

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»
в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим Хруслов
«___» червня 2025 р.

Пояснювальна записка до кваліфікаційної роботи бакалавра

на тему: «Модель технічного зору для розпізнавання монет»

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
Галузь знань 15 – Автоматизація та приладобудування
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

Захищено на засіданні
Екзаменаційної комісії № 46
протокол № ___ від __.06.2025 р.
Оцінка _____ / _____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
Студент групи КУ– 41
СТАРКОВ Єгор Володимирович 

Керівник: канд. техн. наук,
доцент ЗВО кафедри комп'ютерних систем та
робототехніки
БИКОВА Тетяна Володимирівна 

Рецензент: к.т.н., доцент кафедри
математичного моделювання та аналізу
даних

МАЗОРЧУК Марія Сергіївна 

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається із вступу, трьох розділів , висновків до кожного розділу , списку використаних джерел і трьох додатків. Загальний обсяг роботи становить 47 сторінок, серед них 33 сторінки основної частини містять 25 рисунків , 14 найменувань списку використаних джерел та 3 додатки.

Метою кваліфікаційної роботи є розробка моделі технічного зору для автоматизованого розпізнавання монет на основі нейронної мережі.

Об'єкт дослідження - процес розробки системи, яка здатна аналізувати та розпізнавати об'єкти на зображеннях у реальному світі.

Предметом дослідження є алгоритм глибокого навчання та методи комп'ютерного зору, які використовуються для класифікації та розпізнавання зображень монет.

Область застосування - розроблена модель може бути використана у нумізматиці, або у тих галузях, що включає в себе ідентифікацію монет.

Ключові слова : *технічний зір, розпізнавання монет, нейромережа, обробка зображень, класифікація, автоматизація.*

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, three chapters, conclusions to each chapter, a list of references and three appendices. The total volume of the work is 47 pages, including 33 pages of the main part containing 25 figures, 14 references and 3 appendices.

The purpose of the qualification work is to develop a model of technical vision for automated coin recognition based on a neural network.

The object of study is the process of developing a system that is able to analyze and recognize objects in images in the real world.

The subject of the study is the deep learning algorithm and computer vision methods that are used to classify and recognize coin images.

Scope - The developed model can be used in numzematics, or in those industries that includes the identification of coins.

Keywords: *technical vision, coin recognition, neural network, image processing, classification, automation.*

ЗМІСТ

ПЕРЕЛІК СКОРЕЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1 АНАЛІЗУВАННЯ ТА ФУНКЦІЮВАННЯ МОДЕЛІ ТЕХНІЧНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ МОНЕТ.....	8
1.1 АНАЛІЗ ТА ФУНКЦІЮВАННЯ МОДЕЛІ ТЕХНІЧНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ МОНЕТ.....	8
1.2 АНАЛІЗ АНАЛОГІВ МОДЕЛІ ТЕХНІЧНОГО ЗОРУ.....	9
1.3 ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ МОДЕЛІ РОЗПІЗНАВАННЯ МОНЕТ.....	12
Висновок до розділу 1.....	13
РОЗДІЛ 2 ПРОЄКТУВАННЯ СИСТЕМИ.....	14
2.1 ПОСТАНОВКА ЗАДАЧІ ПРОЄКТУВАННЯ.....	14
2.2 АРХІТЕКТУРА СИСТЕМИ.....	14
2.3 АЛГОРИТМ РОБОТИ СИСТЕМИ.....	17
2.4 ОСОБЛИВОСТІ НАВЧАННЯ НЕЙРОННОЇ МЕРЕЖІ	18
Висновок до розділу 2.....	20
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ.....	21
3.1 РЕАЛІЗАЦІЯ КЛІЄНТСЬКОЇ ЧАСТИНИ.....	21
3.2 РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ.....	25
3.3 ТЕСТУВАННЯ СИСТЕМИ.....	28
3.4 АНАЛІЗУВАННЯ ТЕСТУВАННЯ.....	31
Висновок до розділу 3.....	32
ВИСНОВКИ.....	34
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	35
ДОДАТКИ.....	37
Додаток А.....	37
Додаток Б.....	39
Додаток В.....	43

ПЕРЕЛІК СКОРЕЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

HTML — *HyperText Markup Language* — мова розмітки для створення веб-сторінок, яку інтерпретують браузери.

CSS - *Cascading style Sheets* - технологія оформлення HTML-документів.

CNN - *Convolutioanl Neural Network* - згорткова нейронна мережа.

JS - *JavaScript* - мова програмування для динаміки веб-сторінок і взаємодії з елементами інтерфейсу.

HTTP - *HyperText Transport Protocol* - протокол для обміну даними між клієнтом і сервером у мережі інтернет.

API - *Application Programming Interface* - набі інструментів для взаємодії з бібліотеками.

Matplotlib - бібліотека Python, що спеціалізується на побудові графіків і візуалізації даних.

TF - *TensorFlow* - це фрейворк для машинного навчання, який спеціалізується на глибокому навчанні.

ВСТУП

Актуальність роботи. Останнім часом комп'ютерний зір набуває більшого значення як прогресивна технологія. Він дає змогу комп'ютерам аналізувати зображення, забезпечуючи їх здатністю “бачити” навколишній світ. Завдання, які виконує машинний зір, побідні до тих, що виконує людський мозок за допомогою зору. У процесі обробки зображень і відео комп'ютерне бачення широко застосовує нейронні мережі [1].

Розпізнавання монет, наразі, є актуальним напрямком, щоб застосувати технічний зір, котрий є практичним у таких галузях як: нумізматики, банківські справи, колекціонерів. Автоматичне розпізнавання монет за зображенням, що людина надає, надає швидке надання інформації про монету : рік випуску, номінал , а також вартість даної монети, це значно спрощує процес.

Модель технічного зору на основі нейронної мережі, дозволяє підвищити точність і швидкість обробки, зменшуючи помилки. Таке використання надає нові можливості для створення мобільних застосунків та веб-сайтів про розпізнавання монет. Це робить дану тему актуальною та перспективною на сьогодення.

Метою кваліфікаційної роботи є розробка моделі технічного зору для автоматизованого розпізнавання монет на основі нейронної мережі.

Об'єкт дослідження - процес розробки системи, яка здатна аналізувати та розпізнавати об'єкти на зображеннях у реальному світі.

Предметом дослідження є алгоритм глибокого навчання та методи комп'ютерного зору, які використовуються для класифікації та розпізнавання зображень монет.

Розв'язання завдань у процесі кваліфікаційної роботи

1. Аналізування методів технічного зору , що допоможуть розпізнавати зображення.

2. Збирання та підготовка датасетів зображень монет однакових за номіналом та різних по рокам випуску.
3. Використання аугментації для підвищення якості розпізнавання.
4. Будування моделі з використанням бібліотеки TensorFlow.
5. Навчання моделі за певним датасетом, котрий складається з класів монет .
6. Проведення тесту для з'ясування ефективності моделі.

РОЗДІЛ 1.

АНАЛІЗУВАННЯ ТА ФУНКЦІЮВАННЯ МОДЕЛІ ТЕХНІЧНОГО ЗОРУ ДЛЯ РОЗПІЗНАВАННЯ МОНЕТ

1.1 Аналізування та функціювання моделі технічного зору для розпізнавання монет

Останнім часом комп'ютерний зір набуває великого значення в інноваційних технологіях. Комп'ютерний зір дозволяє комп'ютерам аналізувати зображення, надаючи їм “бачити” світ. Комп'ютерний зір виконує функції, подібні до тих, які виконує людський мозок через зір. У комп'ютерному зорі для обробки зображень активно використовуються нейронні мережі.

Залежно від сфери застосування, системи комп'ютерного зору можуть суттєво відрізнятися. Деякі функціонують автономно та вирішують конкретні завдання, а інші є частиною більших систем. Наприклад, це можуть бути системи, що керують базами даних або забезпечують взаємодію між людиною та комп'ютером [1].

Існують загальні етапи, які проходить більшість систем комп'ютерного зору:

1. Отримання зображень

Це перший крок, коли дані надходять від користувача. Це можуть бути 2-D зображення.

2. Попередня обробка

Щоб приступити до аналізу, важливо підготувати зображення: покращити якість, привести до потрібного масштабу або орієнтації. Цей етап важливий, бо якість попередньої обробки суттєво впливає на подальші результати.

3. Виділення особливостей

З зображення виокремлюють значущі елементи: лінії, межі, кути, форми об'єктів.

4. Сегментація

На даному етапі приймається рішення, які області зображення мають значення. Наприклад, можна виокремити конкретний об'єкт або ділянку, яка представляє інтерес для подальшого аналізу.

5. Високорівнева обробка

На завершальному етапі система аналізує об'єкти, проводить їхню класифікацію. Усе це дозволяє розуміти, що саме зображено, і застосувати ці дані у реальних завданнях [2].

1.2 Аналіз аналогів моделі технічного зору

Numista — вебсайт, що містить каталог монет, жетонів, паперових грошей та екзонумії. Сервіс дозволяє колекціонерам (нумізматам) зберігати та ділитися інформацією про свою колекцію, дізнатися приблизну ринкову вартість колекції, та доповнювати її шляхом обміну з іншими користувачами [3].

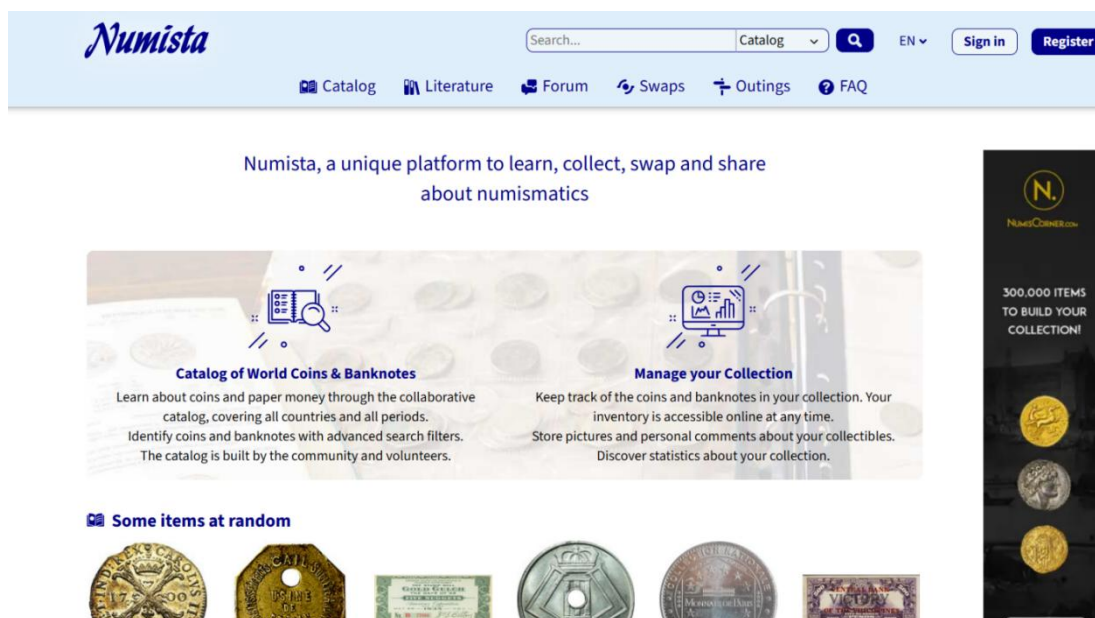


Рисунок 1.1.- Меню вебсайту Numista.

У данному вебсайті не має автоматичного розпізнавання, тільки зручний пошук монет з каталогами.

Maktun — висококласний додаток для нумізматів, філателістів і любителів банкнот. Застосунок має в собі багато функцій: збереження зображень ваших монет та банкнот. Дозволяє знайти детальну інформацію о монетах та банкнотах : країну , номінал, роки випуску, вагу , розмір [4].

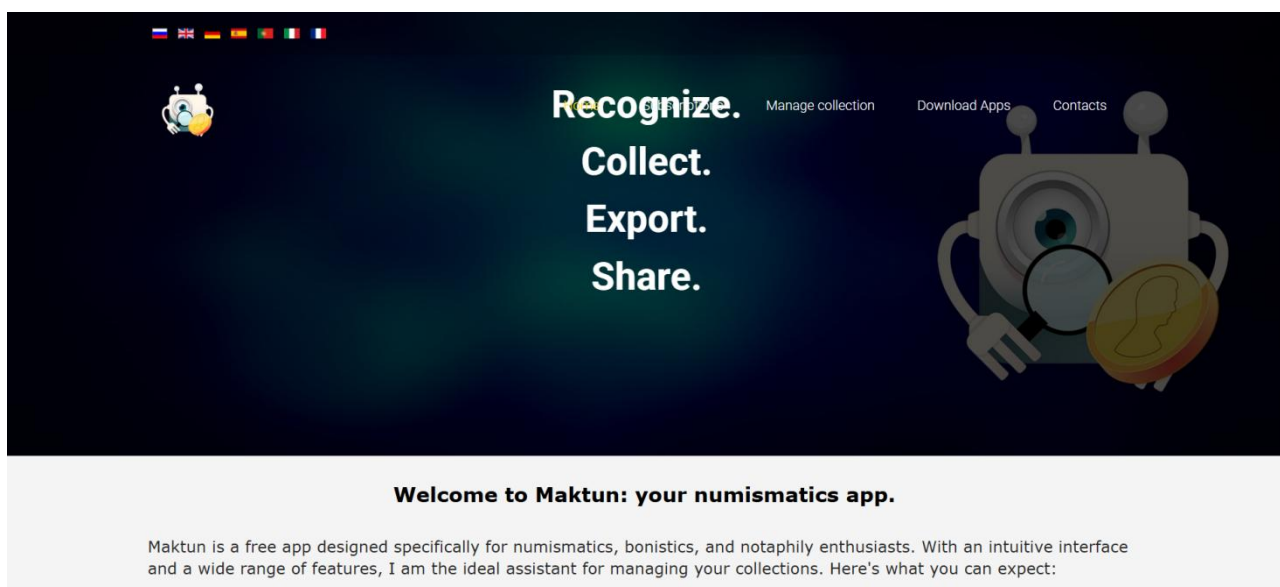


Рисунок 1.2. - Меню застосунку Maktun.

CoinSnap - застосунок, що допоможе визначити вартість монети за допомогою експертної інформації та ринкових даних у реальному часі. На основі штучного інтелекту за лічені секунди надасть вам детальну інформацію , рівні рідкості [5].

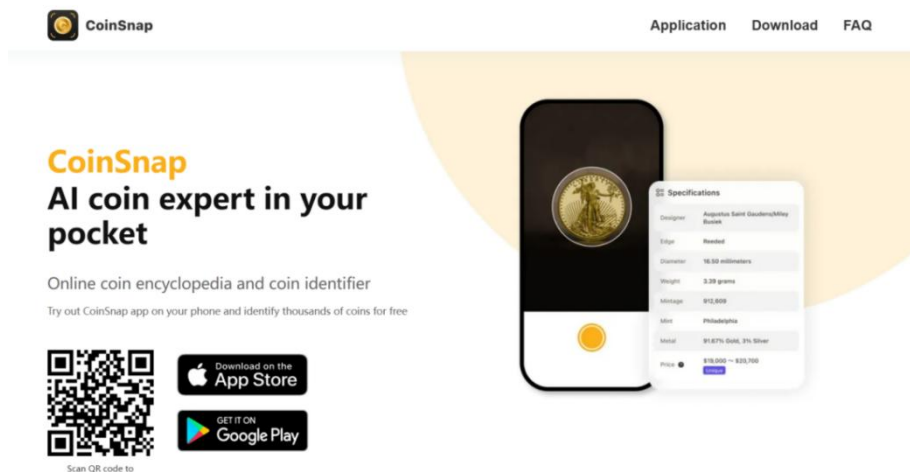


Рисунок 1.3 - Меню вебсайту CoinSnap.

Google Lens — технологія розпізнавання зображень, розроблена Google, покликана збирати відповідну інформацію, що стосується об'єктів, які вона ідентифікує за допомогою візуального аналізу на основі нейронної мережі [6].

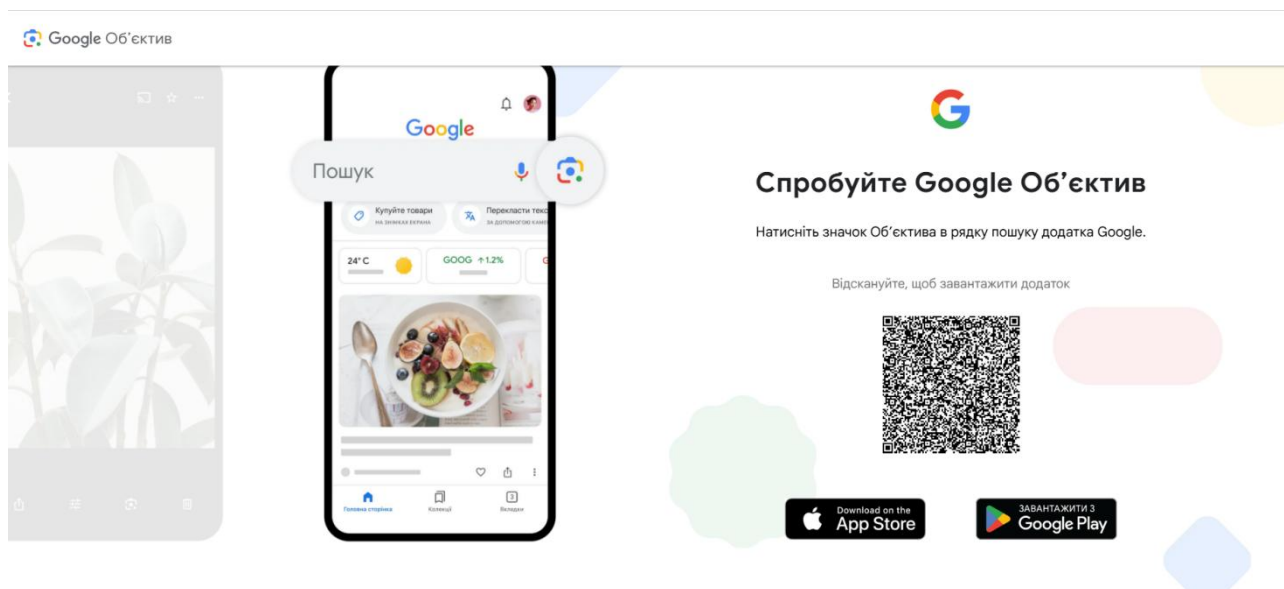


Рисунок 1.4. - Меню вебсайту Google Lens.

Мобільний застосунок **Coinoscope** робить ідентифікацію монет та оцінювання простим — просто зробіть фото монети на телефон і застосунок покаже список схожих монет. Це швидко та точно. Незамінний інструмент для кожного колекціонера та нумізмата [7].

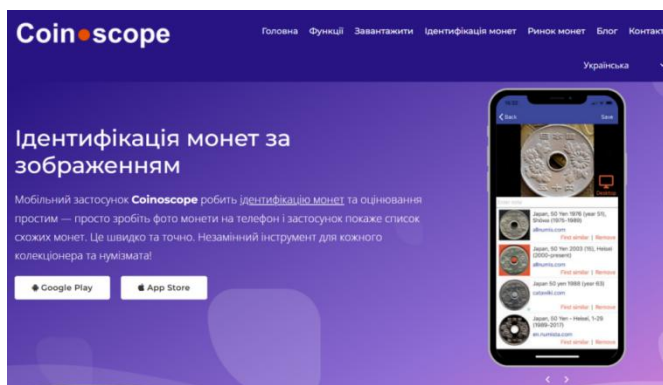


Рисунок 1.5 - Меню вебсайту Coinoscope.

Дані додатки мають спільну мету - надати детальну інформацію стосовно завантаженої монети. Але вони мають незмінні бази даних, що можуть рідко оновлюватися. Не має можливості власноруч додати зображення монет. Це може знизити ефективність розробленої системи, для пошуку рідкісних або недавно випущених монет. Розширення бази даних (датасетів), забезпечує більш гнучке розпізнавання.

1.3 Вибір інструментів для реалізації моделі розпізнавання монет

Серверна частина :

Python - це багатофункціональна мова програмування високого рівня, маючи інтерпретовану , об'єктно-орієнтовану структуру та динамічну семантику, створює сценарії і розробки різноманітних застосунків, що робить її зручною [8].

Matplotlib - це бібліотека для візуалізації даних у Python , яка забезпечує функціональність, подібну до графічних засобів MATLAB, і дозволяє створювати графіки та діаграми [8].

Keras - це високорівнена надбудова над TensorFlow, яка спрощує створення, налаштування та навчання моделей глибокого навчання, дозволяючи ефективно вирішувати багато завдань , пов'язаних із використанням TensorFlow [9].

Pathlib — модуль, що пропонує класи, що представляють шляхи файлової системи із семантикою, відповідною до різних операційних систем.

Клієнтська частина:

HTML — HyperText Markup Language — основна мова розмітки, яка використовується для створення веб-сторінок. HTML-документи обробляються веб -браузерами та відображаються на екрані у зручному для користувача форматі [10].

JS - скриптова мова програмування, яка базується на прототипному підході. Зазвичай її використовують як вбудовану мову для програмного керування об'єктами додатків.

CSS - Cascading style Sheets - стильові специфікації з ієрархічною структурою, які мають унікальний синтаксис, будову та призначення [11].

Датасетом уявляє собою класи монет за роками випуску, дані класи мають певну кількість зображень.

Ці інструменти допоможуть детально обучити нейронну мережу, показуючи графіки навчання на основі датасету. Також для покращення гнучкості та ефективності нейронної мережі.

Висновки за розділом 1

У розділі 1 проведено детальний аналіз комп'ютерного зору для розпізнавання монет. Запевнено, комп'ютерний зір об'єднаний з нейронної мережею надає високу точність розпізнавати деталі на зображеннях.

У розгляді були аналоги системи технічного зору для монет, такі як Numista, CoinSnap, Maktun Google Lens та Coinoscope. Зроблено висновок, що деякі системи мають незмінні бази даних, що сповільнить роботу з рідкими та новими монетами.

Проведен аналіз з вибору інструментів для реалізації моделі технічного зору: для серверної частини обрана мова програмування Python, що більше підходить для навчання нейромережі. Також бібліотеки TensorFlow, Keras, Matplotlib та Pathlib, які допоможуть детально розібратися в навчанні та показати процес навчання. Для клієнтської частини обрані HTML, CSS, JavaScript, що дозволять створити зручний інтерфейс для користувача.

Важливу роль грає датасет у вигляді зображень монет, які класифіковані за роками випуску.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Постановка задачі проєктування

Метою проєктування полягає у створенні ефективної моделі технічного зору, котра здатна розпізнавати монети, зображень, завантажених користувачем. Результатом розпізнавання служить визначення назви, року випуску та вартість монет. Для цього треба визначити ряд задач, що пов'язані з навчанням нейронної мережі та детальною обробкою зображень.

Функціональні вимоги :

- Завантаження зображень монет користувачем;
- Автоматичне визначення номіналу, року випуску та вартість;
- Надання зручного інтерфейсу користувачу;
- Можливість додавання зображень в датасети, без перебудови структури.

Нефункціональні вимоги:

- Висока точність розпізнавання, більше 90%;
- Швидке надання результату користувачеві;
- Зручне використання додатку з різних пристроїв;

Основним завданням проєктування є визначення структури моделі, роботи з зображеннями за допомогою алгоритмів, визначення інструментів розробки: мова програмування, допоміжні бібліотеки. Розробка клієнтської та серверної розробки. Але найважливішим етапом є підготовка, тестування та навчання датасету.

2.2 Архітектура системи

Архітектура моделі, яка використовується для розпізнавання монет, розробляється на основі згорткової нейронної мережі. Такий тип мережі є 15

спеціалістом у роботі з зображеннями, допомагаючи навчитися розпізнавати риси об'єктів на зображенні.

Згорткові нейронні мережі (CNN) є однією з найпоширеніших архітектур глибинного навчання, оскільки ефективно використовують просторові взаємозв'язки між ознаками зображень, що дозволяє витягнути важливу інформацію, приховану в візуальних даних.

Методи глибинного навчання, як правило, потребують великих обсягів даних, щоб успішно узагальнювати нові приклади та розпізнавати найменші деталі в інформації, що подаються.

Використання згорткової нейронної мережі є основною для ідентифікації зображень.

У структурі згорткової нейронної мережі основними типами шарів є:

- Згорткові шари(convolutional layers)
- Пулінгові шари (pooling layers)
- Повнозв'язані шари (fully connected layers)

Кожен наступний шар у CNN поглиблює розуміння зображення. На початкових етапах мережа зосереджується на виявленні простих ознак, таких як краї та кольори. Подалі проходження через мережу обробка ускладнюється, це дозволяє моделі розпізнавати складні структури.

Типовим прикладом застосування CNN є класифікація зображень, де мережа отримує на вході зображення, а потім визначає до якої категорії воно належить [1].

Компоненти архітектури:

1. Інтерфейс для користувача

Інтерфейс відіграє важливу роль у взаємодії моделі з користувачем. Завдяки зручного та зрозумілого інтерфейсу користувач може легко завантажувати зображення монет, отримувати результати розпізнавання певної монети. Це значно підвищить ефективність роботи користувачів, завдяки швидкого отримання результату.

Інтерфейс побудований за допомогою HTML, CSS та JavaScript, які значно спрощують роботу з користувачем.

Функції, що притаманні інтерфейсу:

- Завантаження зображень користувачем;
- Перегляд результатів користувачем;
- Отримання помилки, при отриманні невідомої монети, або вартості ясної монети.

2. Серверна частина

Серверна частина системи використовується для обробки запитів (зображень) зі сторони клієнта. Використовуючи бібліотеку TensorFlow, серверна частина передає зображення у нейронну мережу. Виконується порівняння зображення, задане користувачем, з класами датасету. Серверна частина розроблена за допомогою мови програмування Python та бібліотек TensorFlow, Keras, Pathlib, Flask . TensorFlow, Keras призначені для реалізації та навчання нейронної мережі, допомагаючи виконати класифікацію монет за певними параметрами такими, як номінал , рік випуску та вартість. Бібліотека Pathlib використовується для зручного управління та з'єднання з датасету. Matplotlib - бібліотека, що значно допоможе відслідкувати навчання нейронної мережі у вигляді графіків прийняття та втрат зображень в датасеті.

3. База даних

База даних основана на зборі зображень, які згруповані за певним класом. Кожен клас відповідає певному виду монети - наприклад, 50 копійок 2018 року , або 50 копійок 1996 року випуску. Така структура значно облегчить та підвищить ефективність навчання нейронної мережі.

2.3 Алгоритм роботи системи

У комп'ютерному зорі ключову роль відіграють два основні типи алгоритмів - алгоритми виділення ознак та алгоритми класифікації. Перші відповідають за виявлення важливих характеристик зображення, таких як краї, кути, кольорові ділянки, що дає змогу системі “побачити” об’єкти на зображенні. Алгоритми класифікації, своєю чергою, призначені для виявлення об’єкту до певної категорії або класу.

Але методи обробки зображень мають низку недоліків. Наприклад при визначенні об’єкта система може ідентифікувати одразу кілька об’єктів разом із їх розташуванням. Крім того, однією з проблем залишається недостатньо висока точність під час розпізнавання та порівняння параметрів, що ускладнює точну ідентифікацію певних об’єктів.

Алгоритм роботи системи розпізнавання монет використовують поетапні дії, від завантаження зображення користувачем до отримання результату розпізнавання. Система має побудову за принципом клієнт-сервер. В клієнтській частині інтерфейс виконує взаємодіючу роль, серверна частина здійснює обчислення та логіку роботи [1].

Послідовність кроків алгоритму роботи системи:

1. Завантаження зображення користувачем:

Через інтерфейс користувач обирає зображення монети для завантаження та натискає кнопку завантажити. Після цього зображення передається на серверну частину.

2. Передача зображення на сервер:

Клієнтська частина надсилає зображення до серверної частини. Сервер приймає файл та зберігає його у тимчасовій пам’яті.

3. Обробка зображення :

Отримане зображення проходить обробку, зменшення розміру(128x128), зміна контрасту. Це потрібно для перетворення зображення у формат, який підтримує нейронна мережа.

4. Передача зображення до нейронної мережі:

Після підготовки зображення, воно подається в модель машинного навчання за допомогою TensorFlow / Keras. Модель виконує аналізування зображення та класифікує його на основі датасетів.

5. Обробка результату класифікації

Результатом роботи моделі є проходження зображення по кожному класу в датасетах. Маючи найвищу ймовірність, клас визначається як відповідь системи. Також результатом служить рік випуску, номінал та вартість певної монети.

6. Відправка відповіді на клієнтський інтерфейс:

Сервер повертає відповідь до інтерфейсу, де вона обробляється та виводиться у вигляді зручного та зрозумілого для користувача інформаційного блоку.

Саме з таким алгоритмом можна досягти стабільності та точності роботи системи, надаючи швидкі результати користувачу.

2.4 Особливості навчання нейронної мережі

В ході створення системи технічного зору для розпізнавання монет важливим кроком є навчання нейронної мережі. Завдяки якісного навчання, точність та надійність моделі набувають високих оцінок при класифікації зображень монет, однакових за номіналом, але різних по рокам випуску.

1. Підготовка база даних для навчання :

Для більш якісного навчання нейронної мережі, кожен клас повинен мати 1000 - 1500 зображень. Важливо знати, що кожне зображення повинно відповідати певному типу монети. Монети можна класифікувати за : номіналом, роком випуску.

2. Аугментація даних :

Не маючи багатого датасету, використовується аугментація. Вона збільшує обсяг навчальних даних та підвищує стійкість моделі до зовнішніх

змін(освітлення, кут нахилу тощо). Збільшення кількості зображень виконується шляхом:

- Обертання зображення на випадкові шляхи;
- Горизонтальні або вертикальні віддзеркалення;
- Зміни яскравості чи контрасту;
- Масштабування.

Процес навчання:

Навчання нейронної мережі здійснюється із використанням бібліотек TensorFlow та Keras.

Основні етапи навчання:

1.Вибір метриків для оцінки якості:

Метрик ассурасу відображає загальну кількість правильних класифікацій, показуючи частку передбачених зразків від загальної кількості всіх прогнозів [12].

2.Встановлення кількості епох:

Встановлення кількості епох може залежати від кількості зображень. Тобто епоха - це повних прохід з урахуванням розміру та складності даних. Малий набір даних призводить до недонавчання, занадто великий набір даних - до перенавчання.

3.Валідація моделі

Валідація - це етап у створенні моделі машинного навчання, який забезпечує відповідність моделі даним і допомагає уникнути перенавчання [13].

4.Побудова графіків навчання :

За допомогою бібліотеки Matplotlib будуються графіки залежності точності та втрат кількості епох. Такі графіки дозволяють виявити етапи недонавчання або перенавчання.

5.Оцінка якості моделі :

Використовуючи метрик Ассурасу, оцінюється модель , що закінчила навчання. Метрик показує наскільки модель точна та ефективна при розпізнанні зображення.

Висновки за розділом 2

Розділ 2 був присвячений детальному аналізу процесу проектуванню системи технічного зору для розпізнавання монет. Детально розглянуто основні функціональні та нефункціональні вимоги для розробки системи. Визначено архітектуру моделі з використанням згорткової нейронної мережі.

Вироблено структуру, де клієнтська та серверна частини взаємодіють. Також описана послідовність обробки зображення - від завантаження до видачі результату користувачу на серверній частині. Показано алгоритм роботи системи, який допоможе забезпечити високу точність в процесі класифікування та зручність у використанні.

Звернено увагу приділенню процесу навчання нейронної мережі: збір датасету, вибору оптимізатору, метрик точності. Розглянута робота валідації та візуалізація ефективності навчання, яка допомагає аналізувати якість моделі на кожній епосі.

Результатом проектування є створення високоякісної та ефективної системи, яка має змогу точно та швидко класифікувати зображення монети.

РОЗДІЛ 3 .

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1 Реалізація клієнтської частини

Клієнтська частина системи для розпізнавання монет реалізована у вигляді веб-сторінки

Використовуючи HTML, CSS та JavaScript, розроблена клієнтська частина системи для розпізнавання монет. Інтерфейс має зручну та максимально зрозумілу побудову.

Структура сторінки HTML

Завдяки HTML, забезпечується каркас інтерфейсу - заголовок, інструкція для користувача, форма завантаження зображення, та блок виводу результатів.

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Розпізнавання монет</title>
  <style>
```

Рисунок 3.1.- Заголовок інтерфейсу.

```
<div class="instructions">
  <h3 style="margin-top: 0;">Як користуватися:</h3>
  <ul style="padding-left: 20px; line-height: 1.6;">
    <li>Натисніть <b>«Оберіть файл»</b>, щоб завантажити фото монети з вашого комп'ютера або телефону.</li>
    <li>Файл повинен бути у форматі JPG, PNG або інший графічний формат.</li>
    <li>Після вибору зображення натисніть кнопку <b>«Завантажити»</b>.</li>
    <li>Система автоматично розпізнає монету і покаже результат нижче.</li>
  </ul>
</div>
```

Рисунок 3.2 - Інструкція для користувача.

```

<form action="/" method="post" enctype="multipart/form-data">
  <label for="file-upload" class="custom-file-upload">
    Оберіть файл
  </label>
  <input id="file-upload" type="file" name="file" accept="image/*" required>
  <div id="file-name" style="margin-top: 10px; color: #555;">Файл не обрано</div>
  <button type="submit">Завантажити</button>
</form>

```

Рисунок 3.3. - Форма завантаження зображення.

```

{% if result %}
<div class="result">{{ result|safe }}</div>
{% endif %}

```

Рисунок 3.4.- Вивід результатів розпізнавання.

Оформлення інтерфейсу CSS

Встановлення кольору, відступи, тіні, шрифти - це все здійснено з використанням CSS.

```

<style>
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  background-color: #f2f2f2;
  margin: 0;
  padding: 0;
}

.container {
  max-width: 500px;
  margin: 50px auto;
  padding: 30px;
  background-color: #ffffff;
  box-shadow: 0 4px 10px rgba(0,0,0,0.1);
  border-radius: 12px;
  text-align: center;
}

```

Рисунок 3.5.- Дизайн інтерфейсу.

JavaScript: Виведення імені файлу

Реалізовано виведення назви обраного файлу за допомогою JavaScript

```
<script>
  const fileInput = document.getElementById('file-upload');
  const fileName = document.getElementById('file-name');

  fileInput.addEventListener('change', function () {
    fileName.textContent = this.files.length > 0 ? this.files[0].name : 'Файл не обрано';
  });
</script>
```

Рисунок 3.6.- Відображення обраного файлу користувачем.

Це покращить зручність використання, інформуючи користувача про обраний файл.

Таким чином, клієнтська частина стає легкою та зручною для використання, забезпечуючи функції:

- Завантаження зображення монети;
- Інтерактивний показ назви обраного файлу;
- Виведення результату розпізнавання або повідомлення про помилку.

Результати інтерфейсу клієнтської частини :

Завантаження зображення монети

Як користуватися:

- Натисніть «Оберіть файл», щоб завантажити фото монети з вашого комп'ютера або телефону.
- Файл повинен бути у форматі JPG, PNG або інший графічний формат.
- Після вибору зображення натисніть кнопку «Завантажити».
- Система автоматично розпізнає монету і покаже результат нижче.

Оберіть файл

Файл не обрано

Завантажити

Рисунок 3.7. - Форма завантаження.

Завантаження зображення монети

Як користуватися:

- Натисніть «**Оберіть файл**», щоб завантажити фото монети з вашого комп'ютера або телефону.
- Файл повинен бути у форматі JPG, PNG або інший графічний формат.
- Після вибору зображення натисніть кнопку «**Завантажити**».
- Система автоматично розпізнає монету і покаже результат нижче.

Оберіть файл

2016.jpg

Завантажити

Рисунок 3.8. - Показ назви обраного файлу.

Завантажити



Результат: Монета 2016 року
Вартість : 1- 250 гривень

Рисунок 3.9.- Виведення результату розпізнавання.

3.2 Реалізація серверної частини

Серверна частина системи виконує: обробку зображення, класифікацію за допомогою нейронної мережі та формування відповіді, що отримує користувач на клієнтській частині. Дана частина реалізована мовою програмування Python з використанням бібліотек : Flask, TensorFlow, Keras,os,uuid.

1. Веб-фреймворк Flask

Основа серверної частини становить веб-фреймворк Flask, який дозволяє створювати маршрути, обробляти запити.



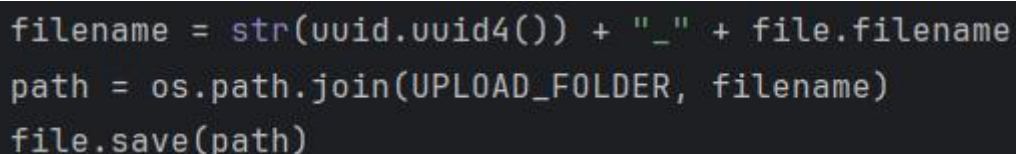
```
@app.route("/", methods=["GET", "POST"])
def index():
    result = ""
    filename = ""
    if request.method == "POST":
        file = request.files["file"]
        if file:
            return render_template("index.html", result=result, filename=filename)
```

Рисунок 3.10.- Обробка запитів.

Цей фрагмент кода визначає головний маршрут , який обробляє GET- , POST- запити. У випадку POST-запиту відбувається завантаження та обробка зображення.

2. Завантаження та збереження зображення

Після того як користувач надсилає зображення, сервер зберігає його у спеціальну папку з унікальним ім'ям, яке генерується через модуль uuid.



```
filename = str(uuid.uuid4()) + "_" + file.filename
path = os.path.join(UPLOAD_FOLDER, filename)
file.save(path)
```

Рисунок 3.11.-Збереження зображення у папку.

3. Обробка зображення

Перед подачею у модель, зображення приймає потрібний розмір, який відповідає розмір, що очікує модель.

```
img = tf.keras.utils.load_img(path, target_size=(img_height, img_width))
img_array = tf.keras.utils.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)
```

Рисунок 3.12.- Обробка зображення.

4. Завантаження та використання моделі

Модель, що створена на базі `tf.keras.Sequential`, спочатку завантажується зі збереженням вагами з файлу `model.keras`. Після цього вона використовується для класифікації нового зображення.

```
model = tf.keras.Sequential([
    layers.Rescaling(1. / 255, input_shape=(img_height, img_width, 3)),
    layers.RandomFlip("horizontal", input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

Рисунок 3.13.- Використання та збереження моделі.

5. Формування результату

Інформацію про монету виводиться у зручному текстовому форматі. Окрім року, додається вартість з попередньо створеного словника.

```
coin_prices = {
    "1992": "50-12000 гривень",
    "1994": "50-14000 гривень",
    "1995": " 5-15    гривень.",
    "1996": "100-1000 гривень",
    "2001": "1500-2000 гривень",
    "2003": "2000-3000 гривень",
    "2006": "30 - 60 гривень",
    "2007": "30 - 60 гривень",
    "2008": "1-2 гривні ",
    "2009": "1-2 гривні.",
    "2010": "1-2 гривні ",
    "2011": "2500 -3500 гривень",
    "2012": "2000-3000 гривень",
    "2013": "1 -250 гривень",
    "2014": "1 -250 гривень.",
    "2016": "1- 250 гривень",
    "2018": "1- 200 гривень",
    "2023": "1-2 гривні"
}
```

Рисунок 3.14.- Інформація про монети

```
price = coin_prices.get(predicted_class, "Невідома ціна ")
result = f"Результат: <b>Монета {predicted_class} року</b><br>Вартість : <b>{price}</b>"
```

Рисунок 3.15.- Вивід інформації про завантаженну монету.

Таким чином, серверна частина забезпечує повноцінну обробку запитів з моменту отримання зображення до виводу результату розпізнавання. Вона легко інтегрується з клієнтською частиною через шаблони Jinja2.

3.3 Тестування системи

Процес тестування нейронної мережі проводиться після завершення навчання моделі розпізнавання монет. Для цього було зібрано та використано набір з 7650 зображень, згрупованих у 18 класів, де кожен клас відповідає певному року випуску монети.

1. Налаштування тестування

Загальний обсяг даних: 7650 зображень

Співвідношення даних для навчання і валідації:

80% (6120 зображень) - тренувальна бівірка;

20% (1530 зображень) - валідаційна вибірка.

```
train_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_dir,
    validation_split=0.2,
    subset="training",
    seed=123,
    image_size=(img_width, img_height),
    batch_size=batch_size
)

vali_ds = tf.keras.utils.image_dataset_from_directory(
    dataset_dir,
    validation_split=0.2,
    subset="validation",
    seed=123,
    image_size=(img_width, img_height),
    batch_size=batch_size
)
```

Рисунок 3.16. - Процес навчання та валідації

2. Результати тестування

Модель тренувалася протягом 15 епох з використанням оптимізатора Adam і функції втрат SparseCategoricalCrossentropy.

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy']
)
```

Рисунок 3.17. - Результати тестування.

Точність моделі:

Максимальна точність на тренуванні: 99%

Точність на валідації: 92,2%

Кількість епох: 15

3.Графік точності та втрат

Для представлення результатів навчання створено графіки :

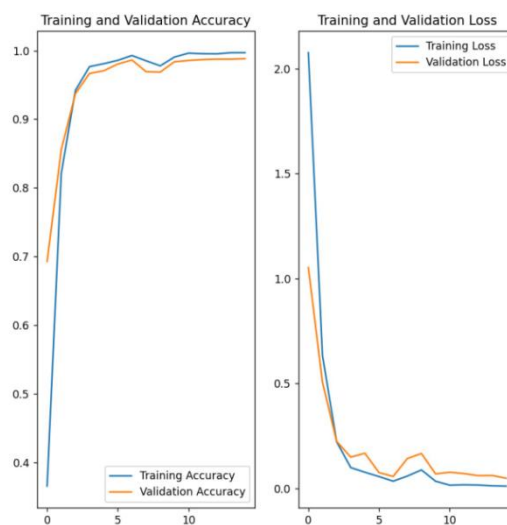


Рисунок 3.18 - Графіки Training and Validation Accuracy та Training and Validation Loss.

```

loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(epochs)

plt.figure(figsize=(8, 8))
plt.subplot(1, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(1, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()

```

Рисунок 3.19. - Показ графіків навчання моделі.

Модель чітко демонструє стабільне навчання без ознак перенавчання. Валідаційна точність залишається високою протягом епох.

4.Оцінка стабільності

Модель стабільно обробляє різноманітні зображення монет. Випадків збоїв або обривів навчання не зафіксовано. Завдяки кешуванню та попередньому завантаженню (`.cache()` та `.prefetch()`), навчання та валідація виконуються ефективно навіть при великій кількості зображень.

```

train_ds = train_ds.cache().shuffle(1000).prefetch(buffer_size=AUTOTUNE)
vali_ds = vali_ds.cache().prefetch(buffer_size=AUTOTUNE)

```

Рисунок 3.20. - Кешування моделі.

3.4 Аналізування результатів

Після проведення навчання та тестування моделі було здійснено якісний аналіз результатів. Оцінюються типи монет, яких система розпізнає з високою точністю, а які - з нижчою. Виявити фактори, що впливають на ефективність класифікації.

1. Високо розпізнанні монети

Найвищу точність класифікації система показала для класів:

- 1992, 1996, 2003,2008,2009,2010,2011,2012,2013,2016,2018,2023 - точність перевищувала 95%.
- Ці монети мають чітко виражені візуальні особливості.

Причина високої точності:

- Велика кількість якісних зображень у навчальному наборі.
- Мала схожість з іншими роками випуску.

2. Монети, які розпізнаються гірше

Найнижчі показники точності спостерігались у класах:

- 1994,1995,2001,2006,2007,2014 - точність у межах 50 - 75%.

Причини можливих помилок:

- Візуальна подібність між монетами різних років.
- Погана якість монет.
- Вплив зовнішнього освітлення.
- Часткова відсутність або нерівномірність зображення.

3. Причини помилок розпізнавання

- Нечіткі або розмиті зображення, зроблені з недостатнім фокусом.
- Відблиски на металевій поверхні монети.
- Погане освітлення або занадто темне тло, що ускладнює виявлення контуру монети.
- Частково обрізані монети.
- Рідкісні або нові монети, що мало зібрано у навчальному датасеті.

4. Ідеї для покращення системи

На основі аналізу результатів було сформульовано кілька напрямів для удосконалення роботи моделі:

- Розширення датасету:

Додавання більше зображень у класи, які розпізнаються гірше. Покращити фон, на котрій буде зніматися монета, освітлення та кут зйомки.

- Технічні покращення моделі:

Застосування аугментації дозволить розширити навчального набору даних за допомогою модифікацій зображень. Обертання, масштабування, дзеркальне відображення, зміни яскравості або контрасту - ці зміни не змінюють клас зображення, але ускладнюють задачу моделі та змушують її на ширшому діапазоні варіантів.

Висновки за розділом 3

У третьому розділі було реалізовано систему розпізнавання монет, яка включає як клієнтську, так і серверну частини, також модуль нейронної мережі, яка побудована за допомогою бібліотек TensorFlow та Keras. Клієнтська частина забезпечує зручний веб-інтерфейс, за допомогою якого користувач має можливість завантажити зображення монети та отримати результат розпізнавання. Серверна частина відповідає за обробку зображення, передбачення класу задіючи навченої моделі та формування відповіді.

Реалізований модуль розпізнавання показав високу ефективність у роботі з різними типами зображень. Система досягла точності понад 92% на валідаційній вибірці, після навчання на основі датасету, котрий складається з 7650 зображень, що охоплюють 18 класів. Модель здатна впевнено розпізнавати більшість монет, за умовами відмінностей у якості та умовах зйомки.

Під час аналізу результатів було виявлено, що деякі монети система розпізнає з високою точністю. Водночас певні класи монет, що мають подібний дизайн або зустрічаються рідше, розпізнаються з меншою точністю. Було

виявлено основні причини помилок - нечіткість зображень, відблиски, затемнення та нестача зображень у датасеті.

ВИСНОВКИ

У даній кваліфікаційній роботі було виконано створення моделі технічного зору для автоматичного розпізнавання монет на основі нейронної мережі. Проведено теоретичний аналіз принципів роботи комп'ютерного зору, розглянути сучасні методи глибокого навчання та інструменти для їх реалізації. Створено власний набір даних , що включає 7650 зображень монет, які класифіковано за роками випуску.

Реалізацію системи здійснено за архітектурою “клієнт-сервер”, де клієнтська частина забезпечує взаємодію з користувачем через веб-інтерфейс, а серверна - обробку зображень і виконання класифікації за допомогою згорткової нейронної мережі. Проведено навчання моделі , її тестування та оцінювання якості роботи. Результати свідчать про високу точність розпізнавання понад 92% , що є хорошим показником для задачі класифікації.

У роботі проведено аналіз помилок, виявлено фактори, що впливають на зниження точності, та запропоновано конкретні шляхи вдосконалення системи. Зокрема , покращення якості зображень, розширення навчального набору та застосування сучасніших архітектур нейромереж.

Таким чином , поставлені в роботі цілі були досягнуті повністю, а запропоноване рішення підтвердило свою ефективність як у технічному, так і в прикладному аспектах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. КРОМКАЧ В. РОЛЬ КОМП'ЮТЕРНОГО ЗОРУ В СУЧАСНОМУ СВІТІ: ДОСЯГНЕННЯ, ВИКЛИКИ ТА ПЕРСПЕКТИВИ. *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2024. № 2. С. 79–87.
2. ЛЕКЦІЯ № 8 ОСНОВИ КОМП'ЮТЕРНОГО ЗОРУ.
URL: https://learn.ztu.edu.ua/pluginfile.php/294778/mod_resource/content/1/ОШІ_Л-8_КЗір.pdf (дата звернення: 15.10.2022).
3. Numista. *Numista*. URL: <https://en.numista.com/> (дата звернення: 10.10.2024).
4. Maktun.com. *Maktun.com*. URL: <https://maktun.com/ru/> (дата звернення: 10.10.2025).
5. CoinSnap: Coin Identifier - Learn Coins and Their Value. *Coin*. URL: <https://coinidentifierai.com/> (дата звернення: 10.10.2025).
6. Google Об'єктив. *Google Lens*. URL: <https://lens.google/intl/uk/> (дата звернення: 10.12.2024).
7. Coinoscope - Identify coins by image. *Coinoscope - Identify coins by image*. URL: <https://coinscope.com/> (дата звернення: 10.10.2025).
8. Програмування числових методів мовою Python / А. В. Анісімов та ін. Київ : Видавничо-полігр. центр "Київ. ун-т", 2015. 640 с.
9. Мокін В. Б., Дратований М. В. Наука про дані: машинне навчання та інтелектуальний аналіз даних. Вінниця : ВНТУ, 2024. 264 с.
10. ВСТУП ДО WEB-ПРОГРАМУВАННЯ. ОСНОВИ HTML / А. Ю. Брила та ін. Ужгород : Ужгор. нац. ун-т, 2018. 72 с.
11. Каплун В. А., Ціхоцький М. С., Лукічов В. В. Основи web-програмування. Теорія і практика. Вінниця : ВНТУ, 2023. 129 с.

12. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2nd Edition. Packt Publishing, 2017. 622 c.
13. Approaching Any Machine Learning Problem. Abhishek Thakur, 2020. 300 c.
14. Deep Learning with Python. Manning Publications, 2017. 384 c.

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 15 – Автоматизація та приладобудування
Спеціальність: 151 – Автоматизація та комп'ютерно-інтегровані технології
Освітня програма «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри комп'ютерних
систем та робототехніки
к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року

З А В Д А Н Н Я НА КВАЛІФІКАЦІЙНУ РОБОТУ

СТАРКОВ Єгор Володимирович
(прізвище, ім'я, по батькові студента)

1. Тема роботи «Модель технічного зору для розпізнавання монет»

Керівник: кандидат технічних наук, доцент Бикова Тетяна Володимирівна.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від 16.04. 2025 року № 4101-5/962

2. Строк подання студентом роботи 30 травня 2025 року

3. Перелік питань, які потрібно розробити:

- 1)Визначення технічних параметрів системи технічного зору для розпізнавання монет.
- 2)Аналіз ефективності різних методів розпізнавання монет.
- 3)Розробка алгоритму обробки зображень для ідентифікації монет.
- 4)Проектування інтерфейсу взаємодії користувача із системою розпізнавання монет.
- 5)Навчання нейромережі для розпізнавання монет.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми кваліфікаційної роботи	02.10.2024-16.10.2024
2	Перегляд научної літератури	20.10.2024-10.11.2024
3	Визначення технічних параметрів системи технічного зору для розпізнавання монет	13.11.2024-15.12.2024
4	Аналіз ефективності різних методів розпізнавання монет.	07.01.2025-28.01.2025
5	Розробка алгоритму обробки зображень для ідентифікації монет.	01.02.2025-28.02.2025
6	Проектування інтерфейсу взаємодії користувача із системою розпізнавання монет.	03.03.2025-31.03.2025
7	Навчання нейромережі для розпізнавання монет.	01.04.2025-10.04.2025
8	Підготовка технічного звіту	11.04.2025-14.05.2025
9	Розробка пояснювальної записки.	15.05.2025-29.05.2025
10	Представлення кваліфікаційної роботи керівнику та рецензенту.	30.05.2025
11	Оформлення пояснювальної записки та підготовка презентації.	30.05.2025

5. Дата видачі завдання **02 жовтня 2024 року.**

Студент

Старков Є.В

ініціали, прізвище

підпис

Керівник роботи

Бикова Т.В

ініціали, прізвище

підпис

Затверджую

« _____ » _____ 2024 р.

**Технічне завдання
на розробку моделі**

«Модель технічного зору для розпізнавання монет».

1.	Введення	1.1 Назва роботи – «Модель технічного зору для автоматичного розпізнавання монет» 1.2. Галузь застосування: Автоматизація та приладобудування : нумізматики, банківські установи.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю 151 – Автоматизація та комп'ютерно-інтегровані технології 2.2. Завдання на кваліфікаційну роботу бакалавра № 4101-5/962 від «16 » квітня 2025 року (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3.	Призначення розробки	3.1. Мета розробки: Створення розробка моделі технічного зору для автоматизованого розпізнавання монет на основі нейронної мережі. 3.2. Призначення розробки: Модель призначена для автоматизованого розпізнавання та класифікації монет. 3.3. Вхідні дані: вхідними даними для моделі являються фотографії, котрі користувачі завантажуватимуть через пристрої. 3.4. Вихідні дані розробки: Інформація про монету: "Рік випуску", "Вартість монети", "Назва монети".

4.	Технічні вимоги до програмного виробу	4.1. Функціональні вимоги: -Завантаження зображень монет користувачем; -Автоматичне визначення номіналу, року випуску та назви монети; -Надання зручного інтерфейсу користувачу; -Можливість додавання зображень в датасети, без перебудови структури. 4.2 Нефункціональні вимоги: -Висока точність розпізнавання, більше 90%; -Швидке надання результату користувачеві; -Зручне використання додатку з різних пристроїв; 4.3. Вимоги до інтеграції: - Модель має бути інтегрована з інтерфейсом для завантаження зображень та відображення результатів розпізнавання. - Зображення повинні звірятися з базою даних монет для подання інформації про вартість, рік випуску, країну походження.
5.	Вимоги до програмної документації	Документацією до виробу «Модель технічного зору для автоматизованого розпізнавання монет» вважати: 1) Опис проєкту: Документ, що містить загальну характеристику виробу, його призначення, основні можливості та обмеження. 2) Керівництво користувача: Інструкція для кінцевих користувачів щодо завантаження зображень, використання інтерфейсу та результатів розпізнавання. 3) Тестування: Опис, що містить тестування, результати тестів функціональності . 4) Відомості про використання ресурси : Перелік бібліотек, технологій та датасету, що були застосовані під час розробки вибору.

6.	Вимоги до техніко-економічних показників	Документацією до виробу «Модель технічного зору для автоматичного розпізнавання монет» вважати: 1) Швидкість обробки зображень: Модель повинна швидко забезпечувати розпізнавання монет. 2) Точність розпізнавання: Модель повинна мати високий рівень розпізнавання для видів монет, при мінімальних втрат.	
7.	Стадії і етапи розробки	Дата	Назва етапу
		02.10.2024-16.10.2024	Затвердження теми кваліфікаційної роботи
		25.10.2024-10.11.2024	Перегляд научної літератури . Формування вимог до технічних характеристик.
		13.11.2024-15.12.2024	Визначення технічних параметрів системи технічного зору для розпізнавання монет.
		07.01.2025-28.01.2025	Аналіз ефективності різних методів розпізнавання монет.
		01.02.2025-28.02.2025	Розробка алгоритму обробки зображень для ідентифікації монет.
		03.03.2025-31.03.2025	Проектування інтерфейсу взаємодії користувача із системою розпізнавання монет.
		01.04.2025-10.04.2025	Навчання нейромережі для розпізнавання монет.
		11.04.2025-14.05.2025	Підготовка технічного звіту
		15.05.2025-29.05.2025	Розробка пояснювальної записки.
		30.05.2025	Представлення кваліфікаційної роботи керівнику та рецензенту.

		30.05.2025	Оформлення пояснювальної записки та підготовка презентації.
8.	Порядок контролю приймання моделі	і	1. Захист розробленої моделі провести на засіданні Атестаційної комісії. 2. Пояснювальну записку необхідно подати на паперовому носії в одному примірнику.

Виконавець

Студент групи КУ-41

Старков Є.В



Замовник

канд. техн. наук, доц.

Бикова Т.В



Додаток В

Програма і методика випробувань програмного виробу
«Модель технічного зору для розпізнавання монет»

1. Об'єкт випробувань

1. Назва програмного виробу : «Модель технічного зору для розпізнавання монет»
2. Галузь застосування: Автоматизація та приладобудування : нумізматики, банківські установи.

2. Мета випробувань

Метою кваліфікаційної роботи є розробка моделі технічного зору для автоматизованого розпізнавання монет на основі нейронної мережі.

3. Загальні положення**1. Підстави для проведення випробувань**

Підставою для проведення випробувань є:

- 1) Наказ ХНУ № 4101-5/962 від 16.04.2025 про затвердження тем кваліфікаційних робіт бакалаврів;

- 2) Навчальний план дисципліни.

2. Місце і тривалість випробувань

Випробування моделі технічного зору для розпізнавання монет проводилися онлайн з використанням персонального комп'ютера. Тривалість випробувань відбулося з 25.05.2025 по 30.05.2025.

3. Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

4. Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться за участю Керівника , Виконавця та членів атестаційної комісії.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

1. Використання персонального комп'ютера або мобільного пристрою з камерою для зчитування зображень монет.
2. Вимоги до надійності:
Система повинна стабільно працювати в умовах різних якостей зображень.
3. Захист від некоректних дій користувачів :

Зроблено обробку помилок у разі завантаження нечітких або неформатованих зображень.

4. Забезпечення стабільної роботи системи при великому навантаженні:

Алгоритм повинен ефективно обробляти великі обсяги зображень без зниження продуктивності.

5. Вимоги до програмної документації

Програмою документацією щодо розроблюваного програмного продукту вважати:

1. Технічне завдання на розробку програми

- Визначення технічних параметрів системи технічного зору для розпізнавання монет.
- Аналіз ефективності різних методів розпізнавання монет.
- Розробка алгоритму обробки зображень для ідентифікації монет.
- Проектування інтерфейсу взаємодії користувача із системою розпізнавання монет.
- Навчання нейромережі для розпізнавання монет.

2. Програма та методика випробувань розробленої системи:

1) Методика випробувань:

- Підготовка тестової вибірки:
Сформувані набір зображень монет різного року.
- Запуск системи з датасетом
Завантажити до системи тестові зображення та запустити розпізнавання.
- Фіксація результатів:
Для зображення зафіксувати результат класифікації (рік, номінал, вартість) та порівняти його з еталонним значенням.
- Аналіз точності:
Визначити відсоток правильно класифікованих монет. Точність повинна бути не менше 90%.

2) Програма випробувань:

- Перевірка працездатності інтерфейсу користувача:
Перевірити, як інтерфейс реагує на дії користувача, копки працюють, помилки не виникають.
- Тестування обробки вхідного зображення:
Зображення монети правильно зчитується, відображається у вікні попереднього перегляду.
- Класифікація монети на різних типах монет:
Модель повертає правильний номінал, рік, вартість; результати відповідають з очікуванням.
- Перевірка системи при запуску:
Проаналізувати, чи стабільно працює система при послідовному розпізнаванні кількох зображень підряд.

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань необхідний персональний комп'ютер або пристрій з камерою. Також набір зображень для тесту.

6.2 Порядок проведення випробувань

1-й етап - Ознайомчий

Перелік перевірок:

1. Перевірка комплексної пояснювальної записки та наявності програмної документації.
2. Перевірка відповідності змісту документації технічному завданню.
3. Аналіз комплексності технічних і програмних засобів, необхідних для запуску моделі.
4. Перевірка якості оформлення документації на відповідність ДСТУ.
5. Ознайомлення з інструкцією користувача.

2-й етап - Основне випробування програмного виробу

Перелік перевірок:

1. Перевірка відповідності технічних характеристик вимогам технічного завдання.
2. Оцінка функціональності розпізнавання: правильність визначення року, номіналу та вартості монети.
3. Оцінка швидкості обробки зображення.

Порядок проведення випробувань:

1. Підготовка зображень монет
2. Визначення відсоток точності розпізнавання зображень.
3. Перевірка системи на зображення іншим номіналом.

Приклади тестів:

Тест 1

1. Завантажується зображення монети, що входить до складу тренувального датасету.
2. Модель видає правильні характеристики.
3. Результат вважається позитивним, якщо всі три ознаки співпадають.



Рисунок В.1 - Тест 1.

Результат:
Номінал: 50 копійок
Рік: 2003
Вартість: 2000-3000 гривень

Рисунок В.2 - Тест 1.

Тест 2

1. Завантажується зображення монети, що не входить до складу тренувального датасету.
2. Видача помилки.



Рисунок В.3 - Тест 2.

Помилка! Монету не вдалося розпізнати.

Рисунок В.4 - Тест 2.

Тест вважається успішним, якщо зазначені дії виконуються коректно та результати відображаються у програмному інтерфейсі відповідно до очікувань.

Висновки: Тест 1 та тест 2 були успішно виконані, це всвідчить про позитивні результати випробувань.

Виконавець: студент групи КУ-41, Старков Є.В

