Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Комп'ютерна система класифікації об'єктів на зображеннях з використанням штучних нейронних мереж
(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-1
Олександр СТУКОТА
(власне ім’я, прізвище)

Спеціальність
123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп’ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Ірина КЛИМОВА
(посада, власне ім’я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім’я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Стукоті Олександру Володимировичу
(прізвище, ім’я, по батькові)

1. Тема роботи Комп'ютерна система класифікації об'єктів на зображеннях з використанням штучних нейронних мереж

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи
кластеризація
реалізація алгоритмів класифікації
програмні засоби
Python
вибір датасету
розпізнавання тварин

4. Перелік питань, що потрібно опрацювати у роботі
Аналіз предметної області
Вибір програмних засобів для розробки
Реалізація програмних засобів класифікації

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 10 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих систем	31.05.2025–03.06.2025	
3	Вибір засобів для реалізації	04.06.2025–06.06.2025	
4	Програмна реалізація	07.06.2025–08.06.2025	
5	Реалізація штучної нейронної мережі	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Ірина КЛИМОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 58 с., 16 рис., 2 дод., 6 джерел.

РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, ГЛИБОКЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, КОМП'ЮТЕРНИЙ ЗІР, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, ІДЕНТИФІКАЦІЯ ВИДІВ, ТАКСОНОМІЯ, БІОРІЗНОМАНІТТЯ, ВЕБ-ЗАСТОСУНОК, ШТУЧНИЙ ІНТЕЛЕКТ, ОБРОБКА ЗОБРАЖЕНЬ.

Метою кваліфікаційної роботи є розробка комп’ютерної системи класифікації об’єктів на зображеннях з використанням Python.

У ході виконання кваліфікаційної роботи розроблено застосунок, який характеризується модульною архітектурою, що складається з кількох взаємопов'язаних компонентів. Центральним елементом системи є клас AnimalRecognizer, який інкапсулює логіку обробки зображень та взаємодії з нейронною мережею. Цей компонент забезпечує попередню обробку завантажених зображень відповідно до вимог моделі MobileNetV2, включаючи масштабування до розміру 224×224 пікселі та нормалізацію значень пікселів. Важливою особливістю реалізації є система перекладу результатів розпізнавання на українську мову. Створено спеціалізований словник відповідностей між англійськими назвами з набору ImageNet та їх українськими еквівалентами, що забезпечує природність сприйняття результатів користувачами. Додатково імплементовано систему таксономічної класифікації, яка надає інформацію про приналежність розпізнаних організмів до відповідних біологічних родин.

ABSTRACT

Bachelor’s thesis: 58 pages, 16 figures, 2 appendices, 6 sources.

IMAGE RECOGNITION, MACHINE LEARNING, DEEP LEARNING, NEURAL NETWORKS, COMPUTER VISION, IMAGE CLASSIFICATION, SPECIES IDENTIFICATION, TAXONOMY, BIODIVERSITY, WEB APPLICATION, ARTIFICIAL INTELLIGENCE, IMAGE PROCESSING.

The major goal of this thesis is the development of a computer system for image-based object classification using Python.

In order to an application was developed featuring a modular architecture composed of several interconnected components. The core element of the system is the AnimalRecognizer class, which encapsulates the logic for image processing and interaction with the neural network. This component performs preprocessing of the uploaded images in accordance with the requirements of the MobileNetV2 model, including resizing 224 x 224 pixels and pixel value normalization. A notable feature of the implementation is the translation system that converts recognition results into Ukrainian. A specialized dictionary was created to map English labels from the ImageNet dataset to their Ukrainian equivalents, enhancing the natural perception of the results by users. Additionally, a taxonomic classification system was implemented, providing information about the recognized organisms' affiliation with their respective biological families.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – штучний інтелект

API – інтерфейс програмування застосунків

CNN – згорткова нейронна мережа

CPU – центральний процесор

CRUD – створити, читати, оновити, видалити

CSV – значення, розділені комами

DB – база даних

DL – глибоке навчання

Flask – мікрофреймворк для веб-розробки

GIF – формат обміну графікою

GPU – графічний процесор

HTML – мова розмітки гіпертексту

HTTP – протокол передачі гіпертексту

JPEG – стандарт стиснення зображень

JSON – нотація об'єктів JavaScript

ML – машинне навчання

MVC – модель-вид-контролер

OCR – оптичне розпізнавання символів

ORM – об'єктно-реляційне відображення

PIL – бібліотека обробки зображень Python

PNG – переносна мережева графіка

RAM – оперативна пам'ять

REST – передача репрезентативного стану

RGB – червоний, зелений, синій

SDK – комплект розробки програмного забезпечення

SQL – мова структурованих запитів

SQLite – легка система управління базами даних

TensorFlow – бібліотека машинного навчання

UI – користувацький інтерфейс

URL – уніфікований локатор ресурсів

UTF – формат трансформації Unicode

UUID – універсальний унікальний ідентифікатор

UX – користувацький досвід

WSGI – інтерфейс веб-сервера-шлюзу

ВСТУП

Серед найбільш ресурсоємних і часовитратних завдань у сфері взаємодії з людьми виокремлюється процес ідентифікації та перевірки документів. Це зумовлює нагальну потребу в розробці високоефективних автоматизованих систем розпізнавання символів, здатних оптимізувати та прискорити зазначені процедури. З огляду на зростання обсягів інформації та зростаючі вимоги до оперативності, системи автоматизованої класифікації набувають ключового значення в забезпеченні якості обслуговування в широкому спектрі економічних та адміністративних сфер.

Інструменти, засновані на технологіях комп’ютерного зору для автоматизованого розпізнавання символів, особливо актуальні в контексті застосування на контрольних пунктах доступу, у закладах різного призначення та в організаційних структурах. У таких умовах вони здатні забезпечити як підвищення продуктивності, так і зменшення навантаження на персоМетою кваліфікаційної роботи є розробка комп’ютерної системи класифікації об’єктів на зображеннях з використанням машинного навчання.

Завдання:

- аналіз існуючих програмних засобів;
- аналіз алгоритмів розпізнавання текстів на зображеннях;
- розробка програмних засобів розпізнавання тексту;;
- тестування розроблених засобів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні відомості

Системи комп’ютерного зору становлять собою поєднання теоретичних засад і прикладних технологічних рішень, розроблених для вирішення задач виявлення та класифікації об’єктів у цифровому середовищі. Попри динамічний розвиток засобів автоматизованого розпізнавання текстової інформації, ця галузь поки перебуває на ранньому етапі становлення, що, однак, не применшує її високого прикладного потенціалу. На сьогодні технології комп’ютерного бачення вже успішно впроваджуються в різноманітні бізнес-сфери, сприяючи цифровізації процесів, хоча все ще поступаються людському сприйняттю за точністю класифікації. Водночас, прогрес у галузі дозволяє припустити, що найближчим часом буде досягнуто якісно нового рівня ефективності.

Державні документи відіграють ключову роль у процесах легітимного пересування осіб як всередині країни, так і за її межами. Їх оформлення та видача здійснюється через систему надання адміністративних послуг, що діє відповідно до національного законодавства. Дані документи мають затверджену форму і супроводжують громадянина протягом усього життя, починаючи з народження.

У світлі цифрової трансформації державного управління зростає необхідність у впровадженні інноваційних рішень, які забезпечують не лише вищу якість обслуговування, а й підвищення безпеки даних. Сучасні державні документи дедалі частіше містять елементи біометричної ідентифікації та компоненти, призначені для машинного зчитування. Одним із таких інновацій є спеціальна зона в паспортних документах, оптимізована для автоматичного розпізнавання, що істотно прискорює процес перевірки особи та забезпечує відповідність міжнародним стандартам.

Розробка спеціалізованого програмного забезпечення для обробки таких документів підвищує продуктивність бізнес-процесів, пов’язаних із верифікацією, доступом до захищених об’єктів і послуг, а також дозволяє досягти значної економії ресурсів. Автоматизація розпізнавання передбачає й обробку текстових символів відповідно до стандартів машинного зору – наприклад, заміна пробілів на спеціальні символи або застосування алгоритмів скорочення надлишкових рядків, що забезпечує коректність зчитування.

1.2 Аналіз існуючих програмних засобів розпізнавання

Технології комп’ютерного зору репрезентують собою програмні засоби, призначені для автоматичного розпізнавання та цифрової обробки текстової інформації, що є доступною для сприйняття людиною. Основна функція таких рішень полягає у відтворенні візуального вмісту у вигляді структурованих даних, які зберігаються в цифровому форматі у комп’ютерній пам’яті. Таким чином, комп’ютерне бачення забезпечує ефективне перетворення зорової інформації у форму, придатну для подальшого аналізу та інтеграції в автоматизовані інформаційні системи.

Автоматизовані методи класифікації символів активно застосовуються для обробки друкованих матеріалів, таких як журнали, офіційні документи, посвідчення особи, проїзні квитки та інші носії текстової інформації. Використання технологій автоматичної інтерпретації символьних даних сприяє підвищенню ефективності та якості роботи в багатьох галузях економіки, дозволяючи переводити текстову інформацію з аналогових носіїв у цифровий формат.

Застосування таких технологій є особливо актуальним в умовах об’єктів з контрольованим доступом, де обсяг символьних або графічних даних потребує оцифрування для забезпечення швидкого доступу та

подальшої обробки. На практиці цей процес, як правило, здійснюється за допомогою пристроїв для захоплення зображень, що є першим етапом у ланцюгу автоматизованої класифікації даних.

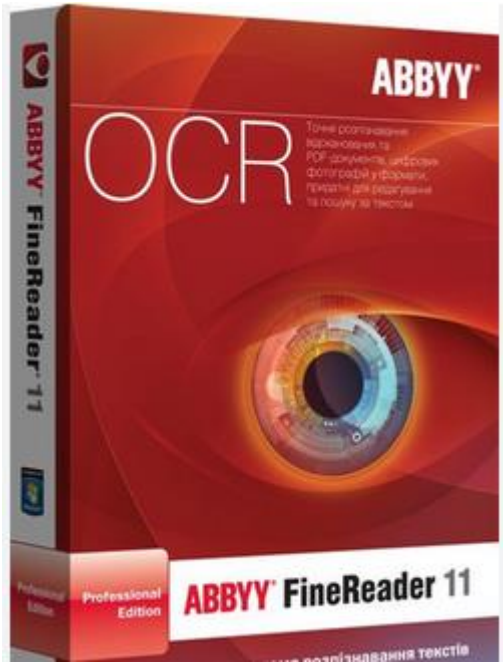

Рисунок 1.1 – Система розпізнавання ABBYY FineReader

У цій сфері особливу роль відіграють спеціалізовані програмні комплекси. Одним із прикладів є ABBYY FineReader – система автоматичного розпізнавання символів, яка забезпечує високу точність класифікації символьної інформації з візуальних джерел, таких як фотографії та відскановані документи. Точність розпізнавання в межах від 83% до 99% свідчить про її високу надійність. Додатковою перевагою є підтримка понад 200 мов та можливість збереження розпізнаних даних у різних форматах, що дозволяє легко інтегрувати отриману інформацію в існуючі бізнес-процеси та цифрові середовища.

Серед розповсюджених рішень у галузі оптичного розпізнавання символів (OCR) вирізняється Simple OCR – програмний комплекс, орієнтований на класифікацію символів переважно європейських мов. Базовий функціонал цього застосунку доступний безкоштовно, що робить

його привабливим для широкого кола користувачів. Водночас його ефективність значно знижується при роботі з рідковживаними символами, нестандартними форматами або зображеннями низької якості. Для вирішення таких завдань необхідне придбання платних розширень, що забезпечують розширений набір функцій.

Рисунок 1.2 – Simple OCR

Simple OCR характеризується низькими вимогами до обчислювальних ресурсів, що забезпечує його стабільну роботу навіть на малопотужних пристроях. Однак одним з основних обмежень є залежність точності результатів від якості початкових зображень. У разі недостатньої деталізації значне підвищення витрат часу та ресурсів не гарантує суттєвого покращення результатів класифікації.

Ще одним безкоштовним інструментом є Free OCR – програмне забезпечення, яке також забезпечує розпізнавання друкованих символів. Воно підтримує обробку файлів у різних форматах, зокрема зображень, і є зручним варіантом для задач базового рівня без необхідності фінансових витрат.

Рисунок 1.3 – Free OCR

Серед переваг аналізованого застосунку варто відзначити його безкоштовну доступність та інтуїтивно зрозумілий інтерфейс користувача, що значно спрощує процес ознайомлення та взаємодії навіть для користувачів без технічного досвіду. Програмне забезпечення підтримує обробку більшості форматів файлів та документів, за винятком PDF. У випадку з PDF-файлами функціональність суттєво обмежена: без наявності відповідного обґрунтування система дозволяє обробляти лише одне текстове поле за одну сесію.

Крім того, серед технічних обмежень зазначається неможливість обробки більш ніж п’яти фотодокументів одночасно, що знижує ефективність використання застосунку в умовах високого навантаження. Водночас згідно з офіційною документацією, інструмент підтримує мовну класифікацію для всіх європейських мов, включно з українською, що розширює його застосування для локалізованих задач.

Fider OCR – це багатофункціональний алгоритмічний інструмент, який вирізняється широкою сумісністю з іншими системами оптичного розпізнавання символів. Завдяки добре структурованій документації, його інтеграція та налаштування не викликають труднощів навіть у середовищах з багатокомпонентною архітектурою. Застосунок є мультиплатформенним, що

забезпечує гнучкість його використання на різних операційних системах.

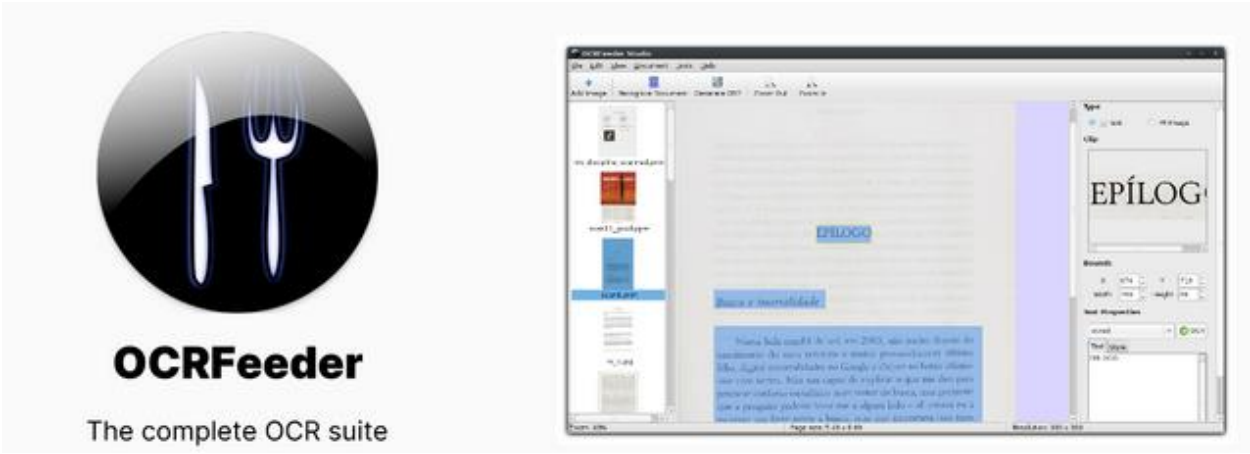

Рисунок 1.4 – Fider ORC

Основу Fider OCR становлять кілька ключових складових. Серед них PI1 - фреймворк для роботи з графічними зображеннями, написаний на Python; One-P – модуль для алгоритмічного розпізнавання; бібліотеки для взаємодії з графічним інтерфейсом користувача, зокрема GTK. Додаткову функціональність забезпечує модуль перевірки правильності орфографії, що підвищує точність кінцевого результату.

Функціонально система здатна приймати зображення у поширених графічних форматах на вхід та повертати текстові дані на виході. Для підвищення якості розпізнавання застосовуються алгоритми передобробки, які зменшують ймовірність помилок при класифікації. Присутній також модуль контролю якості розпізнаного тексту. Fider ORC підтримує інтеграцію з іншими програмами, що здійснюють захоплення зображень, а також дозволяє обробляти декілька зображень одночасно.

Tesseract – це спеціалізована бібліотека, створена для промислового розпізнавання текстової інформації на основі графічних зображень. Вона підтримує обробку текстів понад ста мовних груп, включаючи можливість адаптації до нових, раніше не відомих алфавітів. Платформа розвивається як відкритий проєкт, який підтримується глобальною спільнотою розробників, що забезпечує постійне оновлення ядра бібліотеки, а також її довготривалу

підтримку.

Рисунок 1.5 – Tessaract

Одним із найважливіших етапів її еволюції стало впровадження рекурентних нейронних мереж із використанням механізму уваги, що значно підвищує ефективність класифікації тексту навіть у складних умовах. Відкрита архітектура Tesseract дозволяє інтегрувати її з широким спектром сучасних мов програмування, що відкриває можливість розробки гнучких прикладних рішень. Протягом років численні учасники проєкту створили велику кількість додаткових інтерфейсів і оболонок, що розширюють функціональність базової бібліотеки.

Окремо варто згадати внесок компанії Google, яка підтримує екосистему інструментів і надає стабільний API-доступ до своїх технологій. Незважаючи на розширення функціоналу, програмний інтерфейс залишається стабільним, що забезпечує довготривалу підтримку вже розроблених рішень і полегшує розширення їх можливостей. Водночас важливо враховувати, що повноцінне використання API передбачає регулярні фінансові витрати, що може обмежувати доступність цієї технології на стадії створення прототипів або у наукових дослідженнях.

На сучасному етапі розвитку цифрових технологій автоматизоване введення великих обсягів текстової інформації та її трансформація у придатну до зберігання форму здійснюється за допомогою спеціалізованого

програмного забезпечення, здатного ефективно захоплювати інформацію з реального середовища. За результатами аналізу й тестування провідних інструментів для автоматичного розпізнавання символів, було виокремлено найбільш результативні платформи – Tesseract OCR та Google Vision API.

Назва	Витрати	Платформа	Точність	Документація
ABBYY Finereader	Відсутні	Довільна	98%	Задовільно
Simple OCR	Відсутні	Довільна	50-90%	Погано
Free OCR	Відсутні	Довільна	60-90%	Добре
OCR Finder	Відсутні	Linux	60-90%	Задовільно
Tesseradct OCR	Відсутні	Хмара	60-99%	Задовільно
Google Vision	$1,5/1000	Хмара	>90%	Добре

Рисунок 1.6 – Порівняння проаналізованих інструментів

Кожен із розглянутих інструментів має свої унікальні переваги та обмеження. Інструмент від Google вирізняється високою точністю класифікації символів, адаптованістю до широкого кола користувачів і зручністю у використанні. Проте його функціонування передбачає стабільне інтернет-з'єднання та регулярні фінансові витрати, що може обмежити його застосування в умовах обмежених ресурсів.

Натомість Tesseract OCR є відкритим мультиплатформним рішенням, що забезпечує автономну роботу без необхідності підключення до мережі. Активна підтримка міжнародної спільноти сприяє його постійному вдосконаленню, включаючи адаптацію під різні алфавіти. Проте в порівнянні з рішенням Google, Tesseract вимагає більше технічних знань для налаштування і не завжди забезпечує однакову точність розпізнавання.

У цьому контексті зростає актуальність створення програмної системи, що об'єднувала б переваги доступності, простоти використання та автономності, забезпечуючи при цьому високу якість розпізнавання текстової інформації без додаткових фінансових витрат.

2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ

2.1 Робота OCR

У типових програмних системах функціональна структура поділяється на дві основні компоненти. Серверна частина, що реалізує логіку взаємодії та обробки запитів, зазвичай функціонує на віддалених обчислювальних ресурсах. Клієнтська ж частина, орієнтована на взаємодію з користувачем, відповідає за графічне представлення інтерфейсу та контроль елементів введення-виведення. З метою реалізації запропонованої системи для автоматизованого розпізнавання текстової інформації було обрано платформу Telegram як інтерфейс користувача, що забезпечує широке охоплення та простоту вбудування у щоденні цифрові практики.

В основу розробки лягла бібліотека Tesseract OCR, яка вирізняється відкритістю програмного коду. Її публічна доступність сприяє активному розвитку функціоналу силами міжнародної спільноти програмістів, що, у свою чергу, забезпечує високий рівень підтримки, постійне оновлення документації та оперативне вирішення можливих технічних труднощів. Така гнучкість і масштабованість робить Tesseract найбільш доцільним вибором на етапі прототипування та подальшого розгортання програмного продукту, орієнтованого на ефективну автоматичну обробку символьної інформації.

v Розглянута система побудована на відкритому програмному стеку, що не містить обмежень щодо комерційного використання і не вимагає ліцензійних дозволів від розробника. Завдяки модульній структурі та наявності численних API-декораторів для різних мов програмування, Tesseract OCR легко інтегрується з іншими компонентами інформаційної екосистеми. Це робить її придатною для формування каскадного конвеєра автоматичних операцій з розпізнавання символів у великих обчислювальних системах.

У своїй основі Tesseract OCR реалізує комбінацію алгоритмів, серед яких центральне місце належить згортковим нейронним мережам для обробки графічних компонентів та розпізнавання окремих символів, а також рекурентним мережам для аналізу їх послідовності у словах. Використовуючи ці технології, можна досягнути високої точності розпізнавання тексту, проте їх ефективність значною мірою залежить від якості підготовки даних та доступу до великого обсягу навчальних прикладів, зокрема для англомовних матеріалів.

Рисунок 2.1 – Результати в Тесаракт OCR

Проте Tesseract перебуває в постійному розвитку. На сьогодні активно готуються оновлення другого покоління – Tesseract-5, що передбачає впровадження архітектури на увазі (transformer), що дозволить підвищити адаптивність і точність системи. Попри це, для успішного використання сучасних нейронних технологій, що здатні навчатися самостійно, необхідно формувати значні мовні корпуси, що відповідають різноманіттю шрифтів, розмірів і сценаріїв застосування.

У цьому контексті актуальним є питання підготовки вхідних зображень. Необхідно уніфікувати їх формат до значень, придатних для розпізнавачів символів, наприклад у градаціях сірого або чорно-білому режимі, кадрово вирівняти, згладити шум, масштабувати до необхідної

роздільності і орієнтувати у відповідності з алгоритмічними припущеннями, що значно знижує навантаження при навчанні мережі. Афінні трансформації, зокрема обертання та масштабування, забезпечують коректне вирівнювання зображень з низькою витратою ресурсів.

Коли якість сторінок недостатня для прямого розпізнавання, застосовуються методи сегментації сторінок, що дозволяють виділити структуровані текстові блоки. У Tesseract передбачена низка підходів до інтерпретації структури документа – від автоматичного визначення зон до виділення окремих символів або рядків. Це відкриває значні можливості для адаптації під різні формати вхідних даних та рівні їхнього уніфікації, що є ключовим елементом у проєктуванні гнучких комплексних рішень для розпізнавання символів.

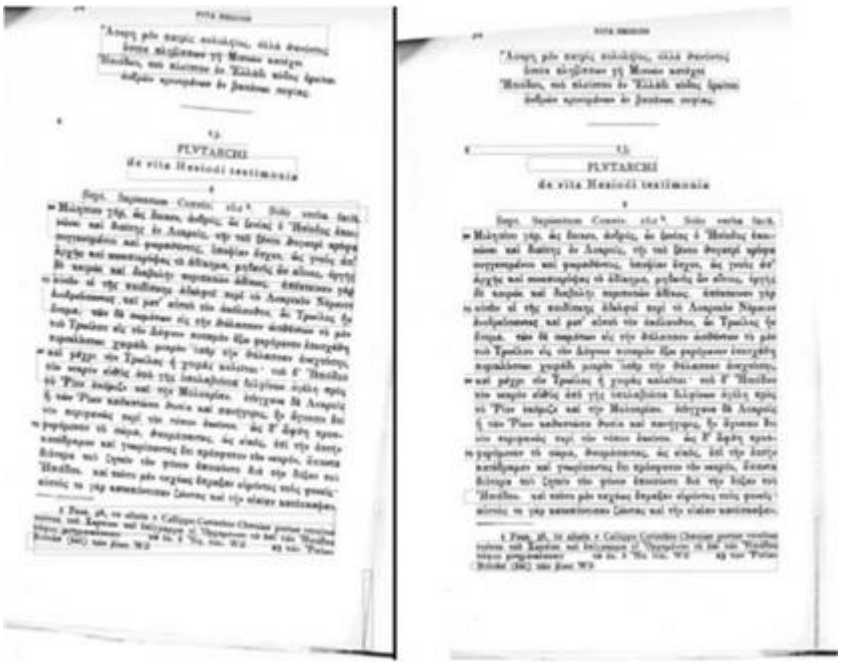

Рисунок 2.2 – Некоректна орієнтація

Функціональні можливості сучасних систем комп’ютерного зору забезпечують ефективну класифікацію та збирання символьної інформації безпосередньо з графічних зображень, зокрема з фотографій. У цьому контексті використовуються два основних підходи до обробки даних. Один з

них зосереджений на точному виявленні координат текстових елементів на зображенні з подальшим перетворенням вмісту в текстовий формат, що забезпечує глибоку обробку та стандартизоване представлення результату. Інформація, сформована в цьому процесі, містить як вхідні параметри графічного файлу, так і структуровану текстову складову, що може бути застосована у подальших етапах цифрової обробки.

Альтернативний підхід передбачає виявлення символьного вмісту з подальшим збереженням результату у форматі, придатному для інтеграції з іншими програмними модулями або сервісами. Такий формат більш орієнтований на оперативну передачу даних в інформаційні системи та програмні інтерфейси.

При розробці алгоритмів для класифікації текстової інформації доцільно застосовувати ті функції, що спроектовані спеціально для обробки відповідних типів вхідних даних. Це дозволяє автоматизувати процес та значно підвищити його ефективність за рахунок адаптованості алгоритмів до конкретних завдань.

2.2 Нормалізація вхідних даних

Завдяки реалізації функціоналу для детального аналізу текстової інформації, вхідні дані проходять обробку з подальшим формуванням вихідного об’єкта – ієрархічної структури, поданої в стандартизованому форматі. Ця структура включає такі рівні: сторінка, блок, параграф, слово та символ. Кожний рівень відображає певний ступінь структурування тексту: сторінка, що містить базові елементи та метадані (як-от розширення та рівень довіри до розпізнавання); блок тексту, що об’єднує фрагменти із спільним змістом; параграф, що формує смисловий ряд; слово, як основна одиниця тексту; символ – атомарна одиниця. Окрім цього, система здатна повертати посилання на зображення, які повністю або частково відповідають аналізованому тексту, що значно спрощує подальшу обробку.

З точки зору конфіденційності, реалізована система дотримується вимог законодавства щодо захисту персональних даних, а провайдер, який надає API, не зберігає графічну інформацію – обробка відбувається «в польоті». Відповідна політика конфіденційності забезпечує юридичну відповідальність за недопущення неправомірного використання, що гарантує користувачу безпеку і спокій.

За архітектурним рішенням, клієнтська взаємодія здійснюється через Telegram-інтерфейс, який використовує протокол захищеної передачі гіпертексту HTTPS. Комунікація з сервером передбачає надсилання команд або медіа-контенту, ідентифікація кожного з запитів виконується за допомогою ID взаємодії. Сервер обробляє вхідні повідомлення у вигляді оновлень, структурованих за часовим штампом та типом даних, що забезпечує ефективну обробку навіть за умов нестабільного з’єднання. У разі високого мережевого навантаження використовується логіка масштабування сервера та застосування спеціалізованих методів обробки потоків, що дозволяє обробляти тисячі запитів на секунду.

Реалізація доступу до Telegram API та обробки вхідних/вихідних повідомлень включає формалізовану структуру запитів і відповідей із перевірками статус-кодів. У разі помилок – виявлення їх типу, запис у лог-файли та сповіщення користувача про зупинку роботи. При потребі передати файл, не знайдений раніше, система використовує інструкції API: через нове навантаження файлу або за його унікальним ідентифікатором. Дозволений обсяг переданих даних регулюється провайдером Telegram і може змінюватися, що накладає певні обмеження на проектування архітектури системи.

2.3 Обґрунтування вибору мови програмування

У процесі реалізації програмної системи було обрано мову програмування Python, що є однією з найбільш динамічно розвиваних та

широко застосовуваних у сучасному програмному середовищі. Високий рівень популярності цієї мови зумовлений її лаконічним синтаксисом, великою кількістю бібліотек для реалізації широкого спектра функціональностей, а також активною спільнотою розробників, що сприяє швидкому пошуку та розв’язанню технічних проблем у процесі розробки.

Використання Python дозволяє дотримуватись сучасних стандартів розробки програмного забезпечення, забезпечуючи водночас ефективне масштабування та зручність у подальшій підтримці програмних продуктів. Для інтеграції з зовнішніми сервісами, зокрема з мережею Telegram, застосовуються спеціалізовані бібліотеки-обгортки, які, базуючись на публічному API, поєднують функціональність сервісу із можливостями мови. Такий підхід дозволяє реалізовувати як базові, так і високорівневі функції у межах складних архітектурних рішень.

Рисунок 2.3 – Python

Крім забезпечення доступу до API, бібліотеки Python надають низку абстрактних механізмів, що сприяють гнучкому проектуванню систем. Документація до таких бібліотек зазвичай містить детальні приклади використання та рекомендації щодо інтеграції, що полегшує їх вп

3 РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ КЛАСИФІКАЦІЇ

З урахуванням вимог щодо конфіденційності та ефективності обробки символьних даних було розроблено систему автоматичної класифікації символів на основі технологій комп'ютерного зору з використанням хмарних обчислювальних ресурсів. Центральною складовою архітектури системи є серверна частина, реалізована з використанням мережі Telegram, що забезпечує оперативність впровадження, широкі можливості взаємодії, наявність масштабної документаційної бази та мінімалістичний інтерфейс. Важливою перевагою обраної технології є її безкоштовне використання, що робить її привабливою для створення прототипів і повноцінних систем.

Розробка програмного коду здійснювалась на мові програмування Python, що надала змогу використовувати різноманітні бібліотеки з відкритим кодом, оптимізуючи процес розробки за рахунок повторного використання наявного функціоналу. Такий підхід дозволив зосередитись на реалізації основних алгоритмічних компонентів системи.

У процесі експлуатації система приймає на вхід зображення державного документа. Після завантаження зображення на сервер воно проходить попередню обробку, яка включає поворот, бінаризацію та фільтрацію шумів. Далі за допомогою методів комп'ютерного зору виконується розпізнавання текстової інформації на зображенні

3.1 Робота з хмарами

Конфігурація хмарної системи розпізнавання образів здійснюється через мережевий інтерфейс, що забезпечує повну взаємодію з технологічною інфраструктурою компанії Google. Основним інструментом взаємодії з підсистемами розпізнавання виступає веб-термінал, який відкриває доступ до функціоналу обраної хмарної платформи.

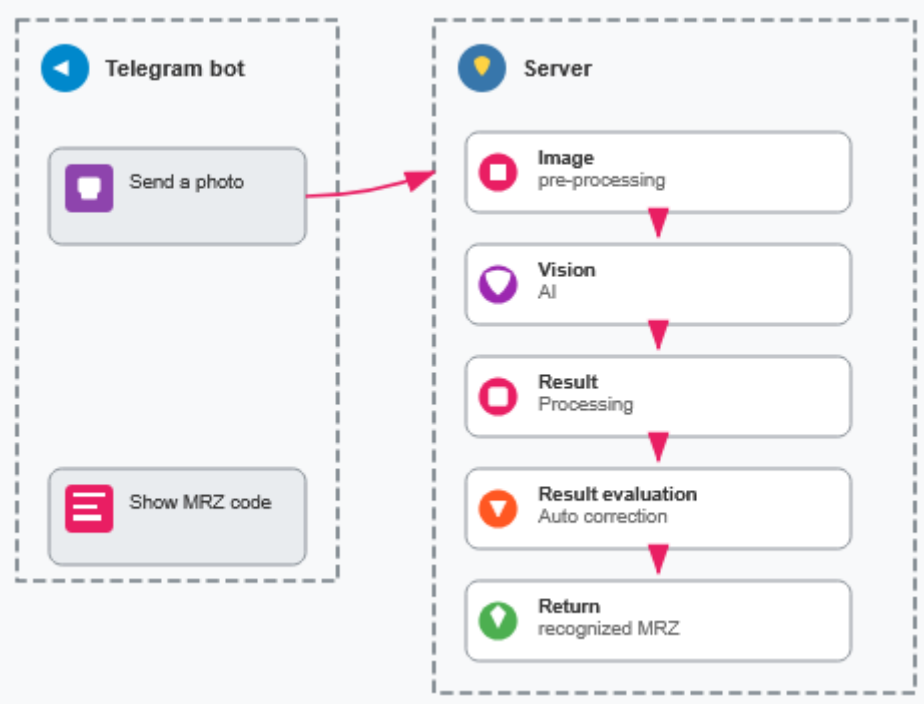

Рисунок 3.1 – Схема етапів виконання ПЗ

Першим кроком для отримання доступу до хмарного середовища є наявність облікового запису Google. Після реєстрації користувач має створити окремий проєкт (solution), який відіграє роль логічної одиниці організації робочого процесу в хмарному середовищі.

Проєкт включає такі компоненти:

- список користувачів із визначеними ролями та правами доступу;
- підключені сервіси та підсистеми, задіяні у реалізації задач;
- моніторингові інструменти, що забезпечують відстеження роботи системи;
- зовнішні контрагенти або сервіси-партнери;
- механізми автентифікації користувачів, що регламентують доступ до окремих сервісів.

Варто підкреслити, що архітектура облікового запису в Google Cloud Platform передбачає можливість створення кількох незалежних рішень (проектів). Кожен проєкт є автономною одиницею, яка містить власні налаштування, політики безпеки, користувачів, ресурси та сервіси. Така

структура дозволяє гнучко адаптувати конфігурацію під конкретні задачі, обмеження чи цілі організації, забезпечуючи високу масштабованість, безпечність та оптимальне використання хмарних обчислювальних ресурсів..

Рисунок 3.2 – Платформа Google

При можливості створення хмарного рішення потрібно пройти певну процедуру тобто виконати інструкції що приводять до фінального вирішення проблеми поставленої перед користувачем. Після завершення виконання наданих напередодні інструкцій користувач отримує зображення рішення. Також необхідно провести настройку системи що зможе приймати оплату до кожного рішення так як платформа є платною.

Використання публічних програмних інтерфейсів

Як тільки було проведено налаштування системи оплати ми маємо змогу додати систему розпізнавання до свого рішення. Щоб це зробити потрібно зайти в налаштування натиснути кнопку огляд та вибрати компоненту розпізнавання, потім це компоненти з'явиться у спеціальному розділі що буде сигналізувати про те що компонент було успішно додано до рішення.

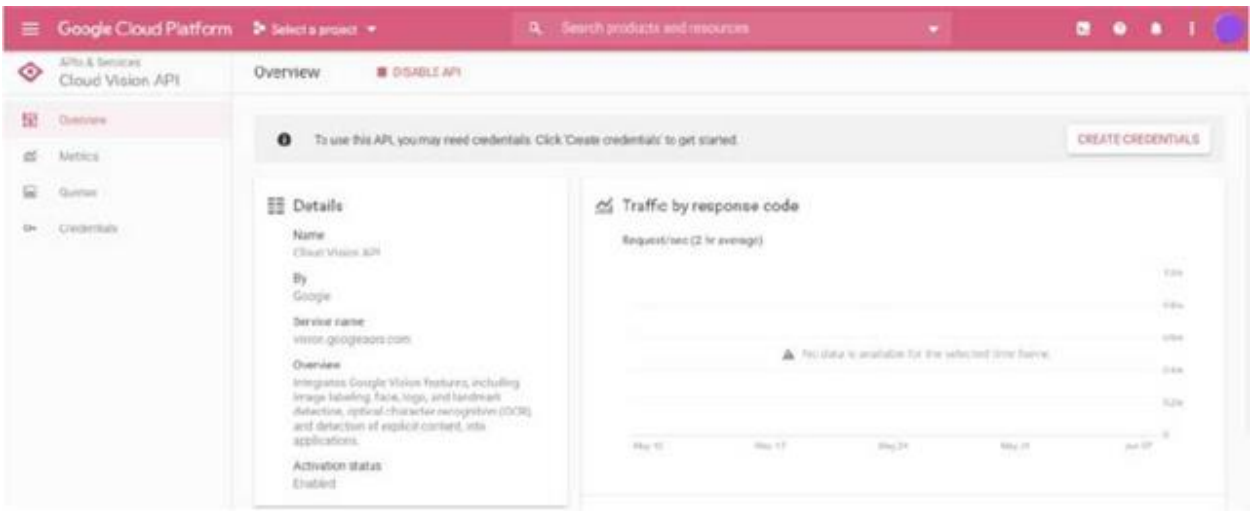

Рисунок 3.3 – Платформа Google

Кожна система, що взаємодіє через програмний інтерфейс, має реалізовувати механізм контролю доступу до ресурсів хмарної інфраструктури. Для забезпечення безпеки такої взаємодії необхідне створення унікального ключа аутентифікації. Процес ініціалізації ключа включає кілька обов'язкових кроків: вибір платіжної опції, активацію компонента системи розпізнавання образів, визначення імені користувача та рівня його доступу, а також надання секретного JSON-файлу, який містить необхідні облікові дані.

Ці дані дозволяють системі точно ідентифікувати користувача, підтвердити його повноваження й авторизувати доступ до відповідних API. Зазначений секретний файл використовується в якості ключа при здійсненні запитів до хмарного середовища, забезпечуючи безперебійну й захищену взаємодію з функціоналом розпізнавання зображень, зокрема через спеціальні форми завантаження.

Після надходження зображення документу задіюється спеціалізований механізм завантаження, який використовує функціональність бібліотеки Telegram для отримання графічного контенту. Отримане зображення трансформується у формат, придатний для подальшої обробки.

З метою розпізнавання символьної інформації, зображення надсилається до хмарної платформи комп'ютерного зору через відповідний

програмний інтерфейс. У результаті обробки система формує структурований файл, що містить дані про виявлені символи. Цей файл безпосередньо інтегрується в логіку серверного компонента Telegram-бота, який після отримання результату виконує аналіз вмісту та формує відповідь у зручному для користувача вигляді.

Рисунок 3.4 – Реалізація функції на Python

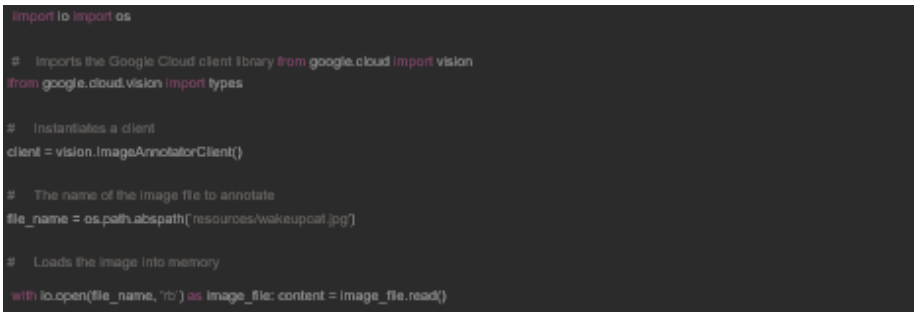

Рисунок 3.5 – Розпізнавання символів

Реалізована функція приймає фотографічне зображення, що містить текстову інформацію, та повертає структуровану відповідь у вигляді багаторівневої ієрархії: сторінка, блок, абзац, слово, символ. Кожен елемент

цієї структури містить координати обмежувальної рамки, текстовий вміст та рівень довіри до розпізнаного результату. Це дозволяє не лише точно локалізувати символи на зображенні, але й отримати кількісну оцінку надійності класифікації.

Після обробки вхідного зображення виконується автоматична класифікація символьних даних, розташованих у визначеній області машинного зчитування. Результати передаються у стандартизованому форматі, що зберігає як сам текст, так і метаінформацію, включно з рівнем упевненості системи щодо кожного розпізнаного елемента.

Для забезпечення коректності обробки критично важливо попередньо визначити та ізолювати область, яка підлягає розпізнаванню. Це досягається за допомогою маркерів, що слугують орієнтирами початку та завершення зони зчитування. Завершальним етапом є перевірка достовірності отриманих результатів, що забезпечує відповідність високим вимогам до точності та надійності, якими керуються користувачі та замовники автоматизованих систем розпізнавання.

Під час даної операції проходить перевірка корректности класифікації цифр в області МЧЗ використовується механізм затверджений відповідними нормативними документами (детальніше у розділі 1.2. Огляд МЧЗ області).

Для другої строки MRZ використовуються 5 контрольних цифр:

- сума для перевірки символів 0-8;
- сума для перевірки символів 13-18;
- сума для перевірки символів 21-26;
- сума для перевірки символів 28-41;
- сума для перевірки символів 0-9, 13-19 та 21-42.

Щоб виконати обчислення, які вимагають контролю за відповідними символами, був розроблений метод що реалізує функціонал відповідного нормативного документа.

Всі числа нумеруються по порядку:

- окремо цифри;

- після цього літери латинського алфавіту;
- після цього спеціальні символи.

В результаті всі символи на відповідних їм місцям проводиться добуток з відповідним коефіцієнтом. Формула для обчислення ваги дорівнює 2 * n + 1;

Після цього автоматично проводиться підрахунок контрольної Суми ми а так як ми використовуємо десяткову систему числення то для того щоб представити отримано інформацію у вигляді однієї цифри необхідно розділити отримано суму за модулем 10.

Після цього вступає в дію спеціальний метод який реалізує алгоритм перевірки правильності розпізнавання та повертає значення у булевій формі.

Програмне виправлення.

При передачі інформації від паспорту до серверної моделі, що проводить обробку даних трапляються помилки передачі інформації, виправлення яких є не завжди можливим, та приводить до помилок розпізнавання тексту.

Практичному розпізнавання символів відбувається передача інформації від не цифрової до цифрової форми під час цифрової обробки отриманих сигналів зазвичай накладаються певні шуми та завади, через це отриманий сигнал спотворюється та розпізнавання символів робиться ускладненим та іноді проходить із помилками.

Було створено перелік деяких випадків коли система найчастіше робить помилку при розпізнаванні зображення яке було спотворене три перетворення у цифрову форму таким чином було створено спеціальний механізм додавання збитковості до коду контрольної Суми що підвищує вірогідність правильного розпізнання та виявлення помилок.

Тож використовуючи спеціально розроблені методи які дозволяють коригувати помилку під час розпізнавання або хоча б знаходити наявність цієї помилки значно підвищує якість розпізнавання символів за допомогою оптичних методів.

Правила щоб були створені засновані на логіки тож якщо що модель намагається розпізнати ім'я людини та серед символів імені було знайдено число то імовірно розпізнавання відбулося із неточність у та потрібно повторно процедура оптичної класифікації символів використовуючи такі процедури та методи вдається значно покращити якість та зменшити навантаження на людину таксі більшість процесів було автоматизована.

З метою зменшення помилок, зроблених системою варто зазначити, що спеціальні мата данні, як наприклад, код держави складаються з трьох символів.

Користуючись наведеними фактами можна створити логіку яка допомагає виправляти помилки збільшує імовірністю ніж без неї.

Так як розроблено програмне забезпечення може працювати з державними документами різних типів використовується спеціальна розроблена схема, яка містить в собі всі можливі позначення багатьох країн Дана бібліотека являється стандартизованої та відповідною нормативним документом відповідних

Тож враховуючи вищевказані дані був скоригований метод алгоритм роботи при виявленні похибки розпізнавання чи класифікації символів, алгоритм роботи наступний якщо система знаходить код якого не існує є то алгоритм робить висновок що коду в розрізі невірне та вимагає повторного розпізнавання символів.

Також був розроблений алгоритм автоматичного виправлення помилок який працює наступним чином, у разі помилкового розпізнавання запускається автоматичний алгоритм саме корекції який представляє під часто помилкові символи дуже схожий на нього наприклад 0 та О у разі якщо один із символів підходить та контрольна співпадає вважається що авто корекція була виконана успішно це дозволяє знизити навантаження на систему автоматичного розпізнавання символів та зменшити витрати на користування цією системою так як вона знаходиться у хмари та вимагає оплати за кожний виклик функції.

Повернення вихідних даних .

Відразу як процедура коригування та автоматичної перевірки була завершена вихідні дані передають тому хто її запрошував.

Для виконання даної процедури використовується спеціальний механізм що за допомогою бібліотеки обгортки телеграм надсилає повідомлення реципієнту. Вихідна інформація дія чи передається користувачів має повне описання МЧЗ області, у разі неможливості розпізнати виводиться повідомлення про неможливість даної операції

Імплементація програмного інтерфейсу для роботи з системою

Для забезпечення функціонування системи у режимі автоматичному та не потребують чому участі людини було вирішено створити публічний програмний інтерфейс що реалізує функціонал оптичного розпізнавання символів за допомогою запитів за протоколом передачі гіпертексту. Даний програмний інтерфейс працює такий система на вхід він приймає є фото або сканований документ а на вихід повертає результати оптичних класифікації що містить рядок розпізнання державного документа.

Використовується спеціальний механізм реалізований на мові Пайтон.

3.2 Програмний інтерфейс

На рисунку 3.6 відображено користувацький інтерфейс веб-додатку для автоматичного розпізнавання живих істот, розробленого українською мовою. Програма являє собою спеціалізований інструмент для ідентифікації біологічних об'єктів на основі технологій комп'ютерного зору та машинного навчання.

Архітектурно додаток побудований на базі нейронної мережі MobileNet v2 у поєднанні з Face-API.js, що забезпечує ефективне розпізнавання понад тисячі видів живих істот, включаючи людей, тварин, птахів та інші біологічні об'єкти. Система підтримує широкий спектр графічних форматів, зокрема JPG, PNG та GIF з максимальним розміром файлу до 10 мегабайт, що

забезпечує універсальність використання різноманітних цифрових зображень.

Інтерфейс користувача організований навколо центральної зони завантаження, позначеної пунктирною рамкою та іконографічним зображенням силуетів людей, що інтуїтивно вказує на функціональне призначення програми. Реалізовано зручний механізм завантаження файлів як через традиційний вибір файлів, так і через функцію перетягування, що підвищує ергономічність взаємодії з системою.

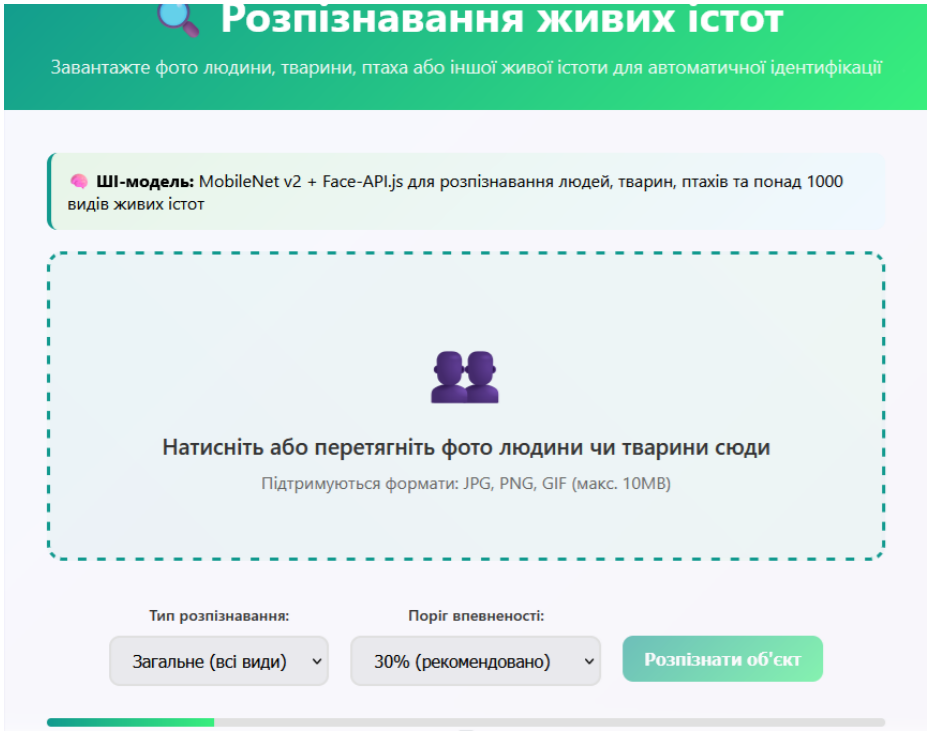

Рисунок 3.6 – Програмний інтерфейс

Додаток включає налаштування параметрів розпізнавання через випадаючі меню, що дозволяють користувачу обирати тип ідентифікації серед усіх доступних видів та встановлювати поріг впевненості на рівні тридцяти відсотків за замовчуванням, що забезпечує оптимальний баланс між точністю та чутливістю алгоритму розпізнавання. Візуальний індикатор прогресу в нижній частині інтерфейсу інформує користувача про стан обробки завантаженого зображення.

На рисунку 3.7 демонструються результати функціонування системи автоматичного розпізнавання живих істот після успішної обробки завантаженого біологічного зображення. Інтерфейс поділено на дві основні функціональні зони, що забезпечують комплексне відображення процесу аналізу та його результатів.

У лівій частині інтерфейсу розміщено завантажене користувачем зображення гекона, виконане у стилізованому графічному представленні з характерними анатомічними особливостями цієї рептилії, включаючи добре розвинені кінцівки з присосками, довгий хвіст та характерне забарвлення тіла. Під зображенням представлено ключові метрики ефективності роботи алгоритму, зокрема кількість виявлених на зображенні об'єктів, що становить п'ять одиниць, найвищий рівень впевненості системи у розпізнаванні, який досягає дев'яноста одного відсотка, та час обробки зображення, що склав одну цілу та одну десяту секунди.

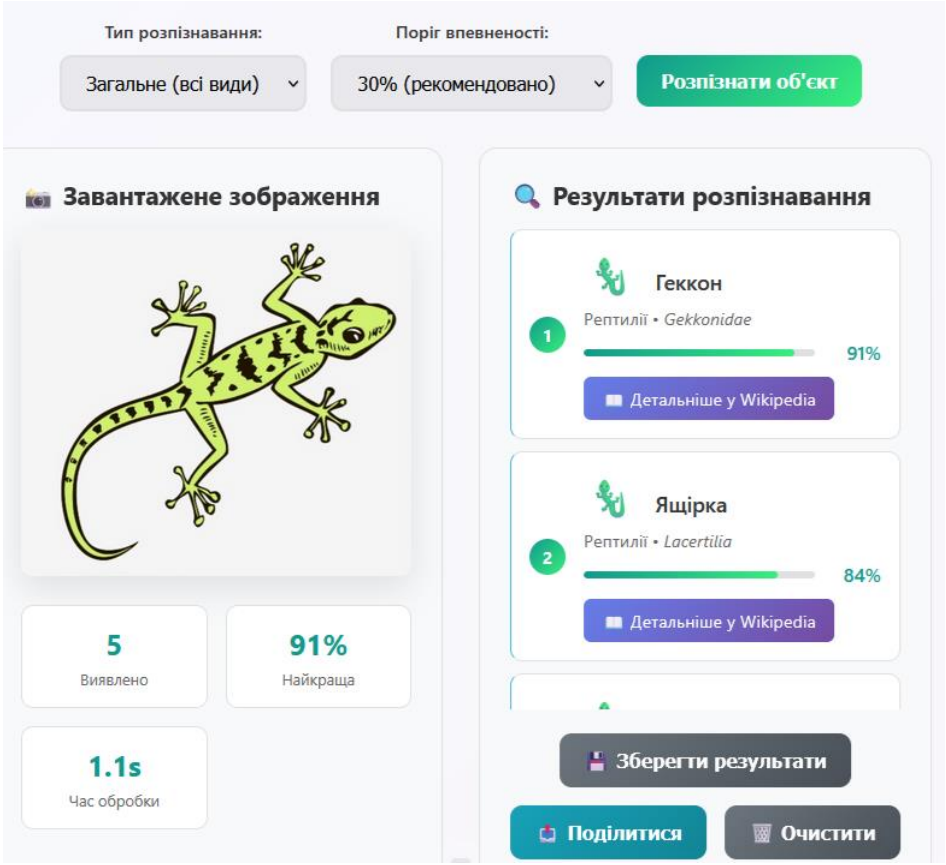

Рисунок 3.7 – Результати роботи

Права частина інтерфейсу містить деталізовані результати таксономічної ідентифікації, представлені у вигляді ранжованого переліку можливих видових визначень. Найвірогідніший результат ідентифікації вказує на приналежність об'єкта до гекона родини Gekkonidae з рівнем впевненості дев'яносто один відсоток, що супроводжується візуальним індикатором у вигляді прогрес-бару та іконографічним зображенням рептилії. Альтернативний варіант ідентифікації представляє ящірку роду Lacertilia з рівнем впевненості вісімдесят чотири відсотки.

Система інтегрована з зовнішніми інформаційними ресурсами, що забезпечується наявністю функціональних кнопок для отримання додаткової довідкової інформації з Wikipedia. Додатково реалізовано функції управління результатами, включаючи можливості збереження даних ідентифікації, поділитися результатами з іншими користувачами та очищення поточного аналізу для проведення нової ідентифікації.

ВИСНОВКИ

В ході реалізації даної кваліфікаційної роботи були розроблена комп’ютерна система класифікації даних з використанням мови Python. Проведено аналіз існуючих програмних засобів, аналіз алгоритмів розпізнавання текстів на зображеннях. Розроблені схема роботи програмних засобів розпізнавання. Проведено тестування розроблених програмних засобів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. W. Frawley, G. Piatetsky-Shapiro, C. Matheus Knowledge Discovery in Databases: An Overview. - AI Magazine. - 1992. pp. 213-228.

2. Kitchin Rob. The Data Revolution. United States: Sage. 2014, p. 6.

3. Piatetsky-Shapiro G, Frawley W J. Knowledge Discovery in Databases. USA: MIT Press, 1991.

4. Agrawal R., Mannila H., Srikant R., Toivonen H. and Verkamo I. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, eds. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Menlo Park, Calif.: AAAI Press, 1996, pp. 307-328.

5. Fayyad U., Piatetsky-Shapiro G., Smyth P., Advances in Knowledge Discovery and Data Mining, (Chapter 1), AAAI/MIT Press, 1996.

6. Judea Pearl, Stuart Russell. Bayesian Networks. UCLA Cognitive Systems Laboratory, Technical Report (R-277), November 2000.