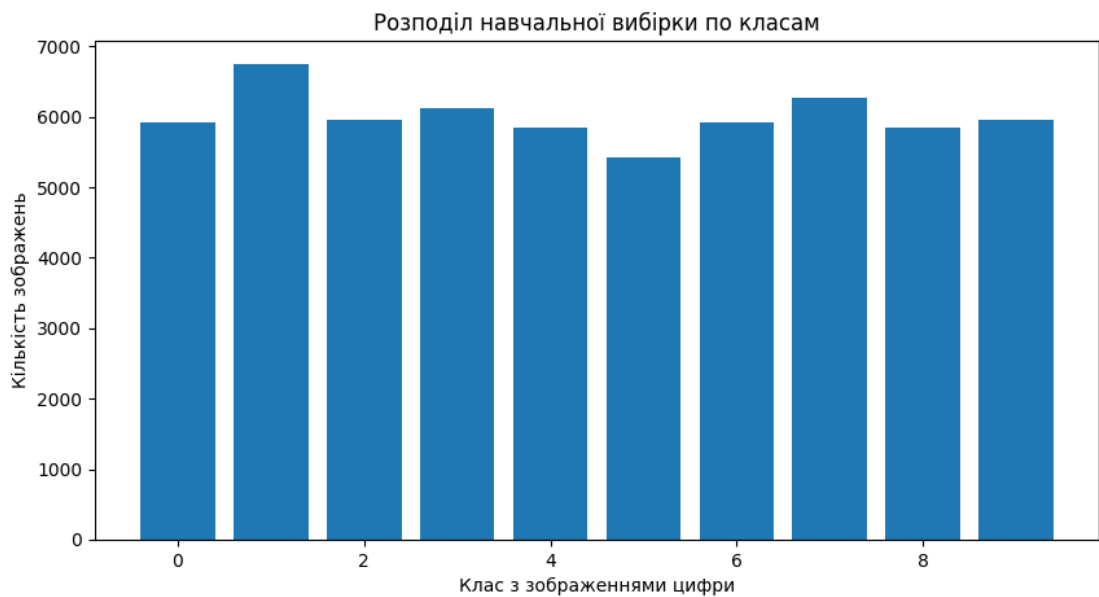


Рисунок 2.3 – Розподіл навчальної вибірки по класам цифр

При навчанні нейронної мережі було сформовано `trained_model.p` файл. Його призначення – інтерпретація результатів, що отримано на виході нейронної мережі до предметної області.

Висновок до розділу

Протягом написання розділу, присвяченому інформаційному забезпеченню, були описані вхідні та вихідні дані системи та описано структуру масиву інформації, використаного для даної системи.

До категорії вхідних даних, були використані два типи даних, а саме: дані для правильного функціонування системи (набір даних для навчання нейронної мережі) та дані, які надходять від користувача (зображення бланку для аналізу та файл з правильними відповідями для проведення перевірки).

До категорії вихідних даних входили масив чисел, отриманий шляхом розпізнавання чисел з завантаженого бланку для відповідей та оцінка, обрахована на базі отриманого масиву розпізнаних чисел та масиву чисел,

які являють собою правильні відповіді, які отримуються з другого завантаженого файлу з правильними відповідями.

Також, при розробці алгоритму розпізнавання чисел було використано нейронну мережу, що була навчена на великій вибірці зображень, які були отримані з набору даних MNIST, та наведений детальний опис даного набору даних.

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Даний програмний продукт призначений для облегшення перевірки викладачем (або вчителем) робіт студентів (або учнів), які подаються у вигляді заповненої таблиці з відповідями, які являють собою рукописні числа, шляхом автоматизації обробки даних.

Маємо вибірку з великою кількістю зображень усіх існуючих арабських цифр та заданий список з переліками назв цих цифр. За кожним з зображень закріплено ярлик (маркер) – назву цієї цифри.

Задача полягає в тому, аби обробити вхідне зображення заповненого бланку з відповідями та на основі навчаючої вибірки виконати класифікацію фрагментів бланку, які відповідають певним числовим відповідям, до певного числа.

3.2 Математична постановка задачі

Нейронні мережі, також відомі як штучні нейронні мережі (ANN) або модельовані нейронні мережі (SNN), є частиною машинного навчання і лежать в основі алгоритмів глибокого навчання. Їх назва та структура обумовлені тим, що вони імітують спосіб сигналізації біологічних нейронів один одному у мозку людини [6].

Штучні нейронні мережі (ANN) складаються з шарів вузлів, що містять вхідний шар, один або кілька прихованих шарів та вихідний шар. Кожен вузол, або штучний нейрон, з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого окремого вузла перевищує вказане порогове значення, цей вузол активується, надсилаючи дані на наступний рівень мережі. В іншому випадку дані не передаються на наступний рівень мережі [7].

У звичайній нейронній мережі існує три типи шарів:

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

- Вхідні шари (Input Layers): це рівень, в який треба заносити дані для нашої моделі. Кількість нейронів у даному шарі дорівнює загальній кількості ознак у використовуваних даних (кількість пікселів у випадку зображення).
- Прихований шар (Hidden Layer): вхідні дані з вхідного рівня передаються у прихований шар. Залежно від використовуваної моделі та розміру її даних може бути багато прихованих шарів. Кожен прихований шар має можливість мати різну кількість нейронів, яка, частіше за все, перевищує загальну кількість характеристик. Вихідні дані з кожного шару обчислюються за допомогою матричного множення вихідних даних попереднього шару із вагами, що мають вагу, цього шару, а потім за рахунок додавання зрушень, що вивчаються, з подальшою функцією активації, яка робить мережу нелінійною.
- Вихідний рівень (Output layer): Вихід із прихованого рівня надається в логістичну функцію, яка перетворює результати кожного класу в оцінку ймовірності цього ж класу.

Далі дані передаються до моделі і отримуються вихідні дані з кожного шару. Цей етап зветься прямим перенаправленням (feedforward), потім обчислюється за рахунок функції помилки, деякі загальні функції помилок - це поперечна ентропія, помилка квадратних втрат тощо. Після цього ми повертаємося до моделі за допомогою обчислення похідних. Цей етап називається зворотним розповсюдженням (Backpropagation), який в основному використовується для мінімізації витрат [7].

Переходячи до самої задачі навчання нейронної мережі, можна стверджувати, що вона полягає у коригуванні порогової функції та ваг шляхом ітераційного процесу. Розглянемо далі загальний процес для навчання нейронної мережі на основі даних, опублікованих в [8]:

Крок 1: Ініціалізація всіх ваг $w_{ij}^{(l)}$ зв'язків нейронної мережі, де $w_{ij}^{(l)}$ – вага зв'язку від нейрону i у $(l - 1)$ шарі до нейрону j у шарі l .

Крок 2: Виконання наступних дій:

Крок 2.1: Використання однієї категорії даних, саме з навчальної вибірки, і встановлення значення вхідного шару мережі (шар 0) у значення результату та значення самих даних у вихідному шарі.

Крок 2.2: Обрахунок значення для кожного входу нейрону у наступному прихованому шарі $x_j^{(l)}$ та застосування функції активації, для отримання вихідного значення для наступного прихованого шару. Повтор даного кроку для всіх наступних шарів:

$$X^{(l)} = Y^{(l-1)} W^{(l-1)}, \quad (3.1)$$

$$W^{(l-1)} \in \mathbb{R}(d(l-1) \times d(l)), \quad (3.2)$$

$$Y^{(l)} = \sigma(X^{(l)}), X^{(l)} \text{ and } Y^{(l)} \in \mathbb{R}(d(l) \times 1), \quad (3.3)$$

в яких $d(l)$ означає кількість нейронів в шарі l .

Крок 3: Обрахунок зворотнього прогону: $W^l = W^l - \mu dW^l$

Крок 3.1: Обрахунок градієнтної матриці dW для ваг з предостаннього шару до останнього – вихідного шару ($l = n$). Нехай похибка обчислень $E = \frac{1}{2}(\text{label}_j^{(n)} - y_i^{(n)})^2$, тоді градієнт кожного, є вагою між останнім та передостаннім шаром:

$$\delta^{(n)} = -(E - Y^{(n)}) \sigma'(X^{(n)}), \quad (3.4)$$

$$\delta^{(n)} \in \mathbb{R}(d(n) \times 1), \quad (3.5)$$

$$dW^{(n-1)} = (Y^{(n-1)})^T \delta^{(n)} \quad (3.6)$$

Крок 3.2: Обрахунок значення градієнту на попередніх шарах ($l = n - 1$):

$$\delta^{(l)} = W^l (\delta^{(l+1)}), \quad (3.7)$$

					ДП 7233.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

$$dW^{(l-1)} = (Y^{(l-1)})^T \delta_{(l)} \quad (3.8)$$

Повтор даного процесу до тих пір, поки всі ваги нейронної мережі не будуть оновлені.

Крок 4: При не повному виконанні заданої кількості ітерацій, є необхідним повернутися на шаг 2., в іншому випадку, процес навчання є завершеним.

3.2.1 Компоненти ANN

Компоненти, з яких складається ANN, представлені на рисунку 3.1. Ці компоненти дійсні, незалежно від того, використовується нейрон для вхідного, вихідного шару чи знаходиться у прихованому шарі[9].

На рисунку 3.1 зображене базове представлення штучного нейрона [9].

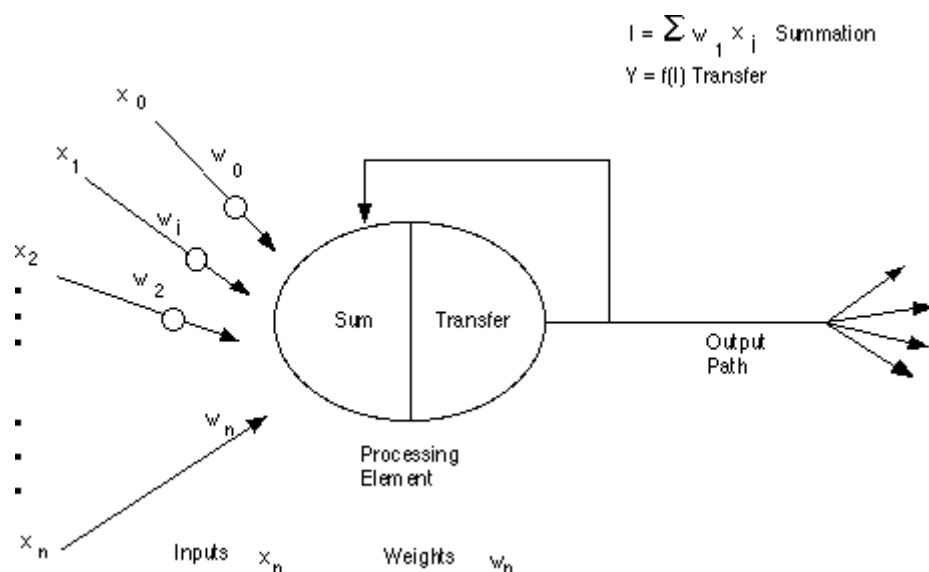


Рисунок 3.1 – Базове представлення штучного нейрона

На рисунку 3.1 символом $x(n)$ позначені різні входи в мережу. Кожен з цих входів множиться на вагу з'єднання, позначену $w(n)$. У найпростішому випадку ці продукти сумуються і надходять у функцію передачі (функцію активації) для отримання результату, і цей результат надсилається як вихідний

результат. Має місце ще можливість використання бінарних властивостей OR або AND входів разом з операцією сумування. Такі функції можуть бути вбудовані в функції сумування та передачі даних мережі[9].

Ваговий коефіцієнт:

Нейрон частіше за все отримує багато входів в один момент часу і має власну вагу. Вагові коефіцієнти є адаптивними, які позначають інтенсивність вхідного сигналу і є мірою сили з'єднання вхідного сигналу[9].

Функція підсумовування:

Функція сумування додає добуток нейрона та їх відповідну вагу, наприклад $\sum x_n w_n$. Функція підсумовування може бути більш складною, ніж простий добуток [9].

Функція передачі (активації):

Кінцевий результат функції сумування передається на вихід через алгоритмічний процес, який також має назву передавальна функція. Цю функцію можна розділити на дві категорії [9].

а) Поріг:

Вихід функції сумування порівнюється з деяким пороговим значенням для визначення нейронного виходу. Якщо сума є більшою за задане порогове значення, елемент обробки генерує сигнал, а якщо менше порогового значення, сигнал не генерується. У випадку функції порогових значень, будь-які значення, що перевищують або дорівнюють даному пороговому значенню, перетворюються в 1, тоді як усе, що падає нижче, перетворюється в 0 під час активації [9].

б) Функція Сигмоїди

Вихід змінюється постійно, але не лінійно, оскільки вхід змінюється, що

представлено на функції Сигмоїди на рисунку 3.2 [9].

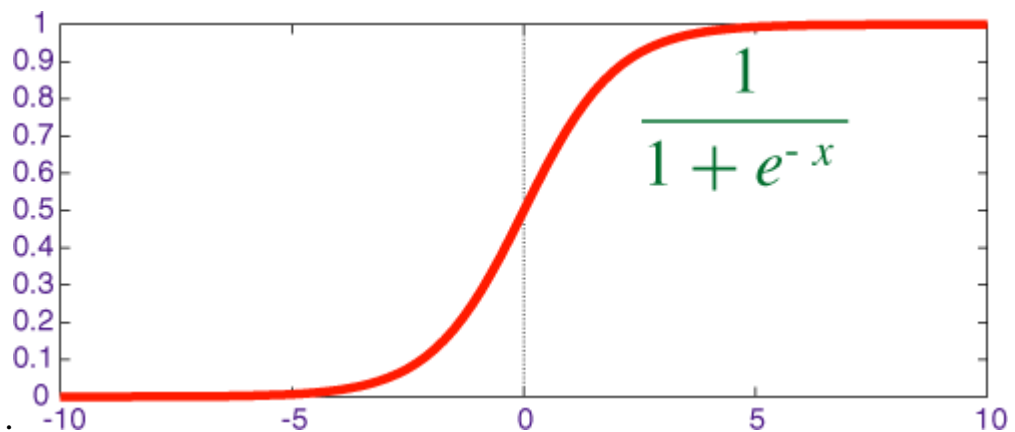


Рисунок 3.2 – Функція Сигмоїди [9]

3.3 Навчання ANN

Після структуризації нейронної мережі, вона є готовою до процесу навчання, який являє собою регулювання ваг з'єднання. На початку даного процесу зв'язкам присвоюються ваги з випадковим значенням. Зазвичай існує два підходи до навчання, які будуть розглянуті нижче.

3.3.1 Навчання з учителем

При навчанні з учителем дані для навчання отримуються як і на вхід, так і на вихід. Отже, навчальна мережа має вхідне значення, вагу та очікуване вихідне значення. Протягом даного процесу мережа обробляє вхідні дані та генерує виходи, потім отримані результати порівнюються з бажаними, що надаються на початку процесу навчання. Помилки, отримані після порівняння, пізніше передаються назад до системи, змушуючи останню регулювати ваги, які контролюють мережу. Цей процес повторюється безперервно, доки не отримуються бажані результати [3].

3.3.2 Навчання без учителя

Протягом процесу навчання без учителя, мережа отримує лише вхідні дані, без вихідних. Система сама повинна вирішити, які функції вона

використовуватиме для групування вхідних даних. Системі надається навчальний набір, а метою її роботи є категоризація або виявлення особливостей або закономірностей у даних для навчання. Мережа внутрішньо контролює їх роботу і відповідно регулює ваги [3].

3.4 Коефіцієнт швидкості навчання (Learning rate)

Задачею коефіцієнта швидкості навчання є модифікація ваги з'єднання на входах кожного оброблювального елемента. Сам процес модифікації ваги вхідного з'єднання з метою досягнення бажаного результату можна назвати функцією адаптації, а також режимом навчання.

Якщо даний коефіцієнт низький, тоді саме навчання є більш надійним, але оптимізація займе багато часу, оскільки кроки до мінімуму функції втрат є незначними.

Якщо ж даний коефіцієнт високий, то навчання може не збігатися і навіть не розходитися. Зміни ваги можуть бути настільки великими, що оптимізатор перевищує мінімум і погіршує втрати.

Навчання слід розпочинати з відносно великим темпом навчання, оскільки спочатку випадкове значення ваг далеко не оптимально, і тоді швидкість навчання може зменшуватися під час тренувань, щоб забезпечити більш дрібне оновлення ваги [9].

3.3 Обґрунтування методів розв'язання

3.3.1 Штучні нейронні мережі (Artificial Neural Networki)

Штучні нейронні мережі є непараметричними класифікаторами.

Структура штучних нейронних мереж імітує процес функціонування нервової системи людини. Найважливішою одиницею цього типу мереж є уніфікований процесор обробки, який має назву нейрон. Кожен нейрон має дві

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

фази- тренування та використання [9]. На фазі тренування нейрон вчиться як саме виконувати операцію, а вже на етапі тестування використовує інформацію з фази навчання для прогнозування результату. Зазвичай, дані нейронні мережі застосовуються для виявлення особливих тенденцій або закономірностей у даних.

Штучні нейронні мережі мають ряд переваг, такі як [10]:

- висока швидкість обчислення;
- можливість ефективно справлятися з шумами, що потрапляють на вхід;
- базування на інформації, отриманої під час навчання.

Хоча ці штучні нейронні мережі популярні, вони мають деякі недоліки [10]:

- процес навчання є трудомістким, оскільки вимагає ґрунтовної попередньої підготовки;
- вважається семантично бідним.

3.3.2 Рекурентні нейронні мережі (Recurrent Neural Network):

Рекурентні нейронні мережі (RNN) містять вихідні дані обробних вузлів і передають результат обратно до моделі. Кожен вузол у RNN діє як комірка пам'яті, яка продовжує обчислення та реалізацію операцій. Якщо передбачений мережею результат є неправильним, система само навчається і продовжує працювати над покращенням прогнозування під час зворотного розповсюдження [11].

Переваги:

- RNN запам'ятовує елементи інформації в часі. При прогнозуванні часових рядів є корисним запам'ятати попередні введені дані, лише завдяки функції. Це зветься довгостроковою короткочасною пам'яттю.

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

- рекурентні нейронні мережі можуть також використовуватися із шарами згортки для збільшення ефективного сусідства пікселів [11].

Недоліки:

- градієнтні проблеми зникнення та вибуху;
- навчання RNN є доволі складною задачею;
- відсутність можливості оброблювати дуже довгі послідовності, якщо використовує деякі функції активації [11].

3.3.3 Метод опорних векторів (Support Vector Machines)

Метод опорних векторів, також відомий під назвою SVM, є класифікатором непараметричним. SVM - це класифікатор двійковий, який поділяє класи за рахунок лінійної межі. Цей класифікатор передбачає відсутність будь-якої попередньої інформації про класифікацію даних. Це є оптимальним рішенням для використання навчальних даних, що являє собою найважливішу перевагу цього класифікатора перед іншими, такими як класифікатор максимальної ймовірності [12]. Справжня потужність SVM полягає в зображенні ядра, бо вона полегшує нелінійне відображення вхідного простору до простору ознак [13]. Через це, вибір функції ядра є найважливішим кроком у підтримці SVM. Деякі з найбільш часто використовуваних функцій ядра:

- лінійне ядро;
- поліноміальне ядро;
- Гаусове ядро.

Оскільки SVM оптимізує використання навчальних даних, це значно збільшує швидкість класифікатора. Це також мінімізує помилки класифікації, які могли б виникнути внаслідок попередніх припущень щодо некерованих даних [12].

До переваг SVM входять:

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

- можливості узагальнення;
- відсутність з проблемою переобладнання;
- використання нелінійного перетворення.

Недоліками цього класифікатора є:

- складність структури алгоритму і через це важкість її розуміння;
- неможливість легко визначити оптимальні параметри.

3.3.4 Згорткові нейронні мережі (Convolutional neural networks)

Згорткові нейронні мережі (CNN) - одна з найпопулярніших моделей, що використовуються сьогодні. Ця обчислювальна модель нейронної мережі використовує варіацію багатошарових персептронів і містить один або кілька згорткових шарів, які можуть бути повністю пов'язані або об'єднані. Ці згорткові шари створюють карти функцій, які фіксують область зображення, яка в підсумку розбивається на прямокутники та надсилається для нелінійної обробки[11].

Переваги:

- доволі висока точність при розпізнаванні зображень;
- автоматичність виявлення важливих характеристики без будь-якого людського нагляду;
- Поділ ваги[11].

Недоліки:

- Відсутність функціоналу для кодування положення та орієнтацію об'єкта;
- відсутність можливості бути просторово інваріантною до вхідних даних;
- необхідність великої кількості навчальних даних[11].

CNN розглядається як базова технологія, яка буде використана в даній роботі саме через її відомість та ефективність у галузі машинного навчання в

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

останній час, а особливо через забезпечення значних успіхів у різних галузях розпізнавання рукописного вводу, таких як розпізнавання цифр, розпізнавання символів, розпізнавання слів та перевірка підпису .

3.4 Опис методу розв'язання

Загалом, CNN є свого роду моделлю з ієрархію, вхідна інформація якої включає в себе необроблені набори даних, такі як RGB-зображення, необроблені аудіо дані, тощо. CNN виокремлює високорівневу семантичну інформацію з вхідних даних шляхом пошарового стекування операцій згортки, операцій об'єднання та відображення нелінійних функцій активації, що називається "зворотна передача". Різні типи операцій частіше за все називають "шарами" в контексті згорткової нейронної мережі: операція згортки відповідає "шару згортки", операція об'єднання відповідає "шару об'єднання", і тому подібне. Врешті-решт, останній шар згорткової нейронної мережі формалізує цільові задачі (класифікація, регресія тощо) на цільові функції. Обчислюючи похибку або втрату між очікуваним значенням і реальним, алгоритм зворотного розповсюдження повертає помилку або втрату з останнього шару пошарово, оновлює параметри кожного шару та подає - пересилання знову після оновлення. До тих пір, поки модель мережі не сходиться, мета навчання моделі не буде досягнута [1].

3.4.1 Шар згортки (Convolution Layer)

З математичної точки зору, згортка (convolution) - це функція, отримана інтегруванням двох заданих функцій. Вона демонструє, як форму однієї функції модифікує інша[1]:

$$s(t) = \int x(a)w(t - a) da \quad (3.9)$$

Операція згортки зазвичай позначається зірочкою:

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

$$s(t) = (x * w)(t), \quad (3.10)$$

де x означає вхід шару згортки, а аргумент w - ядро.

Результатом є сформована карта об'єктів [14].

При практичному застосуванні, вхідними даними моделі глибокого навчання є частіше за все, багатовимірний масив, а само ядро являє собою також багатовимірний масив з параметрами, які піддаються навчанню. Прийнято позначати ці багатовимірні масиви тензорами. У типовій моделі CNN частіше за все треба виконувати згортку більше ніж на одній осі одночасно. Наприклад, при використанні двовимірного зображення I як вхідний сигнал, ми, вірогідніше за все, також схильні використовувати двовимірне ядро (kernel) K [14]:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (3.11)$$

Згортка (Convolution) є комутативною, тобто ми можемо рівнозначно написати:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (3.12)$$

Різні ядра, що містяться в шару згортки, певним чином інтерпретуються як детектори ознак. У великої кількості випадків розглянуті мережею унікальні ознаки будуть непомітні для людського ока, саме через це згорткові нейронні мережі є такими ефективними. Маючи достатню підготовку, вони мають можливість допомогти людині при виконанні великої кількості задач з точки зору обробки та класифікації зображень [1].

3.4.2 Шар об'єднання (Pooling Layer)

Незважаючи на те, що шар згортки мінімізує масштаб зображення під час вилучення ознак, його вихід все одно фіксує точне положення об'єктів на

вході. Завдяки цьому обмеженню, навіть невеликий набір позицій об'єкта на вхідному зображенні призведе до відповідних змін відображення об'єктів, таких як повторне обрізання, обертання, зсув та інші операції із зображенням [1].

Кажучи про процес обробки сигналів загальноприйнятим методом вирішення цієї проблеми, який має назву «вибірка вниз» (down sampling), можна стверджувати, що він полягає у збереженні версії вхідного сигналу з низькою роздільною здатністю, яка все ще містить важливі структурні елементи, відкидаючи деталі, що не мають значення для об'єкта. Вибірка вниз може бути досягнута за допомогою шарів згортки, змінивши по всьому зображенню крок згортки. Більш надійним та ефективним підходом є побудова шару об'єднання. Пул об'єднання - це новий шар, доданий після шару згортки. Зокрема, після того, як нелінійність (наприклад, ReLU) застосована до карт об'єктів, що виводяться шаром згортки операція об'єднання визначається вручну, а не під час навчання. Дві загальні функції, що використовуються в операції об'єднання [1]:

1. Середнє об'єднання (Average Pooling): Середнє значення всіх елементів усередині вікна вибірки приймається як результат об'єднання (3.5)[15].

$$f(x) = \frac{1}{k*k} \sum_{i=i_1 \dots i_{l+k}} \sum_{j=j_1 \dots j_{l+k}} x_{i,j} \quad (3.13)$$

2. Максимальне об'єднання (Max Pooling): Обрання максимального значення всередині вікна вибірки як результат об'єднання (3.14) [1].

$$f(x) = \max (x_{[i,i+k],[j,j+k]}) \quad (3.14)$$

Введення шару об'єднання полягає у зменшенні розмірності (вибірка вниз) та абстрактних візуальних об'єктів введення шляхом імітації зорової системи людини.

3.1.5 Функція активації

Шар функції активації також називають шаром відображення нелінійності. Як випливає з назви, функція активації запроваджена для збільшення виразної здатності всієї мережі (тобто нелінійності). В іншому випадку укладання декількох лінійних операційних шарів може грати лише роль лінійного відображення, а не формалізації складних функцій. На практиці можна вибрати до десятка функцій активації, таких як Sigmoid, $\tanh(x)$, ReLU, Leaky ReLU, Parametric ReLU тощо [1].

Інтуїтивно, активаційна функція імітує характеристики біологічних нейронів: приймає набір вхідних сигналів і генерує вихід. У нейронауці біологічні нейрони зазвичай мають поріг. Коли кумулятивний ефект вхідних сигналів, отриманих нейронами, перевищує цей поріг, нейрони активуються і переходять у збудливий стан; в іншому випадку вони перебувають у гальмівному стані. У штучних нейронних мережах, оскільки функції сигмоїдного типу (показано на рис. 3.2) можуть добре імітувати цей біологічний процес, вони відіграли важливу роль у розвитку нейронних мереж [1].

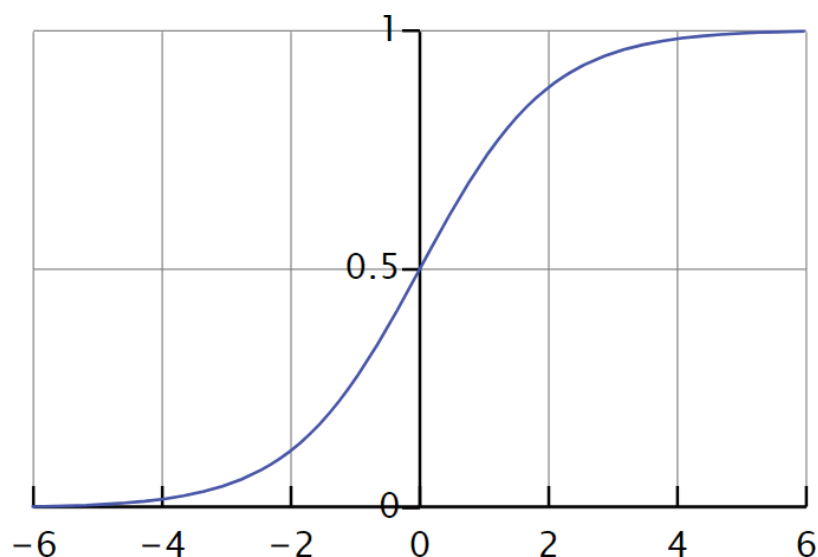


Рисунок 3.2 – Функція Сигмоїди [16]

Сигмоїдну функцію також називають логістичною функцією:

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

$$\delta(x) = \frac{1}{1+e^{-x}} \quad (3.15)$$

Як показано на рисунку 3.2, після процесу функції Сигмоїди діапазон вихідної реакції стискається до $[0,1]$, тоді як 0 відповідає "гальмівному стану" біологічних нейронів, а 1 відповідає "збудливому держава ". Однак подальше спостереження може також виявити, що на обох кінцях функції сигмоїда, яким би великим (або малим) не було вхідне значення, вихід буде стиснутий до 1 (або 0) для значення більше 5 (або менше, ніж 5) вводу. Це призводить до серйозної проблеми, а саме до ефекту насичення градієнтів. Відповідно до зміни градієнта сигмоїдної функції, градієнт деталі більше 5 (або менше, ніж 5) близький до 0, що ускладнить або навіть ускладнить перенесення помилки (похідна якої знаходиться у двох згаданих вище областях) на фронтальний шар у процесі зворотного поширення, а також спричиняє збій навчання в мережі (коли результат 0, параметри мережі не можуть бути оновлені). Крім того, особливу увагу слід приділити ініціалізації параметрів, щоб уникнути безпосереднього перенесення вихідних параметрів вихідного діапазону в ці регіони: один із можливих випадків полягає в тому, що коли параметри ініціалізації занадто великі, ефект насичення градієнта буде безпосередньо спрацьовує, і навчання не може бути проведене [1].

Щоб уникнути ефекту насичення градієнтом, Нейр та Хінтон впровадили випрямлений лінійний блок (ReLU) у нейронну мережу у 2010 р. [17]. Функція ReLU є однією з найбільш часто використовуваних функцій активації в нейронних мережах із глибокою згорткою.

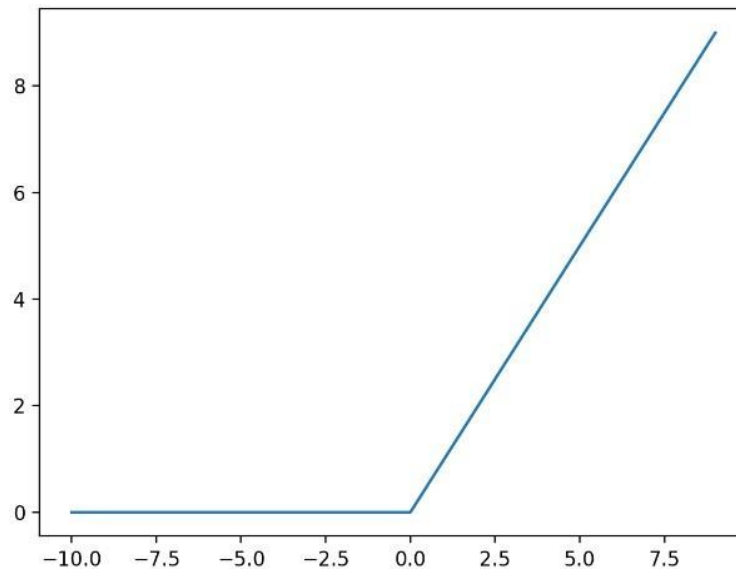


Рисунок 3.3 – Функція ReLU[1]

Насправді функція ReLU є різновидом поштучної функції (Piecewise Function) (показаної на малюнку 3.4), яка визначається як[1]:

$$\text{rectifier}(x) = \max\{0, x\} = \begin{cases} x, & \text{якщо } x \geq 0 \\ 0, & \text{якщо } x \leq 0 \end{cases} \quad (3.16)$$

Як показано на рисунку 3.4, градієнт функції ReLU дорівнює 1, коли x більше або дорівнює 0, що повністю виключає ефект насичення. В іншому випадку градієнт дорівнює 0. У той же час, функція ReLU, було доведено, що функція ReLU приблизно в 6 разів швидша, ніж Sigmoid, у методі збігу випадкового зменшення градієнта. Завдяки цим чудовим характеристикам, функція ReLU стала найкращим вибором для згорткової нейронної мережі та іншої моделі глибокого навчання [18].

3.1.6 Повністю підключений шар

Повністю пов'язаний шар діє як "класифікатор" у всій згортковій нейронній мережі. Це можна було б охарактеризувати так, що згортковий шар, шар об'єднання та шар функції активації відображають вихідні дані до простору об'єктів прихованих шарів, тоді повністю зв'язаний шар відіграє роль відображення вивченого «розподіленого представлення об'єкта» до

простору міток зразків. На практиці повністю підключений шар може бути реалізований за допомогою операції згортки: повністю з'єднаний шар, фронтальний шар якого є також повністю пов'язаним шаром, може бути перетворений в шар згортки з ядром 1×1 ; і повністю зв'язаний шар, лицьовий шар якого є конволюційним шаром, може бути перетворений у глобальну згортку з ядром $H \times W$, H і W - це висота та ширина вихідних даних із попередніх шарів відповідно.

Висновок до розділу

Протягом написання розділу математичного були сформульовані змістовна та математична постановки задачі, проаналізовані існуючі методи розв'язання даної задачі та обрану один для використання у даному дипломному проєкті.

Було описано задача класифікації зображень чисел для процесу автоматизації перевірки робіт, відповіді яких подаються у вигляді рукописної числової інформації в якості змістовної постановки задачі.

У частині з математичною постановкою задачі було представлено короткий опис нейронних мереж, продемонстровано структуру нейрона, та описано процеси, які відбуваються під час навчання нейронної мережі. Також було описано алгоритм навчання звичайної нейронної мережі.

Було представлено різні методи розв'язання даної задачі, виявлені їх переваги та недоліки та був обраний та описаний один метод, який буде використовуватись у даному дипломному проєкті, а саме метод з застосуванням згорткових нейронних мереж (CNN) та було описано чому саме був обраний саме цей метод.

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

До основного стеку технологій, використаного для реалізації даного програмного забезпечення, входять мова програмування Python, бібліотеки Keras та TensorFlow для роботи з моделями нейронних мереж та технологіями машинного навчання, бібліотека OpenCV для роботи з обробкою отриманого зображення бланку з відповідями та бібліотека Streamlit для створення інтерфейсу для користувача. Також, у якості допоміжних бібліотек, було використано такі з них, як numpy, matplotlib та pickle. Розробка продукту виконувалася в програмному середовищі – PyCharm.

Python [19] – це інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Високорівневі вбудовані структури даних у поєднанні з динамічним набором тексту та динамічним прив'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання в якості мови сценаріїв або склеювання для з'єднання існуючих компонентів разом. Простий, легкий у засвоєнні синтаксис Python підкреслює читабельність і, отже, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній або двійковій формі безкоштовно для всіх основних платформ і можуть вільно поширюватися [19].

Нижче представлений список бібліотек з їх описом, що підтримуються Python та які використовуються у даному дипломному проєкті:

- 1) TensorFlow – це комплексна платформа з відкритим вихідним кодом для машинного навчання. Він має всеосяжну гнучку

екосистему інструментів, бібліотек і ресурсів спільноти, яка дозволяє дослідникам просувати новітні досягнення в області машинного навчання, а розробникам легко створювати і розгортати додатки на основі машинного навчання. Можливість легкого створення та навчання моделей машинного навчання за допомогою інтуїтивно зрозумілих високорівневих API, таких як Keras, з активним виконанням, що забезпечує простоту налагодження, можна віднести до головних переваг використання бібліотеки у даному дипломному проєкті[20].

- 2) Keras – це бібліотека глибокого навчання, написана на Python, що працює на платформі машинного навчання TensorFlow. Вона була розроблена з акцентом на швидкі експерименти. Можливість якнайшвидшого переходу від ідеї до результату є ключовим фактором для проведення хороших досліджень. Це високорівневий API TensorFlow 2: доступний, високопродуктивний інтерфейс для вирішення проблем машинного навчання, зосереджений на сучасному глибокому навчанні. Він забезпечує основні готові рішення для розробки та транспортування рішень для машинного навчання з високою швидкістю проходження ітерацій. [21].
- 3) OpenCV (Open Source Computer Vision Library) – це бібліотека з відкритим кодом для комп'ютерного зору та машинного навчання. OpenCV була побудована для забезпечення загальної інфраструктури для програм комп'ютерного зору та прискорення використання машинного сприйняття в комерційних продуктах. Будучи продуктом із ліцензією BSD, OpenCV полегшує бізнесу використання та модифікацію коду. [22].
- 4) Streamlit – це бібліотека Python з відкритим кодом, яка спрощує створення веб-інтерфейсу для програмних забезпечень у галузях для

машинного навчання та науки про дані.

- 5) Matplotlib – це двовимірний Python бібліотека для побудови графіків, яка створює якісні зображення у багатьох інтерактивних середовищах та форматах друкованих копій. Matplotlib можна використовувати в сценаріях Python, оболонках Python та IPython, блокнотах Jupyter, серверах веб-додатків т.п. [23].
- 6) NumPy – це фундаментальний пакет для наукових обчислень за допомогою Python. Він містить такі речі: корисний N-мірний об'єкт масиву, інтеграція інструментів для декількох програмних мов, технологія мовлення, лінійна алгебра, перетворення Фур'є та можливості генерації випадкових чисел. Крім того, NumPy може також використовуватись як багатовимірний контейнер загальних даних з довільними типами даних. Це дозволяє NumPy ідеально та зручно інтегруватися з різними базами даних [24].
- 7) Pickle – це модуль, який реалізує потужний алгоритм серіалізації і десеріалізації об'єктів Python. "Pickling" - процес перетворення об'єкта Python в потік байтів, а "unpickling" - зворотна операція, в результаті якої потік байтів перетворюється назад в Python-об'єкт. Так як потік байтів легко можна записати в файл, модуль pickle широко застосовується для збереження і завантаження складних об'єктів в Python. [25].

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Даний програмний продукт представлений у вигляді веб-додатку, який запускається на локальному сервері.

Мінімальні вимоги до характеристик компонентів технічного

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

забезпечення, при яких значення часових параметрів Системи повинні відповідати пред'явленим в ТЗ вимогам:

- процесор з тактовою частотою не нижче 2 ГГц;
- остатній об'єм оперативної пам'яті (не менше 2 ГБ);
- Python 3.7;
- Tensorflow 2.0.0, keras 2.3.1, matplotlib 3.1, numpy 1.18, opencv 3.4,
- scikit-learn 0.22;

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Розроблений програмний продукт складається з шести класів: Клас Веб застосунку (Main_Interface), клас Обробка бланку (Form_Processing), клас Тренування (Train), клас Числа (MultiDigits), клас Виокремлення фрагментів (RegionSelector), та клас Обрахування оцінки (Mark_estimation).

Кожний з класів реалізує наступний функціонал:

- Клас Веб-застосунку (Main_Interface):
 - 1) Створення функціоналу для користувацького інтерфейсу.
 - 2) Запуск системи.
- Клас Обробка бланку (Form_Processing):
 - 1) Попередня обробка зображень бланку.
 - 2) Виявлення фрагментів бланку, звідки буде зчитуватись інформація.
 - 3) Виявлення об'єктів для класифікації.
 - 4) Збереження результатів класифікації до списку чисел (відповідей з бланку).
- Клас Тренування (Train):
 - 1) Створення моделі згорткової нейронної мережі (CNN).

- 2) Тренування моделі згорткової нейронної мережі (CNN).
- Клас Числа (MultiDigits):
 - 1) Створення функціоналу для можливості класифікації одразу декількох цифр з виділеного фрагменту.
 - 2) Клас Виокремлення фрагментів (RegionSelector):
 - 3) Виділення фрагментів з екземпляра бланку, з яких буде зчитуватись інформація.
- Клас Обрахування оцінки (Mark_estimation):
 - 1) Обчислення кінцевої оцінки, ґрунтуючись на правильних відповідях, та відповідях, отриманих з бланку.

Дана діаграма класів наведена у частині графічного матеріалу.

4.3.2 Діаграма послідовності

Розглянемо структурну схему діаграми послідовності, яка представлена у частині графічного матеріалу. Користувач має можливість завантажити до веб-застосунку системи два файли: файл з правильними відповідями та файл з фотографією заповненого бланку з відповідями, після чого файли передаються до серверу системи, де у першу чергу виконується запит на отримання списку з правильними відповідями (правильні відповіді подаються у печатному форматі), після чого даний список буде виведений на головний екран веб-застосунку, потім виконується запит на обробку отриманого зображення (покращення його якості, прибирання зайвих елементів з зображення, його центрація, поворот та інші), де фотографія заповненого бланку передається у відповідну функцію. Після чого виконується запит на виокремлення фрагментів зображення бланку, на яких зображені відповіді і відповідно на яких буде відбуватися процес розпізнавання чисел. Наступним етапом саме і є процес розпізнавання чисел з виокремлених фрагментів зображення та збереження отриманих результатів у масив з розпізнаними відповідями з бланку, після чого отриманий список розпізнаних чисел буде виведений на

					ДП 7233.00.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

головному екрані веб-застосунку, та це буде першим суттєвим результатом, який побачить користувач. Далі, отриманий список розпізнаних чисел та список з правильними відповідями, отриманий з другого завантаженого користувачем файлу, будуть передаватись у функцію для обрахування кінцевої оцінки, де і буде отримана кінцева оцінка. Останнім кроком буде виведення отриманої оцінки на головному екрані веб-застосунку, що буде другим і останнім суттєвим результатом, який побачить користувач системи.

4.3.3 Діаграма компонентів

Розглянемо структурну схему діаграми компонентів, представлену у частині графічного матеріалу.

З діаграми можна побачити, що система містить такі компоненти: Головний (Веб-Застосування) (Main(Web-Application)), Модель навчання (Trained Model), Обробка зображення бланку (Blank Processing), Обробка фрагментів з числами (Fragments with digits Processing), та Обрахунок оцінки (Mark Estimation).

- Компонент Головний (Веб-Застосування) (Main(Web-Application)) слугує оболонкою для функціонування роботи веб-застосування та відправляє заклики іншим компонентам, отримує результати і виводить на головний екран.
- Компонент Модель навчання (Trained Model) отримує зображення з числом, виконує його класифікацію до певного числа та повертає результат до компоненту Обробка фрагментів з числами (Fragments with digits Processing).
- Компонент Обробка зображення бланку (Blank Processing) отримає на заклик зображення бланку, що відправив користувач на попередню обробку та виокремлення його окремих фрагментів з зображеннями чисел та передає отримані фрагменти до компоненту Обробка фрагментів з числами (Fragments with digits Processing) їх подальшої

обробки.

- Компонент Обробка фрагментів з числами (Fragments with digits Processing) отримує масив з виокремлених фрагментів з зображення бланку, виконує підготовку до передачі зображень на класифікацію, а саме, робить можливим розпізнавання великої кількості цифр з одного зображення.
- Компонент Обрахунок оцінки (Mark Estimation) отримує два списки чисел : з правильними відповідями і відповідями з бланку та їх основі обраховує оцінку та передає результат до Головного компоненту (Веб-Застосування) (Main(Web-Application)).

4.3.4 Специфікація функцій

Опишемо специфікацію функції класів програмного забезпечення.

Детальний опис усіх функцій представлено у таблиці 4.1

Таблиця 4.1 – Специфікація функцій програмного забезпечення

Назва класу	Опис
def myModel():	Функція, яка містить в собі функціонал для створення, тренування, тестування та перевірки результатів моделі згорткової нейронної мережі та збереження даних результатів. Повертає файл з результатами тренування мережі, який надалі використовується для процесу класифікації зображень чисел.

Продовження таблиці 4.1

Назва класу	Опис
<code>def printTrainingResults(model)</code>	Функція, яка приймає в якості аргументу результати навчання згорткової нейронної мережі та будує на їх основі графіки, які демонструють ефективність роботи створеної мережі.
<code>def image_processing(image)</code>	Функція, яка приймає в якості аргументу зображення фрагменту з числом з заповненого бланку з відповідями та містить в собі функціонал для створення можливості розпізнавання та класифікації одразу декількох цифр (одного числа) та їх збереження у вигляді однієї відповіді з бланку.
<code>def mousePoints():</code>	Функція, яка містить в собі функціонал для позначення на зразку пустого бланку для відповідей фрагментів бланку, над якими буде проводитись обробка та класифікація.

Продовження таблиці 4.1

Назва класу	Опис
def calculate_mark(blank_answers, right_answers)	Функція, яка приймає в якості аргументів список правильних відповідей та відповідей з заповненого бланку, та містить в собі функціонал для обрахування кінцевої оцінку, у якості відсотку відповідей з бланку, які співпадають з правильними. Повертає обраховану оцінку.
def blank_preprocess(img)	Функція, яка приймає в якості аргументу зображення заповненого бланку з відповідями та містить в собі функціонал для попередньої обробки зображення бланку.
def blank_crop(img,num)	Функція, яка приймає в якості аргументів зображення заповненого бланку з відповідями та кількість правильних відповідей з бланку з правильними відповідями та містить в собі функціонал для виокремлення та збереження фрагментів бланку, на яких зображені окремі відповіді. Повертає масив цих фрагментів.

Продовження таблиці 4.1

Назва класу	Опис
def answers_process(answers)	Функція, яка приймає в якості аргументу список розпізнаних чисел з фрагментів з одного бланку та містить в собі функціонал для їх обробки, а саме: додано функціонал, щоб усі числа були збережені як дробові.
def load_model()	Функція, яка містить в собі функціонал для завантаження навчальної моделі згорткової нейронної мережі.
def receive_answers_from_excel(file_right_answers)	Функція, яка приймає в якості аргументу файл з правильними відповідями у форматі XLS та містить в собі функціонал, який повертає список правильних відповідей з даного файлу.
def receive_blank_answers(file_blank,num):	Функція, яка приймає в якості аргументів файл з зображенням заповненого бланку з відповідями у форматі JPEG або PNG та кількість відповідей з файлу з правильними відповідями та містить в собі функціонал, який повертає список відповідей з даного файлу.
def run()	Функція, яка запускає систему

4.4 Опис звітів

Під час навчання згорткової нейронної мережі, що була створена в рамках даного програмного забезпечення. Результати її навчання та тестування зберігаються та виводяться у вигляді графіків (рис.4.1-4.2.), на яких можна побачити відсоток правильних відповідей під час тестування моделі в залежності від кількості прогонів тренування моделі (epoch). З отриманих графіків, можна побачити, що створена модель згорткової нейронної мережі показує доволі точні результати, так як відсоток правильних відповідей у процесі тестування досягає приблизно 99 (рис 4.3).

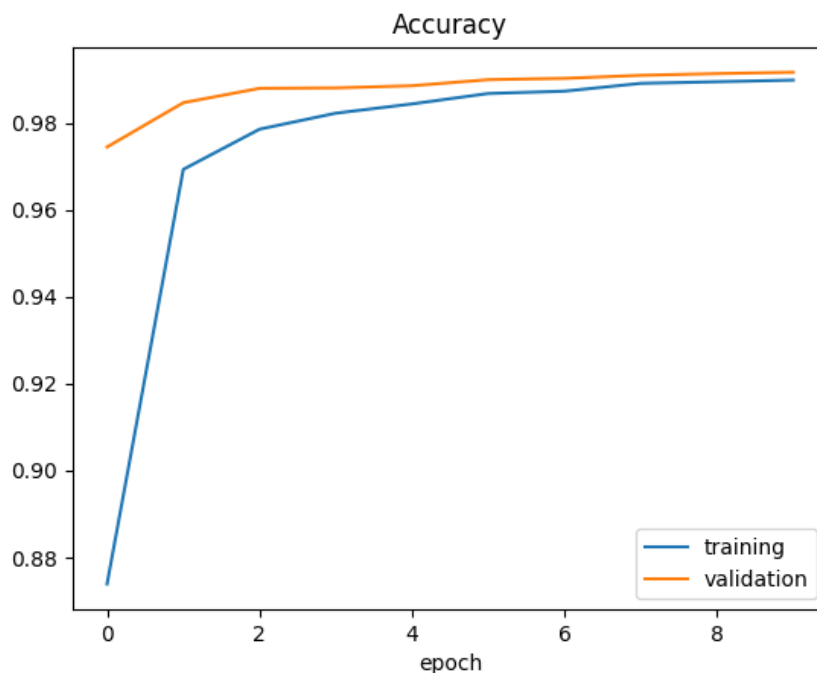


Рисунок 4.1 – Результати тестування точності моделі згорткової нейронної мережі

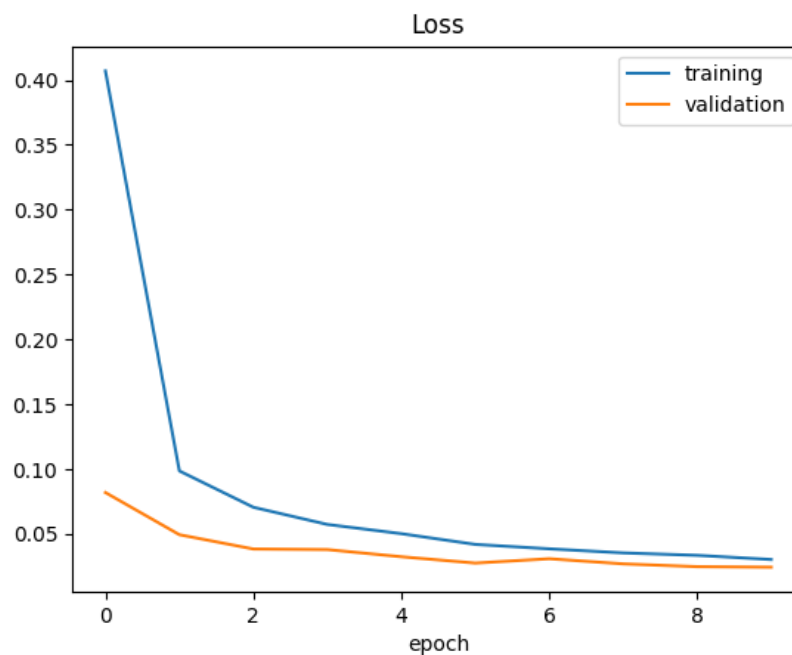


Рисунок 4.2 – Результати функції витрат моделі згорткової нейронної мережі

```
loss: 0.0304 - accuracy: 0.9899 - val_loss: 0.0245 - val_accuracy: 0.9917
```

Рисунок 4.3 – Результати точності та функції витрат для моделі згорткової нейронної мережі для останньому прогоні тестування (епосі)

Також нижче продемонстровані деякі з результатів роботи самого програмного застосунку(рис 4.4 – 4.6), для того щоб ще раз пересвідчитись в ефективності роботи моделі.

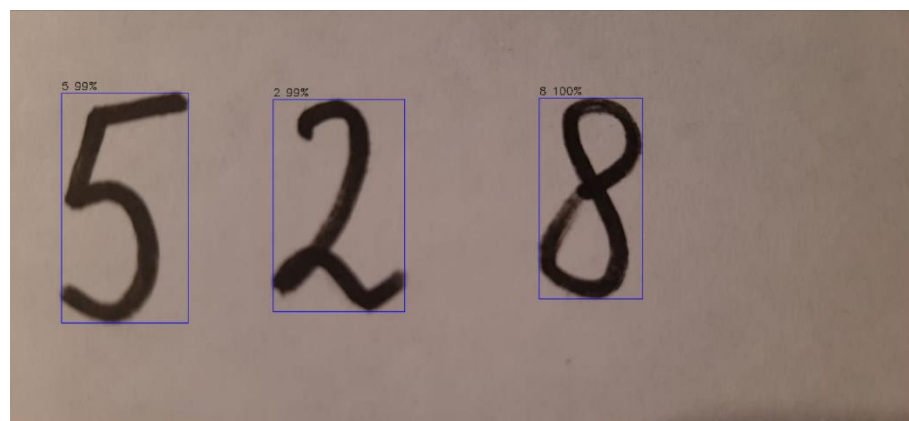


Рисунок 4.4 – Результати роботи програмного застосунку

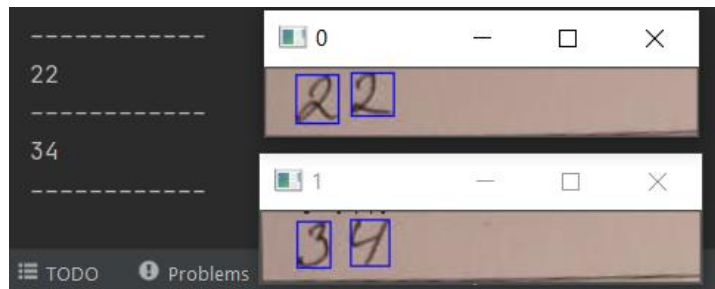


Рисунок 4.4 – Результати роботи програмного застосунку

Ідентифікатор		03
Відповідь 1	48	23
Відповідь 2	5473	91
Відповідь 3	463	80
Відповідь 4	110	15
Відповідь 5		
Відповідь 6		
Відповідь 7		
Відповідь 8		
Відповідь 9		
Оцінка		

Відповіді з бланку: ['603', 48.23, 5473.95, 463.8, 110.15, '', '', '', '', '', '', '', '', '']

Відсоток правильних відповідей студента 603 складає 0.0

Рисунок 4.5 – Результати роботи програмного застосунку

Висновок до розділу

Протягом написання розділу, присвяченого програмному та забезпеченню було описані технології, що використовувалися для створення програмного забезпечення – мова програмування Python та використані з нею бібліотеки.

Були виявлені та описані загальні вимоги до програмного забезпечення.

Було описано архітектуру програмного забезпечення та було спроектовано структурні схеми діаграми класів, послідовності та компонентів і наведено їх детальний опис. Також було наведено специфікацію функцій, що використовувалися при створенні програмного продукту та описано звіти, що генеруються в результаті роботи програми.

Було створено програмне забезпечення, що являє собою веб-застосунок для обробки зображення бланку з відповідями, виділення його фрагментів, над якими буде проводитись класифікація та перевірка отриманих відповідей з бланку, у вигляді розпізнаних рукописних чисел, базуючись на іншому завантаженому файлі з правильними відповідями.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Розглянемо функції користувача, які були реалізовані у програмному забезпеченні. Відповідно до розділу 1.1.2 «Опис функціональної системи», користувач має можливість виконувати наступні дії в рамках системи:

- завантажити до системи фотографію заповненого бланку: користувач робить фотографію бланку з відповідями студента та завантажує її до системи, а сама система у цей час спочатку розпізнає сам бланк з відповідями, потім розпізнає та зчитує відповідні числові символи(у вигляді відповідей студента);
- завантажити до системи файл з відповідями: для того щоб системи мала можливість виставити бали студенту, вона повинна мати дані, з якими треба порівняти числові відповіді, які були отримані з бланку;
- отримати оцінку одразу, без збереження її значення до файлу; користувач має можливість одразу після завантаження зображення та його обробки, побачити оцінку студента, без подальшого її збереження;
- перевірити правильність сканування оцінок: користувач отримує список з відповідями з бланку на головному екрані.

Під час розробки системи були реалізовані всі функції користувача.

Користувач на початку роботи з програмним застосунком має відкрити вікно браузера і перейти на <http://localhost:4200>. Після загрузки моделі користувач побачить головну сторінку, представлену на рисунку 5.1.

Перевірка відповідей

Завантажте файл з правильними відповідями



Drag and drop file here

Limit 200MB per file • XLSX

Browse files

Завантажте фотографію бланка з відповідями



Drag and drop file here

Limit 200MB per file • JPG, PNG

Browse files

Завантажте обидва файли для проведення перевірки

Рисунок 5.1 – Стартовий стан головної сторінки

Розглянемо детальніше головну сторінку користувача:

Кнопка 1: Завантаження файлу з правильними відповідями формату XSLX, що має мати наступну структуру (рис 5.2), а саме:

- 1) Перший рядок файлу має бути порожнім.
- 2) Перший стовпець файлу має мати нумерацію відповідей.
- 3) Другий стовпець має мати правильні відповідно до проставленої нумерації.

1	48.23		
2	5473.90		
3	463.8		
4	110.15		

Рисунок 5.2 – Структура файлу формату XSLX, що містить правильні відповіді

Усі інші елементи, що присутні у файлі, не будуть використовуватись.

Кнопка 2: Завантаження файлу з зображенням заповненого бланку з

відповідями формату JPG чи PNG, що має виглядати приблизно таким чином(рис 5.3).

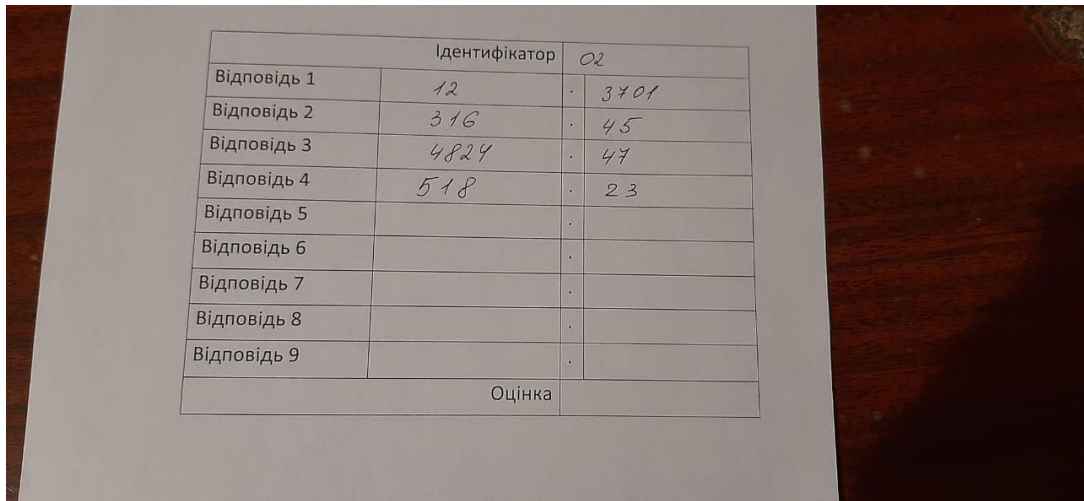


Рисунок 5.3 – Приклад файлу з зображенням заповненого бланку з відповідями формату JPG чи PNG

Шаблон порожнього бланку, куди будуть вноситись відповіді повинен обов'язково бути завжди однаковим і виглядати наступним чином (рис 5.4).

Ідентифікатор		
Відповідь 1		.
Відповідь 2		.
Відповідь 3		.
Відповідь 4		.
Відповідь 5		.
Відповідь 6		.
Відповідь 7		.
Відповідь 8		.
Відповідь 9		.
Оцінка		

Рисунок 5.4 – Шаблон порожнього бланку, куди будуть вноситись відповіді

Після того, як користувач завантажив обидва файла, відбувається усі процеси обробки та класифікації, що займуть певний час, після чого користувач побачить результати класифікації чисел з зображення бланку

на отриману оцінку, на основі порівняння списку відповідей з заповненого бланку та правильних відповідей (рис 5.5).

Ідентифікатор		01
Відповідь 1	25	29
Відповідь 2	37	38
Відповідь 3	845	71
Відповідь 4		
Відповідь 5		
Відповідь 6		
Відповідь 7		
Відповідь 8		
Відповідь 9		
Оцінка		

Відповіді з бланку: ['01', 25.29, 37.38, 845.71]

Відсоток правильних відповідей студента 01 складає 0

Рисунок 5.5 – Кінцевий результат роботи програмного застосунку

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань даного програмного застосунку є виявлення слабких місць у створеній системі та впровадження ефективних рішень

щодо їх усунення.

Необхідно розглянути всі функції які виконує система та описати очікувані результати відповідей функцій. Необхідно створити 5-10 тестів. Кілька тестів повинні приймати граничні значення.

Для кожного тесту необхідно викликати функцію та оцінити достовірність отриманих результатів.

В процесі тестування системи була перевірена уся її функціональність, а саме.

- Завантажити до системи фотографію заповненого бланку: користувач робить фотографію бланку з відповідями студента та завантажує її до системи, а сама система у цей час спочатку розпізнає сам бланк з відповідями, потім розпізнає та зчитує відповідні числові символи(у вигляді відповідей студента);
- Завантажити до системи файл з відповідями: для того щоб системи мала можливість виставити бали студенту, вона повинна мати дані, з якими треба порівняти числові відповіді, які були отримані з бланку;
- Отримати оцінку одразу, без збереження її значення до файлу: користувач має можливість одразу після завантаження зображення та його обробки, побачити оцінку студента, без подальшого її збереження;
- Перевірити правильність сканування оцінок: користувач отримує список з відповідями з бланку на головному екрані.

Також додатково, була перевірена функціональність завантаження бланку з різною кількістю заповнених відповідей.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

– ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Тести включають в себе перевірку функціональності роботи системи. Тестові сценарії з результатами їх проходження наведені у таблицях 5.1 – 5.4.

Таблиця 5.1 – Тестування виведення списку розпізнаних чисел з бланку

Тест:	Система виводить список розпізнаних чисел з завантаженого користувачем файлу PNG або JPG, що містить зображення бланку з відповідями.
Початковий стан системи:	Відкрита веб-сторінка застосунку.
Дія:	Користувач завантажив два файли: файл формату XLSX, що містить правильні відповіді та файл формату PNG або JPG, що містить зображення заповненого бланку з відповідями.
Очікуваний результат:	Виведення списку розпізнаних відповідей з заповненого бланку для відповідей.
Фактичний результат співпадає з очікуваним:	Так

Таблиця 5.2 – Тестування виведення списку чисел з файлу XLSX

Тест:	Система виводить список чисел з завантаженого користувачем файлу XLSX з правильними відповідями.
-------	--

Продовження таблиці 5.2

Початковий стан системи:	Відкрита веб-сторінка застосунку.
Дія:	Користувач завантажив два файли: файл формату XLSX, що містить правильні відповіді та файл формату PNG або JPG, що містить зображення заповненого бланку з відповідями.
Очікуваний результат:	Виведення списку правильних відповідей з файлу XLSX з правильними відповідями .
Фактичний результат співпадає з очікуваним:	Так

Таблиця 5.3 – Тестування виведення оцінки

Тест:	Система виводить оцінку, як відсоток співпадіння кожного числа зі списку розпізнаних відповідей до списку з правильними відповідями .
Початковий стан системи:	Відкрита веб-сторінка застосунку.
Дія:	Користувач завантажив два файли: файл формату XLSX, що містить правильні відповіді та файл формату PNG або JPG, що містить зображення заповненого бланку з відповідями.
Очікуваний результат:	Виведення сповіщення «Відсоток правильних відповідей студента <ідентифікатор студента> складає <обрахований відсоток>».

Продовження таблиці 5.3

Фактичний результат співпадає з очікуваним:	Так
---	-----

Таблиця 5.4 – Тестування можливості встановлення оцінки з різними кількостями відповідей для перевірки.

Тест:	Система виводить оцінку, як відсоток співпадіння кожного числа зі списку розпізнаних відповідей до списку з правильними відповідями, при цьому процес перевіряється ітераційно і кожного разу користувач завантажує файл формату XLSX з різною кількістю правильних відповідей.
Початковий стан системи:	Відкрита веб-сторінка застосунку.
Дія:	Користувач завантажив два файли: файл формату XLSX, що містить правильні відповіді та файл формату PNG або JPG, що містить зображення заповненого бланку з відповідями, при цьому користувач декілька разів повторює даний процес ітераційно і кожного разу користувач завантажує файл формату XLSX з різною кількістю правильних відповідей .
Очікуваний результат:	Кожного разу виконується виведення сповіщення «Відсоток правильних відповідей студента <ідентифікатор студента> складає <обрахований відсоток>».

Продовження таблиці 5.4

Фактичний результат співпадає з очікуваним:	Так
---	-----

Висновок до розділу

Протягом написання технологічного розділу було детально описано створений програмний продукт та представлені його екранні форми.

Було визначено мету проведення випробувань, представлені загальні положення. Було визначено, який саме функціонал має бути протестований.

У процесі тестування системи була перевірена уся її функціональність.

У ході тестування програмного продукту було встановлено, що функціонал розробленого веб-застосунку відповідає встановленим вимогам, які були встановлені при визначення функціональних границь системи.

ЗАГАЛЬНІ ВИСНОВКИ

Даний дипломний проєкт був присвячений вирішенню задачі розпізнавання, класифікації, подальшої обробки та збереження рукописної числової інформації.

Метою створення системи було спрощення процесу перевірки контрольних робіт студентів або учнів (де в якості відповідей маємо рукописні числові значення) за рахунок автоматизації цього процесу шляхом розпізнавання написаних студентами (учнями) відповідей у відповідному бланку для відповідей за допомогою моделей машинного навчання.

У розділі загальних положень було встановлено мету, цілі та задачі розробки, визначено функціональні границі системи та побудована структурна схема варіантів використання, проаналізовано існуючі аналоги та встановлено відмінність від них системи, що проектується.

У розділі інформаційного забезпечення надано детальний опис вхідних та вихідних даних, описано структуру масиву з інформацією, який використовується у даній системі.

Розділ математичного забезпечення присвячений опису змістовної та математичної постановки задачі, аналізу існуючих методів розв'язання задачі даного дипломного проєкту та обґрунтування вибору одного з них з його подальшим детальним описом.

Застосувавши саме CNN у даній роботі та провівши тренування та тестування мережі було доведено її ефективність, за допомогою результатів, які були отримані під час даних тестування та перевірки отриманих даних. Було виявлено, як суттєво змінюється точність визначення чисел та під час восьмого прогону тестування точність вже близьиться до максимальною, що підтверджує ефективність моделі CNN.

Також були продемонстровані результати класифікації зображень чисел, з застосуванням наведеного методу, на яких можна побачити те, як добре

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

система розпізнає де саме знаходяться числа, які саме ці символи та ймовірність того, що це саме те число.

Розділ програмного забезпечення описує засоби розробки програмного продукту та етапи проектування його архітектури. Описано специфікацію функцій та звіти, які генеруються в ході запуску програми.

У технологічному розділі визначено мету проведення випробувань програмного продукту та описано їх результати.

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

ПЕРЕЛІК ПОСИЛАНЬ

1. Mengqiao Zhao. Handwritten digit recognition based on segmentation-free method : Master of Science in Computer Science / Faculty of Computing, Blekinge Institute of Technology - Karlskrona, Sweden, 2020
2. Xuan Yang, Jing Pu. MDig: Multi-digit Recognition using Convolutional Nerual Network on Mobile / Stanford University - Stanford, CA
3. Akkireddy Challa. Automatic Handwritten Digit Recognition On Document Images Using Machine Learning Methods: Master of Science in Computer Engineering / Blekinge Institute Of Technology. Karlskrona, Sweden, 2019
4. Технические требования ABBYY FineReader 14 [Электронный ресурс]. Режим доступа : <https://web.archive.org/web/20170811183907/https://www.abbyy.com/ru-ru/finereader/tech-specs/>, 2017.
5. Yann LeCun, Corinna Cortes, Christopher J.C. Burges. The MNIST database of handwritten digits.
6. Artificial neural networks and other methods if image classification / M.Seetha, I.V. Muralikrishna, Member, IEEE B.L. Deekshatulu, Life Fellow, IEEE, B.L. Malleswari, Nagaratna, P. Hedge - K.: Journal of Theoretical and Applied Information Technology
7. Neural Network a Comprehensive Foundation; a Computational Approach to Learning and Machine Intelligence / S Haykin, - Macmillan, NY, 1994.
8. Chen W. Convolutional Neural Network for Image Classification / W. Chen,
9. X. Yang.. – (Johns Hopkins University).
- 10.Dave Anderson, George McNeill. Artificial neural networks technology / Kaman Sciences Corporation, Utica, New York
- 11.Difference between ANN, CNN and RNN [Электронный ресурс] //

- Jinfonyet Software, Inc.. – 2019. – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/difference-between-ann-cnn-and-rnn/>
12. Maryam Niknejad, Vahid Mirzaei Zadeh, Mehdi Heydari. Comparing different classifications of satellite imagery in forest mapping // International Research Journal of Applied and Basic Sciences – 2014
 13. Rajesh Sharma R Beaula A, Marikkannu P, Akey Sungheeth, C. Sahana. Comparative Study of Distinctive Image Classification Techniqu // 10th International Conference on Intelligent Systems and Control (ISCO) – 2016
 14. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning // MIT press - 2016.
 15. Brownlee Jason. A gentle introduction to pooling layers for convolutional neural networks. [Електронний ресурс] // machinelearningmastery.com – <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
 16. L. Martins, What is the sigmoid function, and what is its use in machine learning's neural networks? How about the sigmoid derivative function?. [Електронний ресурс] // Quora.com – Режим доступу до ресурсу: <https://www.quora.com/What-is-the-sigmoid-function-and-what-is-its-use-in-machine-learnings-neural-networks-How-about-the-sigmoid-derivative-function>
 17. Vinod Nair, Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines// In ICML - 2010.
 18. Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems - pages 1097–1105, 2012
 19. What is Python? Executive Summary. [Електронний ресурс] // Python.org – Режим доступу до ресурсу: <https://www.python.org/doc/essays/blurb/#:~:text=Python%20is%20an%2>

0interpreted%2C%20object,programming%20language%20with%20dynamic%20semantics.&text=Python's%20simple%2C%20easy%20to%20learn,program%20modularity%20and%20code%20reuse.

20. Комплексная платформа машинного обучения с открытым исходным кодом. [Электронный ресурс] // tensorflow.org – Режим доступа до ресурсу: <https://www.tensorflow.org>.
21. About Keras. [Электронный ресурс] // keras.io – Режим доступа до ресурсу: <https://keras.io/about/>.
22. About OpenCC. [Электронный ресурс] // opencv.org – Режим доступа до ресурсу: <https://opencv.org/about/>.
23. Hunter John, Dale Darren, Firing Eric, Droettboom Michael, and the Matplotlib development team. Matplotlib introduction. From <https://matplotlib.org/2.0.1/>.
24. The SciPy community. About numpy. From <https://numpy.org/devdocs/about.html>.
25. Pickle — Python object serialization. [Электронный ресурс] // Python.org – Режим доступа до ресурсу: <https://docs.python.org/3/library/pickle.html>.
26. Nupur Thakur, Deepa Maheshwari. A review of image classification techniques // International Research Journal of Engineering and Technology (IRJET) - Electronics and Telecommunication, Pune Institute of Computer Technology, Pune-Satara Road, Behind Bharati
27. Vidyapeeth College, Dhankawadi, Pune, Maharashtra, India, 2017
28. Кравець О.О. Задача розпізнавання рукописної числової інформації / О.О Кравець, М.О. Сперкач // Матеріали VI всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2021) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 22-23 квітня 2021 р.
29. The Nature of Statistical Learning Theory / V.N.Vapnik, (New York:

- Springer-Verlag), 1995.
30. Statistical Learning Theory (New York: Wiley) / V. N Vapnik, 1998.
31. A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery / C. J. C. Burges, 2, 1998, с. 121–167
32. R. K. Sundaram, A First Course in Optimization Theory, (New York: Cambridge University Press), 1996.
33. Introduction to Convolutional Neural Networks / Jianxin Wu. LAMDA Group. National Key Lab for Novel Software Technology. Nanjing University, China, 2017.
34. Kevin G. An introduction to neural networks / Gurney Kevin. – London and New York: Taylor & Francis e-Library, 2004. – 317 с. – (ROUTLEDGE). – (0-203-45151-1).
35. Sorokina K. Image Classification with Convolutional Neural Networks [Електронний ресурс] / Ksenia Sorokina // A Medium Corporation. – 201. – Режим доступу до ресурсу:
<https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>.
36. Chen W. Convolutional Neural Network for Image Classification / W. Chen, X. Yang. – (Johns Hopkins University).

Додаток А

Тексти програмного коду*Інформаційна система розпізнавання числової інформації*

(Найменування програми (документа))

DVD-R

(Вид носія даних)

12 арк, 8930 Кб

(Обсяг програми (документа) , арк., Кб)

Київ – 2021 року

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

```

import tensorflow as tf

import streamlit as st

import pickle

import cv2

from PIL import Image, ImageOps

import numpy as np

import Forms_Main

import pandas as pd

import marks_estimation

import MultiDigits

from threading import Thread

import keras.backend.tensorflow_backend as tb

tb._SYMBOLIC_SCOPE.value = True

st.set_option('deprecation.showfileUploaderEncoding', False)

@st.cache(allow_output_mutation=True)

def load_model():

    loaded_model = open("trained_model.p", "rb")

    model = pickle.load(loaded_model)

    return model

# t = Thread(target=load_model)

# t.start()

# t.join()

model = load_model()

st.write("

```


Перевірка відповідей

""

file_right_answers = st.file_uploader("Завантажте файл з правильними відповідями",

type=['xlsx'])

file_blank = st.file_uploader("Завантажте фотографію бланка з відповідями",

type=['jpg', 'png'])

def receive_answers_from_excel(file_right_answers):

right_answers = pd.read_excel(file_right_answers)

right_answers = right_answers.to_numpy()

only_right_answers = []

for el in right_answers:

only_right_answers.append(el[1])

return only_right_answers

def receive_blank_answers(file_blank,num):

image_blank = Image.open(file_blank)

image_blank = np.asarray(image_blank)

st.image(image_blank) # Image display

blank_answers = Forms_Main.blank_crop(image_blank,num)

return blank_answers

def error_check(blank_answers):

count1,count2, count3 = 0, 0, 0

user_answer = st.radio("Чи всі відповіді вірно розпізнані?",

('Так', 'Hi'), key=count1)

					ДП 7233.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

```

while user_answer == 'Hi':

    st.write('Вкажіть номер відповіді, яку розпізнано невірно')

    user_answer2 = int(st.number_input(label='Номер відповіді', step=1.0, key = count2))

    print(user_answer2, type(user_answer2))

    user_answer3 = st.text_input(label='Значення відповіді з бланку', key = count3)

    if user_answer2 is not None and user_answer3 is not None:

        blank_answers[user_answer2 - 1] = user_answer3

    count1 += 1

    count1 += 2

    count1 += 3

    user_answer = st.radio("Чи всі останні відповіді вірно розпізнані?",

                           ('Так', 'Hi'), key=count1)

    # user_answer = st.radio("Чи всі відповіді вірно розпізнані?",

    # ('Так', 'Hi'))

    return blank_answers

if file_blank is None or file_right_answers is None:

    st.text("Завантажте обидва файли для проведення перевірки")

else:

    right_answers = receive_answers_from_excel((file_right_answers))

    st.text("Правильні відповіді: " + str(right_answers))

    num_right_answers = len(right_answers)

    blank_answers = receive_blank_answers(file_blank, num_right_answers)

    st.text("Відповіді з бланку: " + str(blank_answers))

st.success('Відсоток правильних відповідей студента ' +

           str(blank_answers[0] + ' складає ')

```

```

+ str(marks_estimation.calculate_mark(blank_answers, right_answers))))

import cv2

import numpy as np

import pytesseract

import os

import MultiDigits

import streamlit as st

def blank_crop(img,num):

    per = 25

    roi = [(632, 24), (880, 62), 'int', 'id'],

          [(396, 70), (626, 106), 'int', 'ans11'],

          [(664, 68), (880, 108), 'int', 'ans12'],

          [(394, 116), (628, 150), 'int', 'ans21'],

          [(662, 116), (878, 150), 'int', 'ans22'],

          [(398, 158), (626, 196), 'int', 'ans31'],

          [(662, 158), (882, 194), 'int', 'ans32'],

          [(394, 206), (630, 242), 'int', 'ans41'], [(662, 206), (882, 242), 'int', 'ans42'],

          [(396, 248), (630, 288), 'int', 'ans51'], [(666, 250), (878, 284), 'int', 'ans52'],

          [(394, 294), (630, 332), 'int', 'ans61'], [(660, 294), (878, 328), 'int', 'ans62'],

          [(396, 336), (630, 376), 'int', 'ans71'], [(664, 338), (876, 376), 'int', 'ans72'],

          [(396, 380), (630, 422), 'int', 'ans81'], [(666, 380), (882, 422), 'int', 'ans82'],

          [(396, 426), (630, 462), 'int', 'ans91'], [(662, 424), (882, 464), 'int', 'ans92']]

```

```

answers = []

pytesseract.pytesseract.tesseract_cmd = 'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'

startImage = cv2.imread('Images/BlankTemplate4.png')

h,w,c = startImage.shape

##detector creating

orb = cv2.ORB_create(1000)

kp1, des1 = orb.detectAndCompute(startImage,None)#unique elements of image, the
representation of kp(easier to comp to understand and differenciate between them)

#imKp1 = cv2.drawKeypoints(startImage, kp1, None)

#cv2.imshow('KeyPoints', imKp1)

#path = 'Images/BlankTesting'

#myPicLlSt = os.listdir(path)

#for j, y in enumerate(myPicLlSt):

print("blank crop 1", type(img))

#img = cv2.imread(img)

print("blank crop 2", type(img))

#cv2.imshow(y, img)

kp2, des2 = orb.detectAndCompute(img,None)

bf = cv2.BFMatcher(cv2.NORM_HAMMING)

matches = bf.match(des2, des1) # matching the decreptors

matches.sort(key= lambda x :x.distance)#sort all the matches based on a distance (the lower
the distance, the better the match is)

good = matches [:int(len(matches)*(per/100))])#extract 25% of the best matches

```

```
#imgMatch = cv2.drawMatches(img,kp2,startImage,kp1,good[:20],None, flags =2)
```

```
#cv2.imshow("match", imgMatch)
```

```
# finding the relationships between start image and a test image
```

```
srcPoints = np.float32([kp2[m.queryIdx].pt for m in good]).reshape(-1, 1, 2)
```

```
dstPoints = np.float32([kp1[m.trainIdx].pt for m in good]).reshape(-1, 1, 2)
```

```
M, _ = cv2.findHomography(srcPoints, dstPoints, cv2.RANSAC, 5.0)
```

```
imgScan = cv2.warpPerspective(img, M, (w, h)) # align form
```

```
#cv2.imshow("Scan", imgScan)
```

```
st.image(imgScan)
```

```
imgShow = imgScan.copy() #final image
```

```
imgMask = np.zeros_like(imgShow)
```

```
myData = [] # data for each of our images
```

```
for x,r in enumerate(roi):
```

```
    cv2.rectangle(imgMask, ((r[0][0]), r[0][1]), ((r[1][0]), (r[1][1])), (0,255,0),cv2.FILLED)
```

```
    imgShow = cv2.addWeighted(imgShow, 0.99, imgMask,0.1, 0)
```

```
#Cropping
```

```
imgCrop = imgScan[r[0][1]: r[1][1], r[0][0]:r[1][0]]
```

```
answer = MultiDigits.image_processing(imgCrop) #string
```

```
#cv2.imshow(str(x) , imgCrop)
```

```
answers.append(answer)
```

```
#a = list(range(1,5))
```

```
a = list(range(1, num+1))
```

```
#b = list(range(1,10,2))
```

```
b = list(range(1, (num+1)*2, 2))
```

```
print(a)
```

```
print(b)
```

```
for i, j in zip(a,b):
```

```
    answers[i] = float('{0}.{1}'.format(answers[j], answers[j+1]))
```

```
    print(answers[i])
```

```
# answers[1]= float('{0}.{1}'.format(answers[1], answers[2]))
```

```
# answers[2] = float('{0}.{1}'.format(answers[3], answers[4]))
```

```
# answers[3] = float('{0}.{1}'.format(answers[5], answers[6]))
```

```
# answers[4] = float('{0}.{1}'.format(answers[7], answers[8]))
```

```
#answers[5] = float('{0}.{1}'.format(answers[9], answers[10]))
```

```
print("1 answers", answers)
```

```
#c = list(reversed(range(5,19)))
```

```
c = list(reversed(range(num+1, 19)))
```

```
print(c)
```

```
for i in c :
```

```
    answers.remove(answers[i])
```

```
# answers.remove(answers[10])
```

```
# answers.remove(answers[9])
```

```

# answers.remove(answers[8])

# answers.remove(answers[7])

# answers.remove(answers[6])

print("2 answers", answers)

st.image(imgShow)

#cv2.imshow('2', imgShow)

return answers

#return answers

#cv2.imshow(str(x) , imgCrop)

#print(f'{r[3]}:{pytesseract.image_to_string(imgCrop)}')

#print(answer)

#print('-----')

#myData.append(pytesseract.image_to_string(imgCrop))

#imgShow = cv2.resize(imgShow, (w // 3, h // 3))

#cv2.imshow(y + '2', imgShow)

#img = 'Images/BlankTesting/blank_test1.jpg'

#print(img.type)

#img = 'Images/BlankTesting/blank_test4.jpg'

#img = cv2.imread(img)

#print(blank_crop(img))

#cv2.imshow('Output', startImage)

```

```
cv2.waitKey(0)
```

```
def calculate_mark(blank_answers, right_answers):
```

```
    answers_amount = len(right_answers)
```

```
    mark = 100
```

```
    for i in range(answers_amount):
```

```
        if blank_answers[i+1] != right_answers[i]:
```

```
            print(blank_answers[i+1],right_answers[i])
```

```
            mark -= 100/answers_amount
```

```
            print(mark)
```

```
    if (mark < 0):
```

```
        mark = 0
```

```
    return mark
```

```
    import cv2
```

```
    import numpy as np
```

```
    from keras.datasets import mnist
```

```
    from keras.layers import Dense, Flatten
```

```
    from keras.layers.convolutional import Conv2D
```

```
    from keras.models import Sequential
```

```
    from keras.utils import to_categorical
```

```
    import matplotlib.pyplot as plt
```

```
    import pickle
```

```
    image = cv2.imread('Images/MultiDigitsTesting/testing3.jpg')
```

```
    def image_processing(image):
```

```
        loaded_model = open("trained_model.p", "rb")
```

```
        model = pickle.load(loaded_model)
```

```
        text_num = []
```

```
        #image = cv2.imread('./test5.jpg')
```

```
        gray = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)
```

```
        ret, th = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV +  
cv2.THRESH_OTSU)
```

```
        contours = cv2.findContours(th, cv2.RETR_EXTERNAL,  
cv2.CHAIN_APPROX_SIMPLE)[0]
```

```
        for cnt in contours:
```

```
            x, y, w, h = cv2.boundingRect(cnt)
```

```
            #print(w)
```

```
            if(w>=10 and h>=10 and w<=30):
```



```

        #print(w,h)
        cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 1)
        digit = th[y:y + h, x:x + w]
        resized_digit = cv2.resize(digit, (18, 18))
        padded_digit = np.pad(resized_digit, ((5, 5), (5, 5)), "constant",
constant_values=0)

        #print(padded_digit.shape)
        digit = padded_digit.reshape(1, 28, 28, 1)
        digit = digit / 255.0

        pred = model.predict([digit])[0]
        final_pred = np.argmax(pred)
        text_num.append([x, final_pred])

        data = str(final_pred) + ' ' + str(int(max(pred) * 100)) + '%'

        font = cv2.FONT_HERSHEY_SIMPLEX
        fontScale = 0.5
        color = (0, 0, 0)
        thickness = 1
        cv2.putText(image, data, (x, y - 5), font, fontScale, color, thickness)

    text_num = sorted(text_num, key=lambda t: t[0])
    text_num = [i[1] for i in text_num]
    final_text = "".join(map(str, text_num))
    return final_text

answer = image_processing(image)
#cv2.imshow(answer , image)
#print(answer)
#print(final_text)
#cv2.imshow('image', image)
cv2.waitKey(0)

import cv2
import random

path = 'Images/BlankTemplate4.png'
scale = 0.5
circles = []
counter = 0
counter2 = 0
point1=[]
point2=[]
myPoints = []
myColor=[]
def mousePoints(event,x,y,flags,params):
    global counter,point1,point2,counter2,circles,myColor
    if event == cv2.EVENT_LBUTTONDOWN:

```

```

if counter==0:
    point1=int(x//scale),int(y//scale);
    counter +=1
    myColor
    (random.randint(0,2)*200,random.randint(0,2)*200,random.randint(0,2)*200 )
elif counter ==1:
    point2=int(x//scale),int(y//scale)
    type = input('Enter Type')
    name = input ('Enter Name ')
    myPoints.append([point1,point2,type,name])
    counter=0
    circles.append([x,y,myColor])
    counter2 += 1

img = cv2.imread(path)
img = cv2.resize(img, (0, 0), None, scale, scale)

while True:
    # To Display points
    for x,y,color in circles:
        cv2.circle(img,(x,y),3,color,cv2.FILLED)
    cv2.imshow("Original Image ", img)
    cv2.setMouseCallback("Original Image ", mousePoints)
    if cv2.waitKey(1) & 0xFF == ord('s'):
        print(myPoints)
        break

from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.utils import np_utils
import pickle
import matplotlib.pyplot as plt
import numpy as np

(X_train, y_train), (X_test, y_test) = mnist.load_data()

print("Сумарна кількість зображень з набору даних MNIST: ",(X_train.shape[0]+X_test.shape[0]))
print("Кількість класів зображень з набору даних MNIST: ",10)
print("Структура даних для тренувального набору:")
print("Структура масиву з матрицями зображень: ",X_train.shape)
print("Структура масиву з ярликами(labels): ", y_train.shape)
print("Структура даних для тестувального набору:")
print("Структура масиву з матрицями зображень: ",X_test.shape)
print("Структура масиву з ярликами(labels): ",y_test.shape)

```

```

X_train = X_train.reshape((X_train.shape[0], 28, 28, 1)).astype('float32')
X_test = X_test.reshape((X_test.shape[0], 28, 28, 1)).astype('float32')
samples_num = []

X_train = X_train / 255
X_test = X_test / 255

print(X_train.shape)
for x in range(10):
    #print(len(np.where(y_train==x)[0]))
    samples_num.append(len(np.where(y_train==x)[0]))

y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
num_classes = y_test.shape[1]

plt.figure(figsize = (10,5))

plt.bar(range(num_classes),samples_num)
plt.title('Розподіл навчальної вибірки по класам')
plt.xlabel('Клас з зображеннями цифри')
plt.ylabel('Кількість зображень')
plt.show()

def myModel():
    model = Sequential()
    model.add(Conv2D(30, (5, 5), input_shape=(28, 28, 1), activation="relu"))
    model.add(MaxPooling2D())
    model.add(Conv2D(15, (3, 3), activation="relu"))
    model.add(MaxPooling2D())
    model.add(Dropout(0.2))
    model.add(Flatten())
    model.add(Dense(128, activation="relu"))
    model.add(Dense(50, activation="relu"))
    model.add(Dense(num_classes, activation="softmax"))

    #compilation
    model.compile(loss="categorical_crossentropy", optimizer="adam",
metrics=["accuracy"])
    return model

model = myModel()
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=10,
batch_size=200)
scores = model.evaluate(X_test, y_test, verbose=0)
print("Large CNN Error: %.2f%%" % (100-scores[1]*100))

```

```
# ##### PLOT THE RESULTS
# plt.figure(1)
# plt.plot(history.history['loss'])
# plt.plot(history.history['val_loss'])
# plt.legend(['Тренувальний набір', 'Тестовий набір'])
# plt.title('Втрати')
# plt.xlabel('Епохи (прогони тренування моделі)')
# plt.figure(2)
# plt.plot(history.history['accuracy'])
# plt.plot(history.history['val_accuracy'])
# plt.legend(['Тренувальний набір', 'Тестовий набір'])
# plt.title('Точність')
# plt.xlabel('Епохи (прогони тренування моделі)')
# plt.show()

pickle_out = open("trained_model.p", "wb")
pickle.dump(model, pickle_out)
pickle_out.close()
```

					ДП 7233.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		80

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

УЗГОДЖЕНО

Керівник проєкту

_____ Майя СПЕРКАЧ
(підпис) (вл. ім'я, прізвище)

“05” квітня 2021 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“06” квітня 2021 р.

Інформаційна система розпізнавання числової інформації.

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 7233.01.000 ТЗ*

на 11 сторінках

Київ – 2021 року

ЗМІСТ

1	ЗАГАЛЬНІ ПОЛОЖЕННЯ	4
1.1	Повне найменування системи та її умовне позначення	4
1.2	Найменування організації-замовника та організацій-учасників робіт ..	4
1.3	Перелік документів, на підставі яких створюється система	4
1.4	Планові терміни початку і закінчення роботи зі створення системи	4
2	ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	5
2.1	Призначення засобів.....	5
2.2	Цілі створення засобів.....	5
3	ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ	6
4	ВИМОГИ ДО СИСТЕМИ	7
4.1	Вимоги до системи в цілому.....	7
4.1.1	Вимоги до структури та функціонування системи	7
4.1.2	Вимоги до надійності	7
4.2	Вимоги до функціональних характеристик	7
4.3	Вимоги до видів забезпечення	8
4.3.1	Вимоги до математичного забезпечення.....	8
4.3.2	Вимоги до інформаційного забезпечення.....	8
4.3.3	Вимоги до прикладного програмного забезпечення.....	8
4.3.4	Вимоги до технічного забезпечення.....	8

					ДП 7233.01.000 ПЗ				
		Прізвище	Підпис	Дата					
Розроб.		Кравець О. О.			Інформаційна система розпізнавання числової інформації	Лім.	Лист	Листів	
Перевірив.		Сперкач М. О.					2	11	
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72			
Н. кон.		Новінський В.П.							
Затв.		Сперкач М. О.							

5	СТАДІЇ ТА ЕТАПИ РОЗРОБКИ	9
6	ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	11
6.1	Об'єм та види випробування	11

					ДП 7233.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування системи: «Інформаційна система розпізнавання числової інформації».

Коротке найменування системи: «Numbers recognition».

1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником системи є кафедра автоматизованих систем обробки інформації і управління факультету інформатики та обчислювальної техніки НТУУ «КПІ».

Розробник системи – студентка факультету інформатики та обчислювальної техніки НТУУ «КПІ імені Ігоря Сікорського, групи ІС-72 Кравець Олександра Олегівна.

1.3 Перелік документів, на підставі яких створюється система

Підставою для розробки «Numbers recognition» є завдання на дипломне проектування.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку робіт з «Numbers recognition»– 1 лютого 2021 року.

Плановий термін завершення робіт з «Numbers recognition»– 1 червня 2021 року.

					ДП 7233.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення засобів

Даний дипломний проект присвячений процесу розпізнаванню числової інформації написаної від руки (у вигляді відповідей студента або учня), яку буде отримано з відповідного бланку для відповідей, її подальша обробка і збереження.

2.2 Цілі створення засобів

Метою розробки є автоматизація процесу перевірки контрольних робіт студентів або учнів (де в якості відповідей маємо числові значення) шляхом розпізнавання написаних студентами (учнями) відповідей у відповідному бланку для відповідей за рахунок застосування моделей машинного навчання.

Завдяки створеній системі користувачі, у вигляді людей, які перевіряють роботи, можуть суттєво зменшити час для їх перевірки.

Для досягнення мети необхідно вирішити такі задачі:

- підготовка вибірок даних для тестування моделей машинного навчання;
- навчання нейронної мережі на підготовленій вибірці;
- класифікація зображень чисел, написаних від руки до певного числа;
- обробка зображення бланку з відповідями (вирівнювання, видалення зайвих частин фото);
- виділення окремих фрагментів бланку, звідки буде зчитуватись інформація;
- оцінка отриманих відповідей;
- оцінка достовірності результатів.

					ДП 7233.01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Говорячи, про процес розпізнавання рукописних чисел, можна стверджувати, що він є темою досліджень протягом багатьох років саме завдяки своїм широким сферам застосування, таким як перетворення книг і документів в електронний вигляд, розпізнавання поштових індексів, обробка форм для відповідей та іншим.

Якщо говорити про обробку форм для відповідей, то можна виділити процес перевірки викладачем великої кількості робіт студентів, який завжди вимагає великої кількості часу та високого рівня уваги.

Саме через це, у даній дипломній роботі пропонується автоматизоване рішення задачі перевірки контрольних робіт студентів та учнів, відповіді яких подаються у вигляді числової інформації, написаною від руки. У ході виконання дипломного проєкту розробляється інформаційна система розпізнавання рукописної числової інформації для автоматизації процесу перевірки викладачем відповідей студента, отриманих з заповненого бланку для відповідей. Система буде реалізована у вигляді веб-застосунку

У процесі роботи створюється продукт, яким зручно користуватися. Для використання даної системи необхідний лише комп'ютер з браузером та доступом до мережі Інтернет.

					ДП 7233.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

4 ВИМОГИ ДО СИСТЕМИ

4.1 Вимоги до системи в цілому

4.1.1 Вимоги до структури та функціонування системи

У системі розпізнавання числових символів функціонує лише один актор – користувач, який має мати можливість завантажити дані для проведення перевірки і отримати підраховану системою оцінку.

4.1.2 Вимоги до надійності

Для коректної роботи з системою технічні засоби користувачів мають мати стабільне підключення до мережі Інтернет та мати базові засоби введення та виведення інформації:

- монітор;
- клавіатура;
- комп'ютерна миша.

4.2 Вимоги до функціональних характеристик

Система «Numbers recognition» має реалізовувати наступний функціонал:

- Користувач має мати можливість завантажити до системи фотографію заповненого бланку: користувач робить фотографію бланку з відповідями студента та завантажує її до системи, а сама система у цей час спочатку розпізнає сам бланк з відповідями, потім розпізнає та зчитує відповідні числові символи(у вигляді відповідей студента);
- Користувач має мати можливість завантажити до системи файл з відповідями: для того щоб системи мала можливість виставити бали студенту, вона повинна мати дані, з якими треба порівняти числові відповіді, які були отримані з бланку;
- Користувач має мати можливість отримати оцінку одразу, без збереження її значення до файлу: користувач має можливість одразу після

					ДП 7233.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

завантаження зображення та його обробки, побачити оцінку студента, без подальшого її збереження;

- Користувач має мати можливість перевірити правильність сканування оцінок: користувач отримує список з відповідями з бланку на головному екрані.

4.3 Вимоги до видів забезпечення

4.3.1 Вимоги до математичного забезпечення

Система має містити натреновану модель нейронною мережі, за допомогою якої буде відбуватися процес класифікації чисел з зображення заповненого бланку.

4.3.2 Вимоги до інформаційного забезпечення

Для навчання нейронної мережі дана система має використовувати набір даних MNIST - це популярний набір даних для всіх інформаційних ресурсів по темі машинного навчання та класифікації зображень. Він містить 70 000 зображень рукописних, з яких потім можна створити набори даних з 60 000 навчальних зображень та 10 000 тестових зображень.

Кожне зображення має роздільну здатність 28x28, а числові значення пікселів вказані у градаціях сірого. Кожне зображення представлене матрицею 28x28, кожен елемент матриці має ціле число від 0 до 255 (пізніше буде приведено до варіанту від 0 до 1). Ярлик(label) кожного зображення - це передбачувана цифра рукописного зображення від 0 до 9

4.3.3 Вимоги до прикладного програмного забезпечення

Для коректної роботи з системою користувачі мають мати комп'ютер зі встановленим одним з браузерів.

4.3.4 Вимоги до технічного забезпечення

Даний програмний продукт представлений у вигляді веб-додатку, який

					ДП 7233.01.000 ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

запускається на локальному сервері.

Мінімальні вимоги до характеристик компонентів технічного забезпечення, при яких значення часових параметрів Системи повинні відповідати пред'явленим в ТЗ вимогам:

- процесор з тактовою частотою не нижче 2 ГГц;
- остатній об'єм оперативної пам'яті (не менше 2 ГБ);
- жорсткий диск (не менше 40 ГБ);
- Python 3.7;
- Tensorflow 2.0.0, keras 2.3.1, matplotlib 3.1, numpy 1.18, opencv 3.4,
- scikit-learn 0.22;

5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

					ДП 7233.01.000 ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні етапи виконання робіт з розробки наведені в таблиці 5.1.

Таблиця 5.1 – Основні етапи виконання робіт

№	Назва етапу	Термін виконання
1	Вивчення рекомендованої літератури	10.04.2021
2	Аналіз існуючих методів розв'язання задачі	14.04.2021
3	Постановка та формалізація задачі	16.04.2021
4	Розробка програмного забезпечення	20.04.2021
5	Налагодження програми	24.04.2021
6	Оформлення пояснювальної записки	28.04.2021
7	Подання ДП на попередній захист	14.05.2021
8	Подання ДП на основний захист	04.06.2021

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

Приймання програмного продукту має відбутися 14.05.2021.

Подання готового дипломного проєкту на основний захист має відбутися 04.06.2021.

6.1 Об'єм та види випробування

В рамках мануального тестування виділяється наступний перелік функцій:

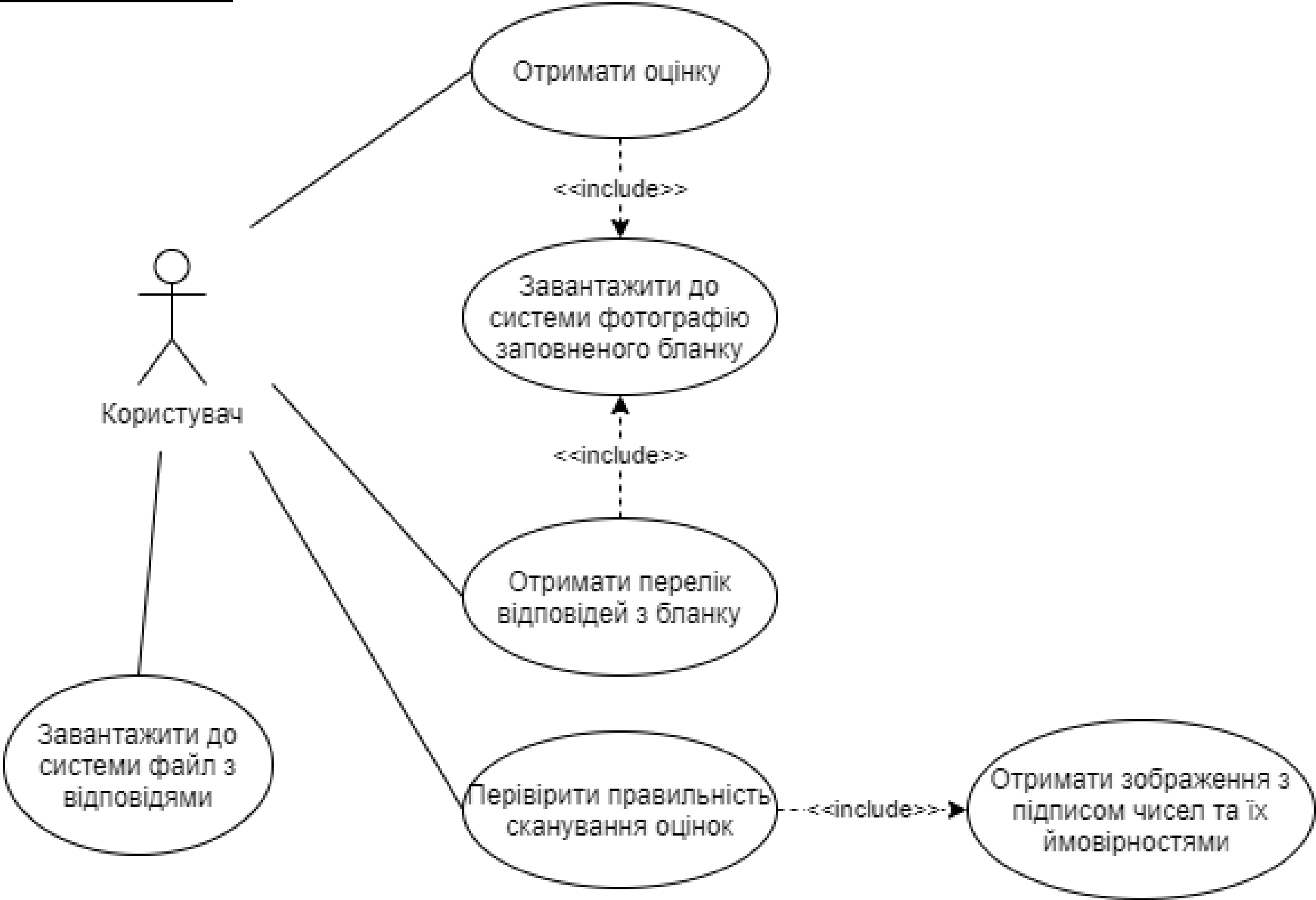
- тестування системи на виведення розпізнаних чисел з зображення заповненого бланку;
- тестування системи на виведення чисел, що являють собою правильні відповіді з відповідного файлу;
- тестування системи на виведення оцінки на основі завантажених до неї даних;
- тестування системи на можливість перевірки обробки і класифікацію чисел з бланку з різною кількістю відповідей;
- тестування системи на можливість виставлення оцінки, при різній кількості отриманих на вхід чисел з бланку.

					ДП 7233.01.000 ТЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

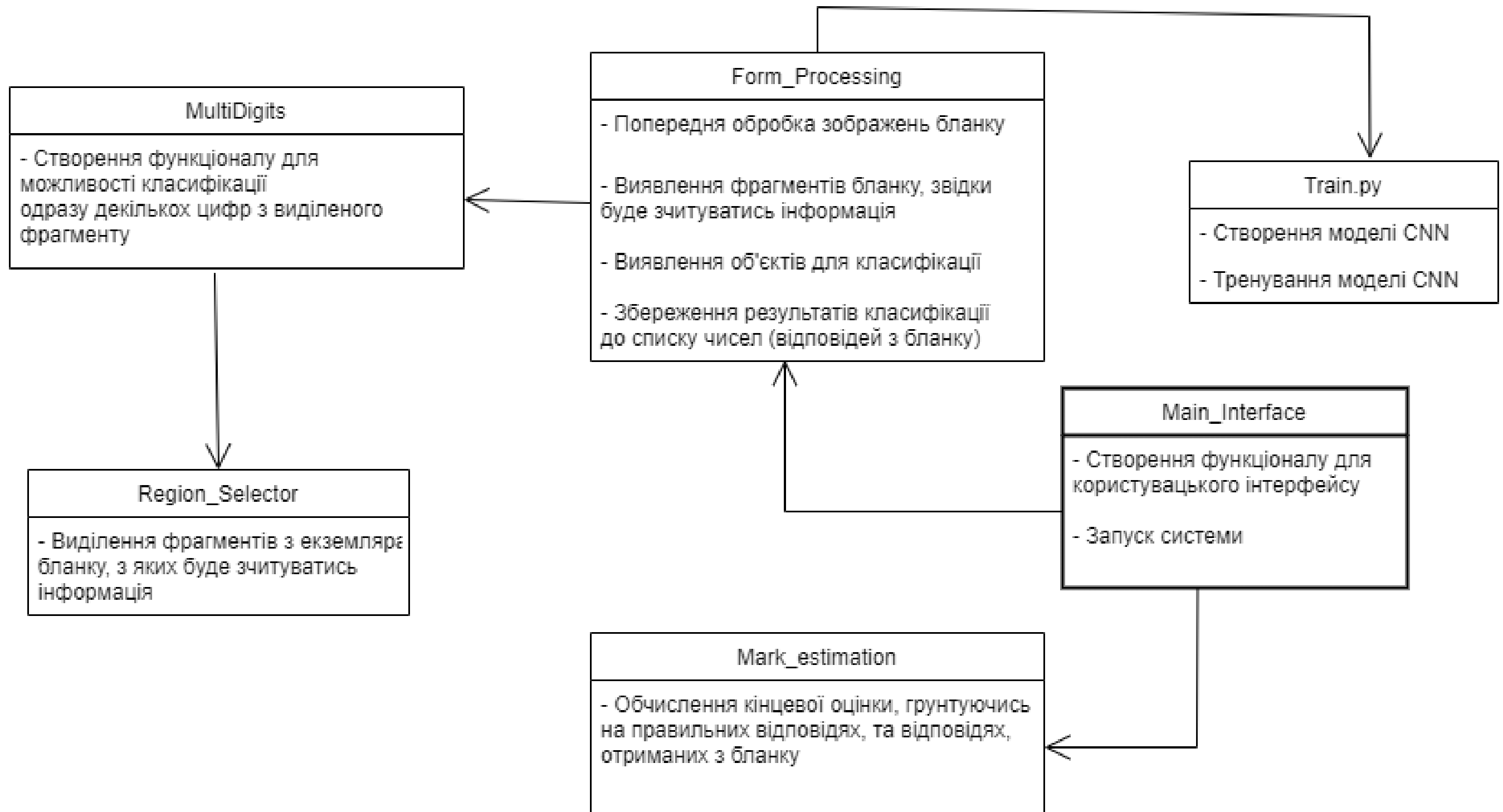
Графічний матеріал до дипломного проєкту

на тему: *«Інформаційна система розпізнавання числової
інформації»*

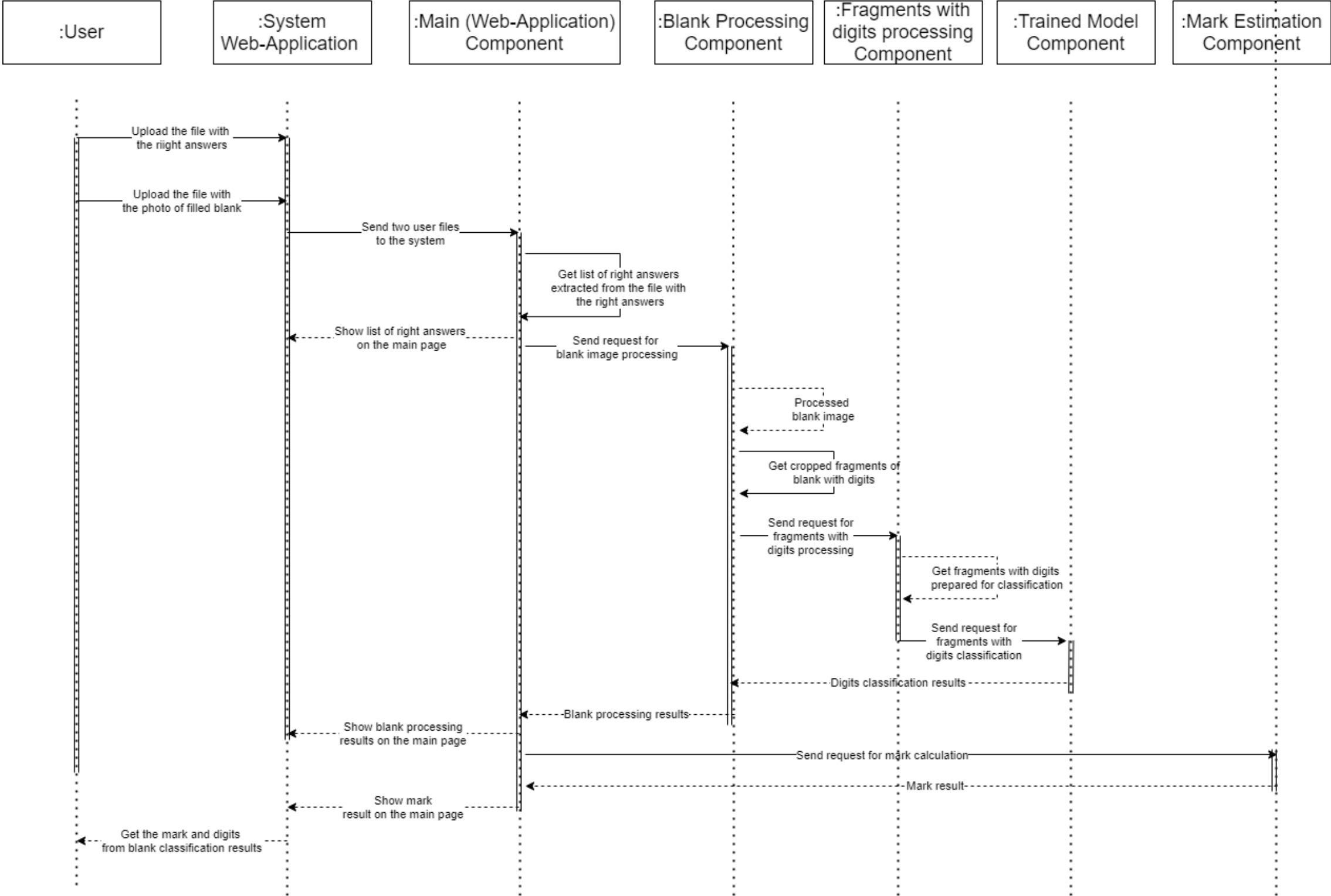
Київ – 2021 року



					ДП 7233.02.000 ССВ				
					Схема структурна варіантів використання				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Кравець О. О.							
Перевірів		Сперкач М. О.							
Консульт.									
Н. кон.		Новінський В.П.							
Затвердив		Сперкач М. О.							
					Інформаційна система розпізнавання числової інформації				
					Літера Маса Масштаб				
					Аркуш 1 Аркушів 1				
					КПІ ім. Ігоря Сікорського каф. АСОІУ гр. ІС-72				



					ДП 7233.03.000 ССВ						
					Схема структурна класів програмного забезпечення	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Кравець О. О.									
Перевірів		Сперкач М. О.				Аркуш 1		Аркушів 1			
Консульт.					Інформаційна система розпізнавання числової інформації	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72					
Н. кон.		Новінський В.П.									
Затвердив		Сперкач М. О.									



					ДП 7233.04.000 ССП			
Зм.	Арк.	№ документа	Підпис	Дата	Схема структурна послідовності програмного забезпечення	Літера	Маса	Масштаб
Розробив	Кравець О. О.							
Перевірив	Сперкач М. О.					Аркуш 1	Аркушів 1	
Консульт.					Інформаційна система розпізнавання числової інформації	КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-72		
Н. кон.	Новінський В.П.							
Затвердив	Сперкач М. О.							

Перевірка відповідей

Завантажте файл з правильними відповідями



Drag and drop file here

Limit 200MB per file • XLSX

Browse files



Книга1.xlsx 8.1KB



Завантажте фотографію бланка з відповідями



Drag and drop file here

Limit 200MB per file • JPG, PNG

Browse files



blank_test3.jpg 49.4KB



Правильні відповіді: [545.72, 34.48, 76.25, 34.11, 201.69]



Ідентифікатор		03
Відповідь 1	48	23
Відповідь 2	5473	91
Відповідь 3	463	80
Відповідь 4	110	15
Відповідь 5		
Відповідь 6		
Відповідь 7		
Відповідь 8		
Відповідь 9		
Оцінка		

Відповіді з бланку: ['603', 48.23, 5473.95, 463.8, 110.15, '', '', '', '', '', '', '', '']

Відсоток правильних відповідей студента 603 складає 0.0

					ДП 7233.06.000 КЕ						
					Креслення вигляду екранних форм	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Кравець О. О.									
Перевірив		Сперкач М. О.				Аркуш 2			Аркушів 2		
Консульт.					Інформаційна система розпізнавання числової інформації	КПІ ім. Ігоря Сікорського каф. АСОІУ гр. ІС-72					
Н. кон.		Новінський В.П.									
Затвердив		Сперкач М. О.									