

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____Олександр ЛИТВИНЕНКО

«___» _____2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль ідентифікації військової техніки на знімках

аеророзвідки

Виконавець: _____ Анна ПЕЛЕХ

Керівник: _____ Наталія АПЕНЬКО

Нормоконтролер: _____ Євгеній ТУПОТА

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр ЛИТВИНЕНКО

«_____» _____ 2023 р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

_____ Пелех Анни Василівни

1. Тема роботи: _____ «Програмний модуль ідентифікації військової техніки на знімках аеророзвідки»

затверджена наказом ректора від «28» серпня 2023 року № 1494 /ст.

2. Термін виконання роботи: з 02.10.2023 до 31.12.2023

3. Вихідні дані до проекту (роботи): постановка задачі до виконання роботи, мова програмування *Python*.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):

1) принципи застосування нейронних мереж для розпізнавання військової техніки на знімках аеророзвідки;

2) моделювання роботи програмного модуля розпізнавання військової техніки на зображеннях;

3) програмне впровадження модуля розпізнавання військової техніки на знімках аеророзвідки.

5. Перелік обов'язкового графічного матеріалу:

1) дерево функцій;

2) представлення роботи програмного модуля у форматі *IDEF0*;

3) представлення роботи програмного модуля у форматі *IDEF0* рівня *A1*;

4) діаграма використання;

5) зв'язки між таблицями БД програмного модуля;

6) інтерфейс програмного модуля;

7) схема алгоритму попередньої обробки і тренування моделі.

6. Календарний план-графік

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1	Провести аналіз літератури за темою кваліфікаційної роботи та аналіз існуючих систем	02.10.2023-11.10.2023	
2	Вибір та формування напряму дослідження	12.10.2023-28.10.2023	
3	Аналіз предметної області	29.10.2023-05.11.2023	
4	Вибір методів рішення задачі	06.10.2023-12.11.2023	
5	Створення програмного модулю	13.11.2023-24.11.2023	
6	Проведення налаштування програмних засобів на сервері	25.11.2023-02.12.2023	
7	Написання пояснювальної записки	03.12.2023-13.12.2023	
8	Підготовка презентації	14.12.2023-18.12.2023	
9	Оформлення супроводжувальної документації	19.12.2023-20.12.2023	
10	Підготовка до захисту	25.12.2023-31.12.2023	

7. Дата видачі завдання «02» жовтня 2023 р.

Керівник кваліфікаційної роботи _____ Наталія АПЕНЬКО
(підпис)

Завдання прийняв до виконання _____ Анна ПЕЛЕХ
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Програмний модуль ідентифікації військової техніки на знімках аеророзвідки» складається з: 85 с., 32 рисунки, 30 літературних джерел, 1 додаток.

РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, ВІЙСЬКОВА ТЕХНІКА, НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ.

Метою роботи є створення програмного модулю ідентифікації військової техніки на знімках аерозвідки.

Об'єкт дослідження – процес розпізнавання військової техніки на знімках аеророзвідки.

Предмет дослідження – програмний модуль ідентифікації військової техніки на знімках аеророзвідки.

Апробація результатів була проведена на міжнародній науковій конференції “Інтелектуальні технології лінгвістичного аналізу” (м. Київ, 23-24 жовтня 2023 р.), науковій практичній конференції “Сучасні тенденції розвитку системного програмування” (м. Київ, 23-24 листопада 2023 р.).

ЗМІСТ

<u>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ</u>	7
<u>ВСТУП</u>	8
<u>РОЗДІЛ 1 АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕХНІКИ НА ЗНІМКАХ</u>	11
<u>1.1. Аналіз методів обробки і розпізнавання знімків аерозвідки для ідентифікації військової техніки</u>	11
<u>1.2. Аналіз наукових досліджень за напрямком розпізнавання військової техніки на зображеннях</u>	20
<u>1.3. Стандартні алгоритми розпізнавання зображень</u>	22
<u>1.4. Аналіз вимог до програмного модуля і постановка задачі на розробку</u>	23
<u>1.5. Висновки до розділу</u>	25
<u>РОЗДІЛ 2 МОДЕЛЮВАННЯ РОБОТИ ПРОГРАМНОГО МОДУЛЯ ІДЕНТИФІКАЦІЇ ВІЙСЬКОВОЇ ТЕХНІКИ НА ЗОБРАЖЕННЯХ</u>	26
<u>2.1. Аналіз основних процесів предметного середовища</u>	26
<u>2.2. Проектування інформаційного забезпечення програмного модуля інтелектуального розпізнавання військової техніки</u>	35
<u>2.3. Проектування інтерфейсу програмного модуля інтелектуального розпізнавання військової техніки</u>	48
<u>2.4. Висновки до розділу</u>	52
<u>РОЗДІЛ 3 РЕАЛІЗАЦІЯ, ТЕСТОВІ ПРИКЛАДИ</u>	53
<u>3.1. Вибір програмного інструментарію для розробки програмного модуля</u>	53
<u>3.2. Розроблені алгоритми для підтримки роботи модуля</u>	54
<u>3.3. Навчання нейронної мережі</u>	60
<u>3.4. Описання інтерфейсу програмного модуля інтелектуального розпізнавання військової техніки за знімками аерозвідки</u>	74
<u>3.5. Висновки до розділу</u>	79
<u>ВИСНОВКИ</u>	80
<u>СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ</u>	83
<u>ДОДАТОК А</u>	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

ANN – *artificial neural networks* (штучна нейронна мережа)

CNN – *Convolutional Neural Network* (конволюційна нейронна мережа)

MCC – *Matthews Correlation Coefficient* (коефіцієнт кореляції Метьюза)

F2 – *F-beta score* зі значенням $\beta = 2$

AUC – *Area Under the Curve* (площа під кривою)

PR – *Precision-Recall* (точність-повернення)

DS – *dataset* (набір даних)

ReLU – *Rectified Linear Unit*

MaxPooling2D – шар пулінгу (зменшення розміру зображення)

ГНМ – глибинна нейронна мережа

ШНМ – штучна нейронна мережа

ВСТУП

Актуальність теми. У сучасному світі інформація є однією з найцінніших сировин, і величезний обсяг текстової інформації генерується кожною секундою завдяки активному використанню соціальних медіа, інтернет-ресурсів, електронних комунікацій тощо. Важливою складовою цієї текстової інформації є її тональність, яка визначає, чи є текст позитивним, негативним або нейтральним у відношенні до певного об'єкту чи події. Визначення тональності тексту має велике практичне значення, оскільки воно дозволяє автоматизувати аналіз великого обсягу текстової інформації та використовувати її для прийняття рішень в різних галузях, включаючи маркетинг, рекламу, фінанси, суспільно-політичний аналіз та інші.

Актуальність проблеми визначення тональності тексту обумовлена необхідністю вдосконалення методів та інструментів аналізу текстової інформації в умовах інформаційного перевантаження. Попит на автоматизовані системи визначення тональності зростає з кожним днем, оскільки бізнес, владні структури, аналітичні агентства та соціальні медіа активно використовують аналіз тональності для прийняття рішень, відстеження громадської думки та прогнозування ринкових тенденцій.

Предметом дослідження даної магістерської роботи є методи та алгоритми для визначення тональності українських текстів. Основним об'єктом дослідження є вплив різних методів попередньої обробки тексту, вибору моделей та методів машинного навчання на точність визначення тональності. Магістерська робота спрямована на розвиток ефективних алгоритмів, які можна використовувати для визначення тональності українських текстів у різних сферах застосування.

Об'єктом дослідження є текстова інформація українською мовою, а предметом дослідження є методи та алгоритми визначення тональності цієї інформації.

Метою даної роботи є розробка та апробація ефективних методів визначення тональності українських текстів за допомогою машинного навчання та аналізу природної мови. Крім того, метою є порівняння різних методів та алгоритмів для визначення тональності та розробка рекомендацій щодо їх використання в практичних застосуваннях.

Для досягнення поставленої мети перед магістерською роботою стоїть низка основних завдань:

Аналіз існуючих методів визначення тональності тексту. Провести огляд і аналіз існуючих методів та підходів до визначення тональності українських текстів, включаючи техніки машинного навчання та аналізу природної мови.

Розробка методів попередньої обробки тексту. Розробити та дослідити методи попередньої обробки текстових даних, такі як токенізація, векторизація, видалення шуму, які впливають на точність визначення тональності.

Реалізація та навчання моделей машинного навчання. Розробити та навчити моделі машинного навчання для визначення тональності текстів на основі вибраних методів та алгоритмів.

Оцінка та порівняння результатів. Порівняти результати різних методів та алгоритмів для визначення тональності та визначити їх ефективність у різних сферах застосування.

Розробка рекомендацій. Розробити рекомендації щодо використання найбільш ефективних методів та алгоритмів для визначення тональності текстів українською мовою в практичних застосуваннях.

Визначення тональності тексту українською мовою є важливим завданням в сучасному інформаційному середовищі. Дана магістерська робота спрямована на дослідження та розвиток ефективних методів визначення тональності, що можуть бути використані в різних галузях. Вона передбачає аналіз існуючих методів, розробку нових підходів, навчання моделей та оцінку їх ефективності. Результати цього дослідження можуть мати практичне застосування та сприяти автоматизації аналізу текстової інформації.

Апробація результатів була проведена на міжнародній науковій

конференції “Інтелектуальні технології лінгвістичного аналізу” (м. Київ, 24-25 жовтня 2023 р.), науковій практичній конференції “Сучасні тенденції розвитку системного програмування” (м. Київ, 23-24 листопада 2023 р.).

За результатами наукових конференцій опубліковано тези доповідей:

1. Пелех А.В. Принципи визначення військової техніки на знімках аеророзвідки / Пелех А.В., Дехтяренко Д.Т., Рябчук Н.А. // Тези доповідей міжн. наук.-техн. конф. “Інтелектуальні технології лінгвістичного аналізу” (м. Київ, 24-25 жовтня 2023 р.) – К.: НАУ, 2023. – С. 68.

2. Пелех А.В. Програмний модуль визначення військової техніки на знімках аеророзвідки / Апенько Н.В., Пелех А.В.// Тези доповідей наук.-практ. конф. “Сучасні тенденції розвитку системного програмування” (23-24 листопада 2023 р.). – К.: НАУ, 2023. – С. 48-50.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ РОЗПІЗНАВАННЯ ТЕХНІКИ НА ЗНІМКАХ

1.1. Аналіз методів обробки і розпізнавання знімків аерозвідки для ідентифікації військової техніки

Розпізнавання об'єктів на аерозвідкових знімках - це складна та багатоаспектна задача, яка вимагає об'єднання декількох сфер знань, таких як комп'ютерний бачення, обробка зображень, машинне навчання, геоінформаційні технології та військова техніка. Перед початком розробки системи розпізнавання об'єктів на аерозвідкових знімках, необхідно ретельно вивчити основні аспекти цієї задачі та визначити специфічні вимоги для конкретного застосування.

Основні етапи задачі розпізнавання об'єктів:

1. Збір та підготовка даних: Перший етап передбачає отримання супутникових чи аерозйомочних знімків, які містять потенційні об'єкти для розпізнавання. Важливо підготувати дані, виконати їх реєстрацію, усунення шуму та зниження роздільної здатності для полегшення подальшого аналізу.

2. Вибір області інтересу: На знімках можуть міститися десятки або навіть сотні квадратних кілометрів земної поверхні. Вибір області інтересу (*ROI*) - це важливий крок, який дозволяє зменшити обсяг обробки даних та спрямувати увагу на конкретні об'єкти.

3. Сегментація зображення: На цьому етапі використовуються методи сегментації, щоб виділити об'єкти із зображення та визначити їх межі. Це може включати в себе бінарну сегментацію, сегментацію за кольором, за текстурою тощо.

4. Витягнення ознак: Після сегментації зображення витягуються ознаки (фізичні, структурні, текстурні тощо), які описують кожен об'єкт на знімку. Це можуть бути характеристики кольору, форми, розміру, орієнтації тощо.

5. Вибір та навчання моделі: Вибір моделі машинного навчання для розпізнавання об'єктів - це важливий крок. Зазвичай використовуються глибокі нейронні мережі, зокрема згорткові нейронні мережі (*CNN*), які дозволяють виявляти складні патерни в зображеннях.

6. Класифікація та визначення об'єктів: Навчана модель використовується для класифікації об'єктів на знімках. Це означає, що кожен об'єкт ідентифікується та класифікується за певною категорією, наприклад, танк, бронетранспортер, літак тощо.

7. Аналіз результатів та прийняття рішень: Результати розпізнавання об'єктів аналізуються та використовуються для прийняття рішень, наприклад, для розвідки, виявлення цілей або моніторингу об'єктів.

Основні виклики та проблеми:

– Складність об'єктів: Об'єкти на знімках аерозвідки можуть мати складну структуру та різні масштаби, що ускладнює їх розпізнавання.

– Зміни освітлення та погодні умови: Зміни освітлення та погодні умови можуть впливати на якість та зміст знімків, що робить завдання розпізнавання об'єктів більш складним.

– Великий обсяг даних: Сучасні системи аерозвідки генерують великі обсяги даних, що вимагають потужних обчислювальних ресурсів для їх обробки та аналізу.

– Варіативність об'єктів: Об'єкти можуть варіюватися в залежності від моделей, типів, розмірів та стану. Ця варіативність ускладнює розпізнавання.

Розпізнавання об'єктів на знімках аерозвідки є важливою складовою сучасних систем обробки зображень та розвідки. Цей процес полягає в автоматичному визначенні і класифікації об'єктів, зображених на аерофотознімках або супутникових знімках. Завдяки швидкому розвитку обчислювальних технологій та штучного інтелекту, задачі розпізнавання об'єктів стали більш точними і широко застосовуються у різних сферах, включаючи військовий розвідку, геологічне картографування, аграрну сферу, екологію та багато інших.

Основні завдання розпізнавання об'єктів:

1. Сегментація областей інтересу (*Region of Interest - ROI*): Один з перших кроків у задачі розпізнавання об'єктів - виділення областей зображення, які містять потенційно цікаві об'єкти. Це може включати в себе виокремлення окремих об'єктів або областей з зображення, які містять об'єкти певного типу, наприклад, ліси, дороги, річки або будівлі.

2. Витягнення ознак (*Feature Extraction*): Ознаки, які описують об'єкти на зображенні, допомагають ідентифікувати їх. Це можуть бути текстурні ознаки, колір, форма, розмір та інші характеристики. Деякі об'єкти можуть бути ідентифіковані за допомогою унікальних ознак, наприклад, радарних або інфрачервоних показників.

3. Класифікація (*Classification*): Після витягнення ознак об'єктів необхідно класифікувати їх на певні категорії або класи. Наприклад, на зображенні можуть бути класифіковані літаки, автомобілі, кораблі, будівлі тощо.

4. Визначення стану (*State Estimation*): Крім класифікації, розпізнавання об'єктів може також включати визначення стану цих об'єктів. Наприклад, визначення, чи літак приземлений або в повітрі, може бути важливим у військових додатках.

5. Відстеження (*Object Tracking*): У випадку, якщо об'єкти на знімках рухаються, важливо відстежувати їх рух та прогнозувати майбутнє положення. Це може бути корисним у військовій розвідці або в метеорологічних додатках для відстеження стихійних лих.

6. Застосування військової техніки (*Military Applications*): В армії задачі розпізнавання об'єктів є критичними. Це включає в себе виявлення ворожих танків, літаків або підрозділів, а також ідентифікацію своїх об'єктів.

7. Картографування та дослідження навколишнього середовища (*Environmental Mapping and Exploration*): У геологічному дослідженні та дослідженні навколишнього середовища також використовуються методи розпізнавання об'єктів для картографування та визначення особливостей природи.

8. Цивільні застосування (*Civilian Applications*): Завдання розпізнавання об'єктів також мають цивільні застосування, включаючи транспортне моделювання, геологічне дослідження, моніторинг вуглеводнів, метеорологічні додатки, аграрну сферу та інші.

Технології та методи розпізнавання об'єктів:

1. Машинне навчання (*Machine Learning*): Машинне навчання, зокрема глибокі нейронні мережі, стали основним інструментом для задач розпізнавання об'єктів. Вони можуть автоматично вчитися на прикладах та визначати складні закономірності в даних.

2. Комп'ютерний зір (*Computer Vision*): Методи комп'ютерного зору дозволяють комп'ютерам аналізувати та інтерпретувати зображення, що відкриває широкий спектр можливостей для розпізнавання об'єктів.

3. Аналіз природної мови (*Natural Language Processing - NLP*): У випадках, коли об'єкти розпізнаються на основі текстових даних (наприклад, на знімках з дронів), аналіз природної мови використовується для обробки цих даних.

4. Обробка сигналів та радары: У військових застосуваннях радары та системи обробки сигналів важливі для виявлення та відстеження об'єктів в реальному часі.

5. Геоінформаційні системи (*GIS*): Геоінформаційні системи використовуються для обробки та аналізу геопросторових даних, включаючи знімки аерозвідки.

Підготовка даних - це один з найважливіших етапів у процесі обробки і аналізу інформації. У випадку розпізнавання об'єктів на супутникових знімках для класифікації військової техніки, правильна підготовка даних визначає якість та ефективність подальшого аналізу. В даному розділі будуть розглянуті основні завдання та виклики, які виникають на етапі підготовки даних для розпізнавання об'єктів на супутникових знімках.

Перший крок у підготовці даних - це збір необхідної інформації та вибір джерел даних. Для задачі розпізнавання військової техніки на супутникових

знімках, джерелами можуть бути різні супутникові служби, державні агентства аеророзвідки, комерційні постачальники знімків тощо. Важливо визначити, які джерела мають найкращу якість та доступність для конкретної задачі.

Дані, отримані з різних джерел, часто потребують обробки та попереднього аналізу, перш ніж їх можна буде використовувати для навчання моделей розпізнавання. Цей процес може включати в себе видалення шумів і артефактів, вирівнювання знімків за параметрами, що забезпечують порівнянність, та визначення регіонів інтересу, де розміщена військова техніка.

Однією з важливих задач підготовки даних є маркування та анотування об'єктів і явищ на супутникових знімках. Це може бути вручну або за допомогою автоматизованих систем. Анотовані дані є ключовим ресурсом для тренування і валідації моделей розпізнавання.

Для ефективного навчання та оцінки моделей розпізнавання важливо розбити дані на три основних набори: навчальний, валідаційний і тестовий. Навчальний набір використовується для навчання моделі, валідаційний - для налаштування гіперпараметрів та контролю перенавчання, а тестовий - для оцінки точності та генералізації моделі на нових даних.

Для покращення роботи моделей розпізнавання, особливо при обмежених обсягах даних, важливо розробити методи аугментації даних. Це може включати в себе збільшення об'єктивності даних шляхом створення штучних зразків або зміну освітлення, кута зйому, масштабу тощо.

Часто дані можуть бути несбалансованими, коли одних класів об'єктів на знімках набагато менше, ніж інших. Це може призвести до перекосу моделі у бік більш представлених класів. Вирішення цієї проблеми включає в себе методи балансування даних, такі як *oversampling* або *undersampling*.

Завершальним етапом підготовки даних є їх перетворення та упаковка у формат, який може бути використаний у моделях розпізнавання. Це може включати в себе кодування зображень у числовий вигляд, стандартизацію даних, збереження анотацій у зручному форматі та інші операції.

Підготовка даних для розпізнавання об'єктів на супутникових знімках - це складний та багатоетапний процес, який вимагає врахування багатьох технічних та методологічних аспектів. Від якості підготовки даних залежить успішність підсистеми розпізнавання та точність результуючих аналітичних даних. Детальне розглядання і вирішення завдань на етапі підготовки даних є кроком вперед у досягненні мети розпізнавання військової техніки на супутникових знімках.

Сегментація зображень - це один із ключових етапів обробки зображень у великій кількості сучасних застосувань, включаючи сфери медицини, автономних автомобілів, аналізу супутникових знімків та багато інших. В контексті задачі розпізнавання військової техніки на супутникових знімках, сегментація зображень є важливим компонентом, який дозволяє визначити точне розташування та межі об'єктів інтересу на зображеннях. У цьому розділі ми розглянемо основні задачі, пов'язані з сегментацією зображень, та методи їх вирішення.

Однією з основних задач сегментації зображень є бінарна сегментація. Вона полягає у виділенні об'єктів інтересу на зображенні та відокремленні їх від фону. У випадку розпізнавання військової техніки, бінарна сегментація може бути використана для виділення техніки на супутникових знімках, роблячи всі інші пікселі чорними (фоном) і лише ті, які належать до об'єктів, білими (військовою технікою). Основні методи бінарної сегментації включають в себе порогову обрізку (*thresholding*), методи морфологічної обробки, а також методи машинного навчання, такі як нейронні мережі.

Семантична сегментація є більш складною задачею, оскільки вона передбачає виділення об'єктів не тільки на рівні пікселів, а й призначення класу кожному об'єкту. У випадку розпізнавання військової техніки, це означає, що кожному виділеному об'єкту потрібно призначити відповідний клас, такий як "танк", "бронетранспортер" і так далі. Одним із методів семантичної сегментації є використання глибоких нейронних мереж, зокрема, сверточних нейронних мереж (*CNN*), які навчаються розпізнавати об'єкти на зображеннях та визначати їх класи.

Інстанс-сегментація є найбільш складною задачею серед сегментаційних завдань. Вона включає в себе виділення окремих об'єктів, що належать до одного класу, та призначення їм унікальних ідентифікаторів.

Етап витягнення ознак є одним із важливих кроків в обробці супутникових знімків для класифікації військової техніки. Ознаки - це характеристики, які витягуються з оброблених зображень і використовуються для подальшого аналізу та класифікації. У цьому розділі розглянемо основні задачі на етапі витягнення ознак та методи їх вирішення.

Геометричні ознаки включають в себе інформацію про форму, розмір і розташування об'єктів на супутникових знімках. Однією з основних задач є визначення геометричних параметрів військової техніки, таких як довжина, ширина, площа, орієнтація тощо. Для цього можуть використовуватися методи обробки зображень, такі як визначення контурів об'єктів, розрахунок боксів (обмежуючих прямокутників) навколо об'єктів та обчислення кутів нахилу.

Колірні ознаки відіграють важливу роль в розпізнаванні військової техніки на супутникових знімках. Оскільки різні види військової техніки можуть мати характеристичні кольори або камуфляж, витягнення і аналіз кольорової інформації можуть допомогти в ідентифікації цих об'єктів. До колірних ознак відносять визначення кольору об'єктів, виявлення кольорових плям та розпізнавання характерних кольорів для різних класів військової техніки.

Текстурні ознаки вказують на особливості текстури поверхні об'єктів на зображеннях. Оскільки супутникові знімки можуть мати обмежену роздільну здатність, важливо виділяти текстурні ознаки для розпізнавання військової техніки. Однією з завдань є виявлення текстурних шаблонів, які характеризують певні класи об'єктів, таких як корпуси танків або гусениці бронетранспортерів.

Залежність від часу означає аналіз динаміки об'єктів на супутникових знімках. Оскільки рух і зміна розташування військової техніки може бути важливими ознаками, задача полягає в виявленні та аналізі цих змін. Методи витягнення ознак залежності від часу можуть включати в себе виявлення слідів

руху, визначення шляхів та швидкостей об'єктів, а також аналіз змін у розмірах та формах протягом часу.

Деякі види військової техніки можуть мати ідентифікаційні текстові або символічні маркування, такі як номери, логотипи або інші індикативні знаки. Витягнення ознак з текстового контенту може допомогти в ідентифікації та класифікації об'єктів. Для цього можуть використовуватися методи розпізнавання тексту, оптичного розпізнавання символів (*OCR*) та аналізу синтаксису.

Контекстні ознаки враховують взаємозв'язки та зв'язки між об'єктами на зображенні. Вони можуть допомогти в розумінні того, як об'єкти взаємодіють між собою та як це впливає на їхню класифікацію. Ознаки контексту можуть включати в себе аналіз відстаней між об'єктами, їхній розмір у порівнянні з іншими об'єктами та орієнтацію.

У багатьох випадках ефективність витягнення ознак залежить від комбінування різних типів ознак.

Наприклад, можна об'єднати геометричні та колірні ознаки, щоб покращити точність класифікації. Також можна враховувати залежності від часу та контексту, щоб отримати більш повний образ об'єкта.

Під час витягнення ознак важливо враховувати обмеження обчислювальних ресурсів, так як обробка супутникових знімків може бути обчислювально витратною. Оптимізація процесу витягнення ознак, використання алгоритмів зменшення розмірності та відбіру найінформативніших ознак можуть значно поліпшити продуктивність та точність системи розпізнавання військової техніки на супутникових знімках.

Етап класифікації є фінальним кроком в розпізнаванні військової техніки на супутникових знімках. На цьому етапі система приймає рішення про приналежність об'єктів до певних класів або категорій, таких як "танк", "бронетранспортер", "авіаційний апарат" і так далі. Для досягнення цієї мети важливо вирішувати ряд ключових задач.

Багатокласова класифікація означає, що об'єкти на зображенні можуть належати до одного чи більше класів. У контексті розпізнавання військової

техніки, це означає, що одне зображення може містити декілька видів військової техніки, і система повинна знати їх всі правильно.

Вирішення цієї задачі може базуватися на різних підходах, включаючи множинну бінарну класифікацію та використання спеціальних алгоритмів для багатокласової класифікації, таких як один проти всіх (*One-vs-All*) або покращені методи навчання з учителем.

Глибокі нейронні мережі, зокрема, згорткові нейронні мережі (*CNN*), є потужними інструментами для класифікації об'єктів на супутникових знімках. *CNN* можуть навчатися розпізнавати характеристичні ознаки об'єктів, такі як форма, текстура, кольори тощо. Їхня здатність розпізнавати об'єкти навіть у важких умовах освітлення, зміні масштабу та позиції робить їх часто використовуваними для розпізнавання військової техніки.

Однією з основних проблем у класифікації є перенавчання (*overfitting*), коли модель вивчає шум в даних та втрачає здатність узагальнення до нових зображень. Регуляризація, така як додавання термінів штрафу до функції втрати, може допомогти уникнути перенавчання та підвищити стабільність класифікаційних моделей.

У випадку нерівномірної розподіленості класів (наприклад, одні класи можуть бути рідше зустрічаються на зображеннях, ніж інші), важливо підтримувати баланс класів для підвищення точності класифікації. Це може включати в себе використання методів семпсування (*oversampling* або *undersampling*) та ваги класів під час навчання моделей.

Аналіз невизначеності є важливою частиною процесу класифікації, оскільки деякі класи об'єктів можуть бути складніше класифікувати, ніж інші. Визначення невизначеності допомагає системі приймати обґрунтовані рішення в умовах неоднозначності та зміни умов спостереження.

Під час класифікації важливо використовувати всю доступну інформацію, включаючи дані з інших сенсорів, джерел або контекстуальні дані. Наприклад, інформація з глобальних позиціонувальних систем (*GPS*), баз даних про військову техніку або дані від інших супутників може покращити точність класифікації.

Стабільність класифікації важлива у відомостях, де об'єкти можуть змінювати свою зовнішню зовнішність через час або наслідок дій ворожого фактору. Моніторинг стабільності класифікації і адаптація моделей до нових умов можуть підвищити надійність системи.

Оцінка результатів класифікації включає в себе визначення метрик ефективності, таких як точність, відновлення, F -мера, а також побудову матриць помилок. Ці метрики допомагають визначити якість роботи системи та виявити можливі напрями покращення.

Класифікація військової техніки на супутникових знімках може бути доповнена іншими завданнями, такими як визначення розташування, шляхів руху, оцінка потенційної загрози тощо. Використання зв'язаних завдань може допомогти в розумінні контексту та прийнятті більш обґрунтованих рішень.

1.2. Аналіз наукових досліджень за напрямком розпізнавання військової техніки на зображеннях

Аналіз наукових досліджень у галузі розпізнавання військової техніки на зображеннях відображає значний інтерес та активність у цьому напрямку. Цей розділ пропонує детальний огляд актуальних публікацій та досліджень у цій області.

Тематичний огляд досліджень:

1. Методи розпізнавання об'єктів:

– Сучасні дослідження відділяють методи розпізнавання об'єктів на зображеннях у дві головні категорії: класичні та глибокі нейронні мережі. Класичні підходи включають методи особливостей, геометричний аналіз та навчання з учителем. З іншого боку, глибокі нейронні мережі, зокрема згорткові нейронні мережі (*CNN*), стали популярними завдяки своїм вражаючим результатам у великих наборах даних. Вони дозволяють автоматично витягти корисні ознаки з зображень.

2. Навчання з учителем та без учителя:

– Одні з методів використовуються навчанням з учителем, де системі надаються позначені приклади для навчання та навчальні набори даних. Інші методи, такі як самонавчання або навчання без учителя, розвиваються для зменшення залежності від позначених даних та збільшення загальної придатності моделей.

3. Сегментація та детекція:

– Сегментація зображень та детекція об'єктів є ключовими складовими розпізнавання військової техніки. Дослідження в цій області охоплюють методи розмежування об'єктів, створення масок, а також точну локалізацію та класифікацію об'єктів на зображеннях.

4. Використання геопросторової інформації:

– Багато досліджень об'єднують геопросторову інформацію, таку як дані *GPS* або дані висоти, для покращення розпізнавання та геолокації військової техніки. Це особливо важливо для задач аеророзвідки.

5. Застосування аугментації даних:

– Багато досліджень визначають важливість аугментації даних. Застосування методів аугментації, таких як обрізання, повороти, зміна освітлення, може покращити загальну придатність моделей та запобігти перенавчанню.

6. Використання супутникових даних та реального часу:

– Деякі дослідження розглядають можливість використання супутникових знімків в реальному часі для відстеження та розпізнавання військової техніки в режимі реального часу.

7. Ефективність та обчислювальна складність:

– Оцінка ефективності моделей та їх обчислювальної складності є важливою складовою досліджень. Вдосконалення обчислювальної ефективності дозволяє застосовувати розпізнавання в реальному часі та на обмежених обчислювальних ресурсах.

Загальний висновок: Аналіз наукових досліджень у сфері розпізнавання військової техніки на зображеннях підтверджує розмаїття методів та підходів, які застосовуються для вирішення цієї складної задачі. Зростаючий інтерес до

застосування глибокого навчання та поєднання різних джерел інформації, таких як геопросторові дані, робить цю область дослідження актуальною та обіцяючою для розвитку сучасних систем розпізнавання військової техніки.

1.3. Стандартні алгоритми розпізнавання зображень

Огляд стандартних алгоритмів розпізнавання зображень, які застосовуються у завданнях класифікації військової техніки на супутникових знімках, показав, що вони включають в себе традиційні та деякі більш сучасні методи, які використовуються для аналізу зображень.

1. Метод головних компонент (*Principal Component Analysis - PCA*):

– *PCA* є одним з найпоширеніших методів зменшення розмірності даних та аналізу головних компонент. В контексті розпізнавання військової техніки, *PCA* може використовуватися для витягнення головних ознак із супутникових зображень та подальшої класифікації на основі цих ознак.

2. Метод опорних векторів (*Support Vector Machines - SVM*):

– *SVM* є потужним методом для бінарної та багатокласової класифікації. Вони працюють шляхом знаходження гіперплощини, яка найкращим чином розділяє класи об'єктів. В сфері розпізнавання військової техніки, *SVM* може бути застосований для класифікації об'єктів на супутникових знімках.

3. Метод каскадних класифікаторів (*Cascade Classifiers*):

– Каскадні класифікатори, такі як *Haar Cascade*, є швидкими та ефективними методами для об'єктного виявлення на зображеннях. Вони базуються на використанні ансамблю класифікаторів для послідовного виявлення об'єкта на зображенні. В розвідці та аеророзвідці, ці методи можуть використовуватися для знаходження військової техніки на супутникових знімках.

4. Метод згорткових нейронних мереж (*Convolutional Neural Networks - CNN*):

– *CNN* є потужним та важливим інструментом для розпізнавання зображень, особливо коли мається справа з великими об'ємами даних. Вони здатні

автоматично витягати ознаки зі зображень та використовувати їх для класифікації. В розпізнаванні військової техніки, *CNN* може надавати високу точність завдяки глибокому навчанню та згортковим шарам.

5. Метод глибокого навчання та нейронні мережі:

– Останнім часом нейронні мережі, зокрема глибокі нейронні мережі, здобувають популярність в задачах розпізнавання військової техніки на супутникових знімках. Вони здатні вчитися складним залежностям та автоматично витягати ознаки з зображень, що покращує точність класифікації.

Переваги:

– Ефективність: Деякі стандартні алгоритми, такі як *PCA* і *SVM*, можуть бути дуже ефективними у відсіюванні зайвої інформації та виявленні патернів на зображеннях.

– Відомість: Ці алгоритми вже добре досліджені і використовуються в багатьох галузях, включаючи медицину та обробку зображень.

Обмеження:

– Обмеженість у роботі зі складними залежностями: Стандартні алгоритми можуть бути менш ефективними у вирішенні завдань, де є складні залежності між ознаками на зображеннях.

– Вимоги до підготовки даних: Деякі алгоритми можуть вимагати складної підготовки даних, такої як вибір ознак або нормалізація.

У цьому підрозділі будуть розглянуті конкретні застосування стандартних алгоритмів в завданнях розпізнавання військової техніки на супутникових знімках. Відзначимо приклади успішних застосувань та сферу застосування кожного з алгоритмів. Загальною метою цього розділу є вивчення та зрозуміння стандартних методів розпізнавання зображень та їх можливого застосування в розпізнаванні військової техніки на супутникових знімках.

1.4. Аналіз вимог до програмного модуля і постановка задачі на розробку

Аналіз вимог до програмного модуля та постановка задачі на розробку є критичними етапами у розробці будь-якої програмної системи, включаючи модулі інтелектуального розпізнавання. Цей процес передбачає визначення функціональних та нефункціональних вимог, збір вхідних даних, визначення очікуваних результатів та інших критеріїв успіху проекту. Ось основні кроки аналізу вимог та постановки задачі:

Визначення цілей та обсягу проекту:

– Цілі проекту: Чітке визначення того, що проект має досягти, наприклад, розробка нейронної мережі для розпізнавання певних типів військової техніки.

– Обсяг проекту: Визначення меж проекту, включаючи технології, методики, обмеження та можливості.

Збір вимог:

– Функціональні вимоги: Специфікація завдань, які програмний модуль повинен виконати, наприклад, точність класифікації, швидкість обробки, інтеграція з іншими системами.

– Нефункціональні вимоги: Вимоги до продуктивності, надійності, масштабованості, безпеки та інших атрибутів якості.

Аналіз і моделювання вимог:

– Аналіз: Розбиття вимог на управління, поняття та розробку моделей для більш глибокого розуміння.

– Моделювання: Створення візуальних схем, діаграм, та інших засобів моделювання для представлення вимог та їх взаємозв'язків.

Перевірка та підтвердження вимог:

– Перевірка: Упевнитись, що вимоги є повними, зрозумілими, можливими для виконання, та відповідають потребам користувачів.

– Підтвердження: Отримання схвалення вимог від зацікавлених сторін, включаючи замовників, користувачів та інших членів проектної команди.

Постановка задачі на розробку:

– Формулювання задачі: Опис конкретних технічних завдань та етапів розробки на основі аналізу вимог.

– Планування ресурсів: Визначення необхідних ресурсів, включаючи час, бюджет, команду, та інші ресурси.

– Стратегія реалізації: Розробка плану виконання проекту, визначення методик, технологій, тестування та впровадження.

Процес аналізу вимог та постановки задачі є фундаментом для успішного проектування та розробки програмного модуля. Чітке розуміння вимог, цілей, та обсягу проекту дозволяє створити відповідне рішення, що відповідає всім очікуванням та потребам.

1.5. Висновки до розділу

Аналізування процесів та використовуваних технік є критичним для створення ефективних систем автоматичного розпізнавання образів. Вивчення існуючих методів обробки зображень та пов'язаних наукових робіт дозволило виокремити ключові фази розробки таких систем та техніки, що застосовуються на кожному етапі.

Протягом кожної фази були ідентифіковані специфічні методології та алгоритми для аналізу даних, включаючи техніки бінаризації, виявлення контурів, згладжування і посилення країв, а також використання різноманітних архітектур нейронних мереж для задач класифікації.

Результати дослідження підтвердили, що інтеграція глибоких нейронних мереж з класичними методами обробки зображень забезпечує найвищу точність у виявленні військової техніки на фотографіях. Вивчення існуючих рішень у цій галузі допомогло сформулювати як функціональні, так і нефункціональні вимоги для розробки програмного модуля для інтелектуального розпізнавання військової техніки, а також окреслити завдання для дипломної роботи.

РОЗДІЛ 2

МОДЕЛЮВАННЯ РОБОТИ ПРОГРАМНОГО МОДУЛЯ ІДЕНТИФІКАЦІЇ ВІЙСЬКОВОЇ ТЕХНІКИ НА ЗОБРАЖЕННЯХ

2.1. Аналіз основних процесів предметного середовища

Функціональний аналіз - це один із ключових підходів в сучасному дослідженні і розробці, який дозволяє детально розібрати і зрозуміти функції та характеристики системи, продукту чи послуги. Цей аналіз спрямований на визначення, як система взаємодіє з оточенням, які функції вона виконує, та як ці функції пов'язані одна з одною. У даному розділі ми розглянемо основні аспекти функціонального аналізу та його значення у різних сферах, включаючи наукове дослідження, інженерію, дизайн та менеджмент.

Основні поняття функціонального аналізу

1. Функція: Функція - це основне призначення або завдання, яке система повинна виконувати. Функції системи можуть бути фізичними (наприклад, рух аероплана), функціональними (наприклад, навігація аероплана) або відображати взаємодію з користувачем (наприклад, управління аеропланом пілотом).

2. Характеристика: Характеристика - це конкретний аспект функції, який визначає його властивості та параметри. Наприклад, для функції навігації аероплана характеристиками можуть бути точність, швидкість виконання та вартість реалізації.

3. Взаємодія: Функції системи можуть взаємодіяти між собою або зовнішніми суб'єктами. Взаємодія включає в себе обмін інформацією, енергією або матеріалами між функціями.

4. Взаємозв'язок: Взаємозв'язок показує, як функції взаємодіють між собою. Взаємозв'язок може бути послідовним, паралельним, ієрархічним або іншим.

Методи функціонального аналізу

1. Розкладання на функції: Один із основних методів функціонального аналізу - це розкладання системи на окремі функції. Це допомагає зрозуміти, які функції є ключовими та як вони взаємодіють між собою. Для цього використовуються діаграми або матриці функцій.

2. Аналіз взаємодії: Для кращого розуміння системи важливо дослідити, як функції взаємодіють між собою. Це допомагає виявити можливі питання та проблеми, пов'язані з інтеграцією функцій.

3. Аналіз вимог користувачів: Функціональний аналіз також включає в себе вивчення потреб і вимог користувачів. Це допомагає зорієнтувати розробку чи проектування на відповідні потреби користувачів.

4. Оптимізація функцій: Оптимізація функцій полягає в знаходженні способів поліпшення функцій системи, зменшення витрат або покращення якості виконання.

Застосування функціонального аналізу

1. Наукове дослідження: У наукових дослідженнях функціональний аналіз допомагає визначити, як система чи процес функціонує та які фактори впливають на його результати. Це дозволяє вченим розробляти нові технології та вдосконалювати існуючі.

2. Інженерія: У галузі інженерії функціональний аналіз є важливим етапом при розробці нових продуктів чи систем. Він допомагає інженерам розібратися в тому, як система повинна працювати та які вимоги до неї пред'являються.

3. Дизайн: В дизайні функціональний аналіз використовується для розробки функціональних вимог до продукту та визначення, як ці вимоги впливають на дизайн.

4. Менеджмент проектів: У менеджменті проектів функціональний аналіз допомагає визначити обсяг робіт, необхідний для досягнення мети проекту, та розробити план реалізації.

Функціональний аналіз - це потужний інструмент для розуміння та оптимізації систем, продуктів чи послуг. Він допомагає інженерам, науковцям та

дизайнерам розробляти більш ефективні та відповідні продукти та системи, а також забезпечує чіткість та зрозумілість у вирішенні завдань у всіх галузях діяльності.

Діаграма *IDEF0* програмного модуля інтелектуального розпізнавання військової техніки за знімками аерозвідки

Діаграма *IDEF0* - це інструмент системного аналізу та проектування, що використовується для моделювання та аналізу функцій системи. Вона дозволяє відобразити різні рівні функціональності системи, ідентифікувати взаємозв'язки між ними та з'ясувати, як вони взаємодіють у процесі роботи. У даному підрозділі ми розглянемо застосування діаграми *IDEF0* для програмного модуля інтелектуального розпізнавання військової техніки за знімками аерозвідки.

1. Моделювання функцій: Діаграма *IDEF0* дозволяє створити структурну модель функцій програмного модуля. Це допомагає чітко зрозуміти, які завдання та функції виконує програмний модуль для розпізнавання військової техніки.

2. Аналіз взаємозв'язків: За допомогою діаграми *IDEF0* можна проаналізувати взаємозв'язки між різними функціями програмного модуля. Це допомагає виявити, як одні функції впливають на інші та як вони спільно сприяють досягненню цілей.

3. Ідентифікація слабких місць: Діаграма *IDEF0* дозволяє виявити слабкі місця в програмному модулі, де може бути необхідна оптимізація або вдосконалення функціональності.

Діаграма *IDEF0* складається з різних рівнів функціональності, де кожен рівень представляє собою більш деталізований аналіз функцій. Основні елементи діаграми включають:

1. Функції: Функції відображають завдання, які виконує програмний модуль. На діаграмі вони представлені у вигляді блоків з коротким описом.

2. Рішення та управління: Вказується, як функції приймають рішення та управляють виконанням завдань.

3. Входи та виходи: Показується, які дані вводяться в систему та які результати виводяться з неї.

4. Керованість: Визначається, як керованість інформацією та процесами впливає на функції системи.

Нехай ціль програмного модуля - розпізнавання військової техніки на знімках аерозвідки. Діаграма *IDEF0* може виглядати наступним чином:

Рівень 0 - Системний рівень:

На цьому рівні відображено загальну функцію програмного модуля, яка полягає в розпізнаванні військової техніки на знімках аерозвідки.

Рівень 1 - Основні функції:

1. Зчитування зображення: Функція отримує зображення з аерозвідки.
2. Попередня обробка: Зображення піддається попередній обробці, яка може включати видалення шуму та покращення якості.
3. Сегментація: Визначення областей на зображенні, де може знаходитися військова техніка.

Рівень 2 - Деталізація функцій:

- 3.1. Сегментація техніки: Розпізнавання конкретних областей на знімку, де знаходиться військова техніка.
- 3.2. Витягнення ознак: Визначення характеристик розпізнаних об'єктів.
- 3.3. Класифікація: Визначення типу військової техніки (танк, бронетранспортер і т.д.).

Рівень 3 - Додаткова деталізація функцій:

- 3.1.1. Розпізнавання техніки за формою.
- 3.1.2. Розпізнавання техніки за текстурою.
- 3.2.1. Визначення розмірів техніки.
- 3.2.2. Визначення кутів нахилу та орієнтації.
- 3.3.1. Визначення типу двигуна (дизельний, бензиновий і т.д.).
- 3.3.2. Визначення призначення (бойовий, транспортний і т.д.).

Ця діаграма дозволяє відобразити ієрархію функцій програмного модуля для інтелектуального розпізнавання військової техніки. Кожна функція розкривається на більш детальний рівень для досягнення точності та ефективності розпізнавання.

Діаграма *IDEFO* є потужним інструментом для аналізу та проектування функцій програмного модуля інтелектуального розпізнавання військової техніки. Вона допомагає структурувати завдання, ідентифікувати взаємозв'язки та розкривати деталізацію функціональності, що допомагає досягти високої точності та ефективності у завданні розпізнавання об'єктів на знімках аерозвідки.

Архітектура інтелектуального програмного модуля для розпізнавання військової техніки на знімках аерозвідки є ключовим елементом розробки таких систем. Вона визначає, як взаємодіють різні компоненти системи, які виконують різні завдання, щоб досягти поставленої мети. У цьому розділі ми розглянемо основні аспекти архітектури програмного модуля для розпізнавання військової техніки на знімках аерозвідки.

При розробці інтелектуального програмного модуля для розпізнавання об'єктів на знімках аерозвідки існують різні типи архітектур, які можуть бути використані в залежності від завдань та обмежень. Основні типи архітектур включають:

1. Засновані на правилах системи: В цьому випадку, розпізнавання ґрунтується на заздалегідь визначених правилах та логіці. Ця архітектура вимагає чіткого опису правил та механізмів прийняття рішень.

2. Основані на статистичних методах: Використовуються методи статистичного аналізу для розпізнавання об'єктів. Це може включати в себе використання байєсівських мереж, методів машинного навчання та статистичних класифікаторів.

3. Засновані на нейронних мережах: Використовують глибокі нейронні мережі для аналізу та розпізнавання об'єктів на зображеннях. Ці архітектури, такі як сверточні нейронні мережі (*CNN*), є дуже популярними у завданнях комп'ютерного зору.

4. Гібридні архітектури: Комбінують в собі різні методи та підходи для досягнення кращої продуктивності та точності. Наприклад, можна об'єднати методи основані на правилах і нейронні мережі.

Архітектура інтелектуального програмного модуля для розпізнавання військової техніки містить кілька ключових компонентів:

1. Модуль завантаження зображень: Цей модуль відповідає за завантаження та підготовку зображень для подальшої обробки.
2. Модуль передпроцесингу: Виконує попередню обробку зображень, таку як видалення шуму, нормалізацію та покращення контрасту.
3. Модуль сегментації: Визначає області на зображенні, які можуть містити військову техніку. Сегментація дозволяє виділити ці області для подальшого аналізу.
4. Модуль витягнення ознак: Визначає характеристики об'єктів на сегментованих зображеннях, такі як розмір, форма, текстура тощо.
5. Модуль класифікації: Визначає тип військової техніки на основі витягнутих ознак. Цей модуль використовує класифікатори для визначення класів.
6. Модуль візуалізації і результатів: Відповідає за відображення результатів розпізнавання, наприклад, шляхом позначення виділених об'єктів на зображенні.
7. Модуль навчання (опціонально): Якщо система використовує нейронні мережі, цей модуль відповідає за процес навчання мережі на даних.

Сучасні архітектури інтелектуальних програмних модулів використовують передові технології для покращення точності та продуктивності розпізнавання. Деякі з цих технологій включають:

– Застосування глибокого навчання: Використання глибоких нейронних мереж для розпізнавання об'єктів на зображеннях. Сверточні нейронні мережі (*CNN*) стали основним інструментом у завданнях комп'ютерного зору.

– Автоматизоване навчання: Застосування автоматизованих методів навчання для навчання моделей розпізнавання без потреби у великій кількості ручних анотацій.

– Використання *GPU* і *TPU*: Використання графічних і тензорних процесорів для прискорення обчислень у нейронних мережах.

Архітектура інтелектуального програмного модуля також включає в себе методи оптимізації та підтримки системи. Це може включати в себе:

– Оптимізацію продуктивності: Пошук способів прискорення обробки зображень та розпізнавання.

– Оптимізацію ресурсів: Рациональне використання обчислювальних і пам'яткових ресурсів.

– Підтримку різних типів знімків: Здатність працювати з різними джерелами та форматами знімків аерозвідки.

– Масштабованість: Здатність працювати з великими обсягами даних і розширювати систему в майбутньому

Підрозділ 2.1.2: Діаграма *IDEF0* програмного модуля інтелектуального розпізнавання військової техніки за супутниковими знімками

Діаграма *IDEF0* (*Integration Definition for Function Modeling*) є потужним інструментом для моделювання та аналізу функціональних процесів програмного модуля інтелектуального розпізнавання військової техніки на супутникових знімках. Ця діаграма допомагає краще розуміти структуру, взаємозв'язок та функції різних складових цього модуля.

1. Головна функція (*A0*): Розпізнавання військової техніки

Це головна функція програмного модуля інтелектуального розпізнавання. Вона визначає, що модуль призначений для виявлення і класифікації військової техніки на супутникових знімках.

2. Вхідні дані (*I1*): Супутникові знімки

Ця функція відображає вхідні дані для модуля - супутникові знімки, які подаються на вхід для подальшої обробки та аналізу.

3. Передпроцесинг (*A1*): Обробка супутникових знімків

Перед тим, як зображення буде подано на аналіз, вони піддаються обробці. Це може включати фільтрацію, корекцію кольору, видалення шуму та інші техніки для поліпшення якості зображень.

4. Виявлення об'єктів (*A2*): Визначення потенційних об'єктів

Після передпроцесингу зображення аналізуються для виявлення потенційних об'єктів, які можуть бути військовою технікою.

5. Витягнення ознак (A3): Отримання характеристик об'єктів

На цьому етапі аналізуються виявлені об'єкти для витягнення характеристик, таких як розміри, форма, яскравість, текстура тощо.

6. Класифікація (A4): Визначення типу військової техніки

Класифікація полягає у визначенні типу військової техніки на основі отриманих характеристик. Модель може використовувати різні методи класифікації, такі як нейронні мережі, методи машинного навчання тощо.

7. Вивід результатів (O1): Відображення результатів аналізу

Ця функція відображає результати аналізу на зображенні або картах, щоб оператори та аналітики могли швидко оцінити виявлену військову техніку.

8. Архівація та зберігання (A5): Збереження результатів та даних

Після аналізу результати та дані зберігаються для подальшого використання, а також для архівації та забезпечення їхньої доступності.

Діаграма *IDEF0* відображає послідовність функцій та їхній взаємозв'язок в програмному модулі інтелектуального розпізнавання військової техніки на супутникових знімках, що допомагає краще зрозуміти його робочий процес та структуру.

Підрозділ 2.1.3: Архітектура інтелектуального програмного модуля

Архітектура інтелектуального програмного модуля, призначеного для розпізнавання військової техніки на супутникових знімках, є ключовою складовою успішної реалізації такого рішення. Основна мета цього модуля - виявлення та класифікація військової техніки з використанням методів обробки та аналізу зображень. Нижче розглянуті ключові аспекти архітектури цього модуля:

1. Вхідні дані та обробка:

– Вхідні зображення: Архітектура передбачає прийом супутникових знімків як вхідних даних. Ці зображення можуть бути в різних форматах і розмірах.

– Передпроцесинг: Перед аналізом дані проходять через передпроцесинг. Це включає в себе операції, такі як видалення шуму, корекцію кольору, збільшення чіткості тощо, щоб підготувати зображення до подальшого аналізу.

2. Виявлення об'єктів:

– Згорткові нейронні мережі (*Convolutional Neural Networks, CNN*): Для виявлення потенційних об'єктів використовуються згорткові нейронні мережі. Вони допомагають виділити області інтересу на зображенні, де може знаходитися військова техніка.

– Архітектура мережі: Архітектура *CNN* включає різні шари, такі як згорткові шари, шари пулінгу та повністю зв'язані шари. Вони допомагають виявити та виділити особливості на зображенні.

3. Витягнення ознак:

– Операції витягнення ознак: Після виявлення об'єктів на зображенні виконують операції витягнення ознак. Це може включати в себе обчислення розмірів, форми, текстури та інших характеристик об'єктів.

4. Класифікація:

– Модель класифікації: Отримані ознаки подаються на входи моделі для класифікації. Модель може використовувати різні методи класифікації, такі як нейронні мережі або методи машинного навчання.

– Навчання моделі: Перед використанням моделі для розпізнавання, її необхідно навчити на великій кількості попередньо класифікованих зображень військової техніки.

5. Вивід результатів:

– Класифікація та локалізація: Модель може повертати результати класифікації (тип військової техніки) та, можливо, локалізації (розташування на зображенні).

– Візуалізація результатів: Результати можуть бути візуалізовані на вихідних зображеннях, щоб користувач міг побачити, де була знайдена військова техніка.

6. Обробка помилок і підтримка:

– Обробка помилок: Важливо передбачити обробку помилок, оскільки розпізнавання на супутникових знімках може бути неточним. Потрібно розробити методи для виправлення помилкових класифікацій.

– Підтримка та оновлення: Архітектура повинна бути гнучкою для можливості оновлення моделі з врахуванням нових даних та покращень.

Архітектура інтелектуального програмного модуля для розпізнавання військової техніки на супутникових знімках включає в себе комплексний набір процесів, які починаються з обробки вхідних даних та завершуються виводом результатів класифікації. Ця архітектура створює основу для розробки програмного забезпечення, яке може визначати та класифікувати військову техніку на супутникових знімках з високою точністю.

2.2. Проєктування інформаційного забезпечення програмного модуля інтелектуального розпізнавання військової техніки

Проєктування інформаційного забезпечення програмного модуля для інтелектуального розпізнавання військової техніки є критичним етапом розробки, оскільки воно визначає структуру даних, взаємозв'язки між компонентами, а також способи зберігання, обробки та передачі інформації. У цьому розділі ми розглянемо ключові аспекти проєктування інформаційного забезпечення для програмного модуля інтелектуального розпізнавання військової техніки.

Об'єктно-орієнтований дизайн (ООД) є підходом до проєктування програмного забезпечення, де система розглядається як набір об'єктів, які взаємодіють між собою. В контексті інтелектуального розпізнавання військової техніки, об'єктами можуть бути зображення, моделі нейронних мереж, результати класифікації та інші.

ООД спрощує розробку та підтримку програмного модуля, оскільки код може бути розділений на невеликі, самостійні об'єкти, кожен з яких відповідає за конкретну функціональність. Наприклад, можна створити об'єкти для роботи зі зображеннями, обробки даних, взаємодії з користувачем тощо. Кожен об'єкт має власний стан і методи, які дозволяють йому взаємодіяти з іншими об'єктами.

Вибір мови програмування та технологій є важливим аспектом проектування інформаційного забезпечення. Для програмного модуля інтелектуального розпізнавання військової техніки, де обробка зображень та нейронні мережі грають важливу роль, можливі варіанти включають *Python*, *TensorFlow*, *Keras*, *PyTorch* та *OpenCV*.

– *Python* є популярною мовою для роботи зі штучним інтелектом, оскільки в ній доступні багато бібліотек та фреймворків для глибокого навчання.

– *TensorFlow* та *PyTorch* - це основні фреймворки для розробки та навчання нейронних мереж. Вони надають інструменти для побудови, навчання та впровадження нейронних мереж різної складності.

– *Keras* є вищорівневим інтерфейсом для *TensorFlow* та *PyTorch*, що спрощує розробку нейронних мереж.

– *OpenCV* - це бібліотека для обробки зображень, включаючи сегментацію та витягнення ознак. Вона допомагає ефективно обробляти та аналізувати зображення.

Вибір мови та технологій повинен бути зроблений на основі потреб конкретного проекту, доступного обладнання та рівня експертизи розробників.

Сховище даних визначає, де та як будуть зберігатися дані, необхідні для роботи програмного модуля. У випадку інтелектуального розпізнавання військової техніки, це може бути сховище для зображень, параметрів моделей нейронних мереж, результатів класифікації та інших даних.

Варіанти сховищ даних включають:

– Локальні файлові системи: Для невеликих обсягів даних можна використовувати локальні файли та каталоги на сервері.

– Реляційні бази даних: Якщо довідкові дані обширні або потрібна складна структура даних, реляційні бази даних, такі як *MySQL* або *PostgreSQL*, можуть бути вибором.

– Безсховищові бази даних: Для зберігання структурованих або неструктурованих даних можна розглядати *NoSQL* бази даних, такі як *MongoDB* або *Cassandra*.

– Хмарні сховища: Популярні хмарні рішення, такі як *Amazon S3* або *Google Cloud Storage*, надають масштабованість і доступність для зберігання даних.

Графічно функції представлено на рис. 2.1.

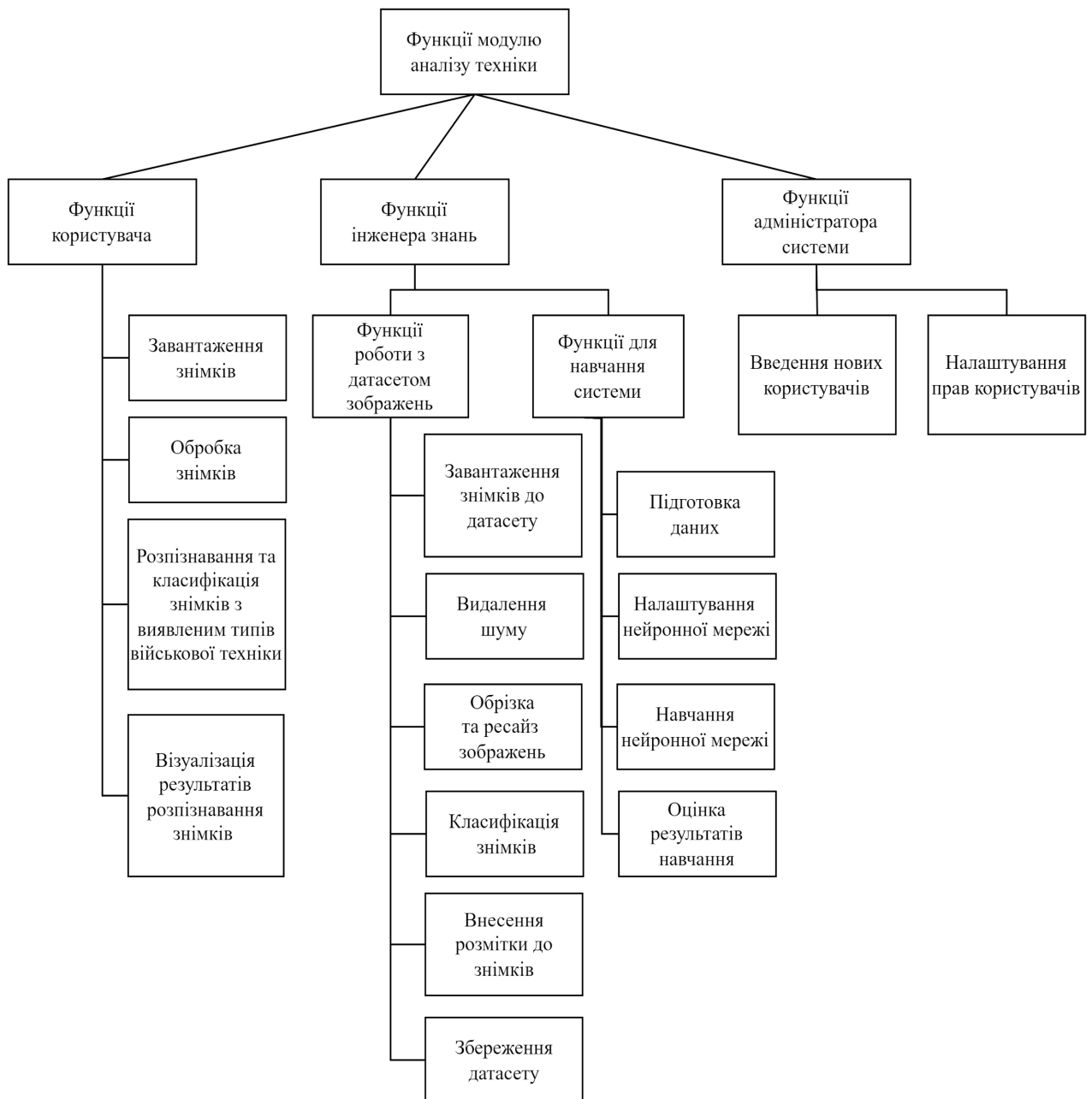


Рис. 2.1. Дерево функцій програмного модуля інтелектуального розпізнавання військової техніки

Вибір сховища даних повинен враховувати обсяг даних, швидкість доступу, потреби в резервному копіюванні та безпеці даних.

Інтерфейс користувача визначає, як користувачі будуть взаємодіяти з програмним модулем. У випадку інтелектуального розпізнавання військової

техніки, інтерфейс може включати в себе веб-інтерфейс, десктоп-додаток, *API* для інтеграції з іншими системами тощо.

Важливо враховувати потреби користувачів та забезпечити зручний та інтуїтивний інтерфейс. Розробка інтерфейсу повинна включати в себе проектування віджетів, створення графічних елементів та забезпечення відповідності дизайну стандартам і бест-практикам.

Наступним етапом є опис принципу функціонування застосунку для інтелектуальної системи розпізнавання військової техніки.

розпізнавання військової техніки за знімками аерозвідки

Контекстна діаграма "ЯК БУДЕ" на рис. 2.2.

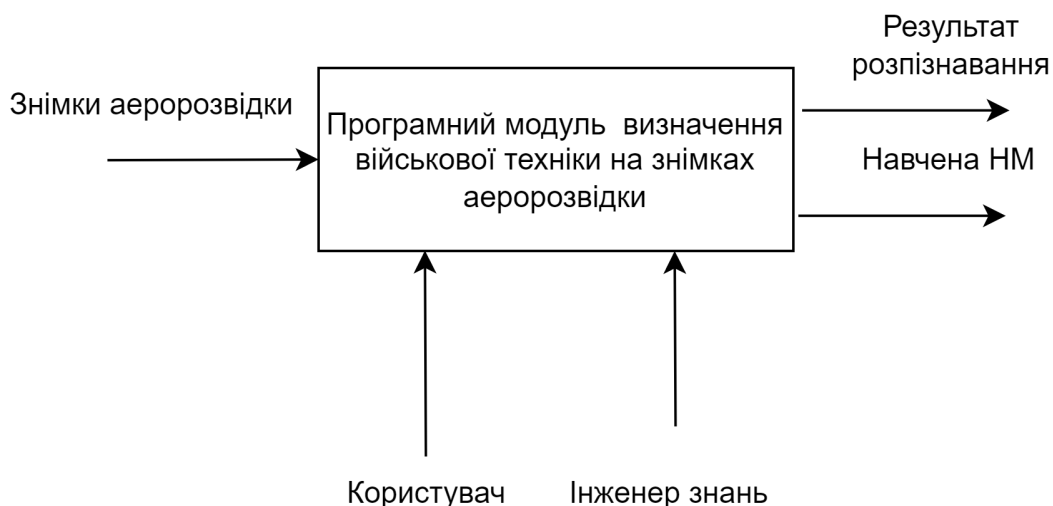


Рис. 2.2. Контекстна діаграма програмного модуля ідентифікації військової техніки за знімками аерозвідки

Декомпозиція контекстної діаграми розпізнавання знімків аерозвідки та навчання нейронної мережі (рис. 2.3), на рис. 2.4 представлено діаграму *IDEF0* рівень 1 процесу завантаження та попередньої обробки знімків аерозвідки, на рис. 2.5 діаграма декомпозиції процесу навчання нейронної мережі розпізнавання знімків аерозвідки, на рис. 2.6. діаграма процесу завантаження та аналізу датасету.

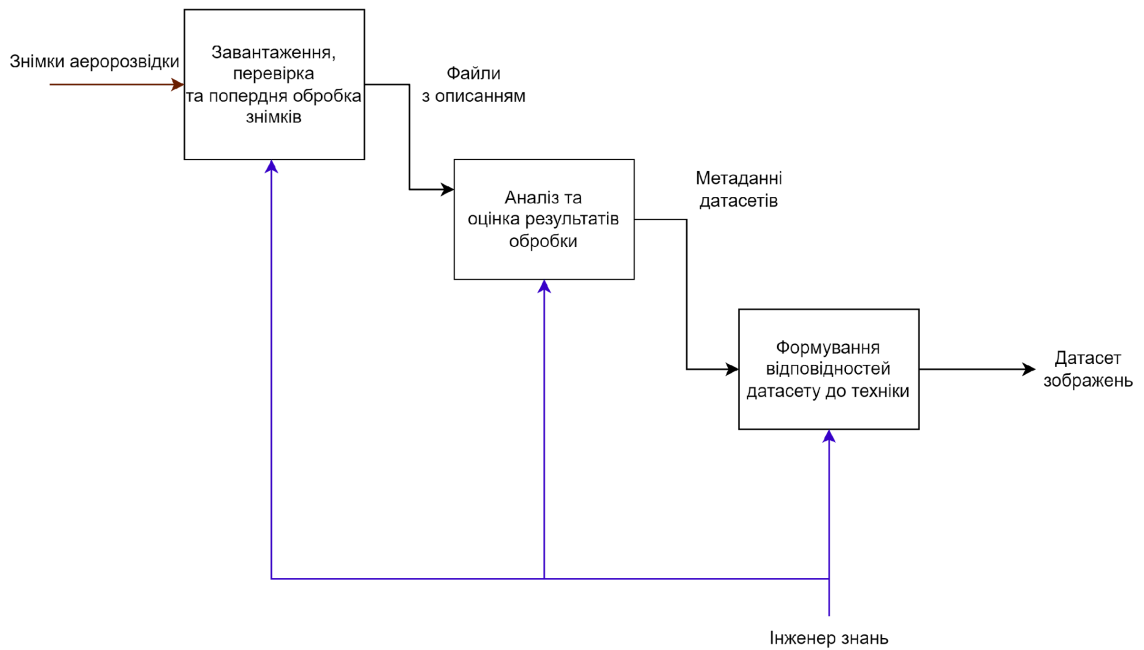


Рис. 2.3. Діаграма програмного модуля ідентифікації військової техніки за знімками аерозвідки рівня 1

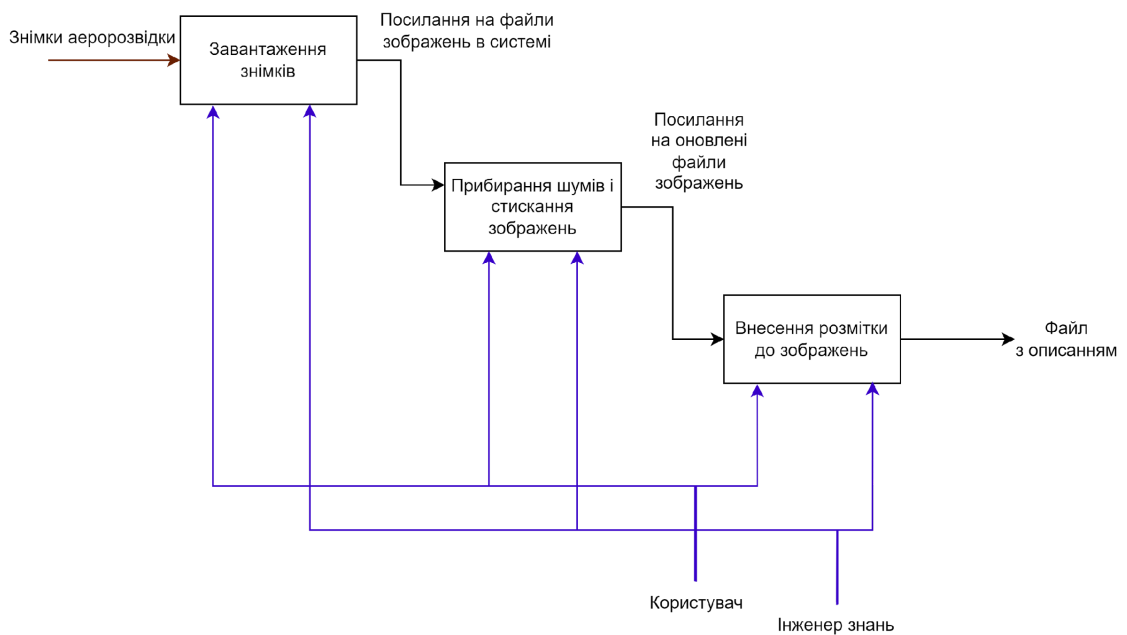


Рис. 2.4. Діаграма IDEF0 рівень 1 процесу завантаження та попередньої обробки знімків аерозвідки

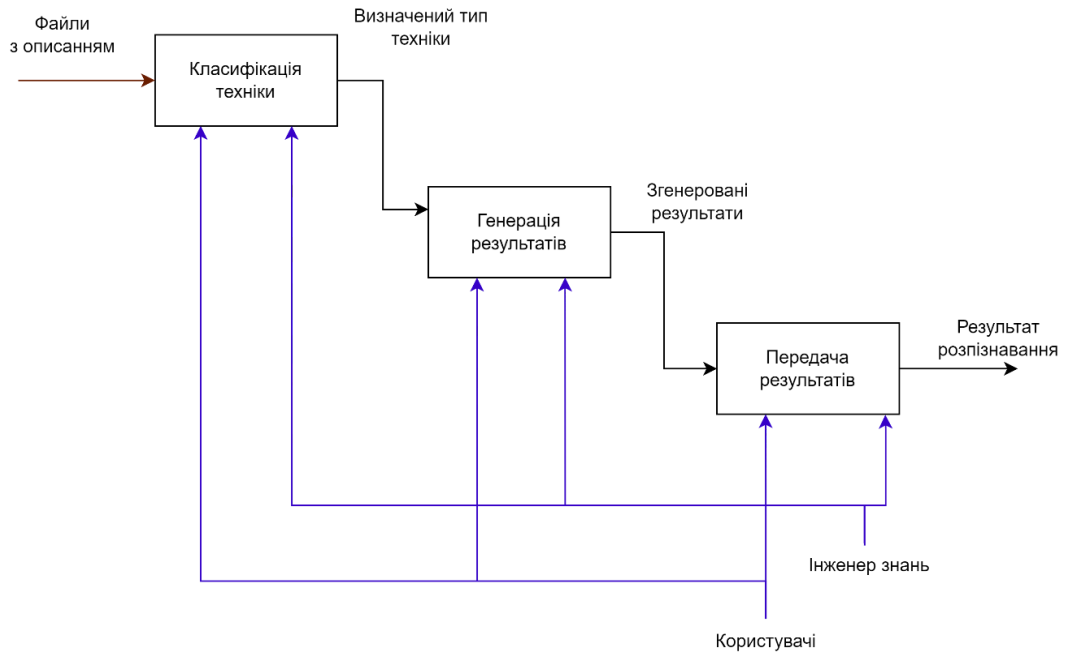


Рис. 2.5. Діаграма *IDEF0* рівень 1 процесу розпізнавання знімків аерозвідки

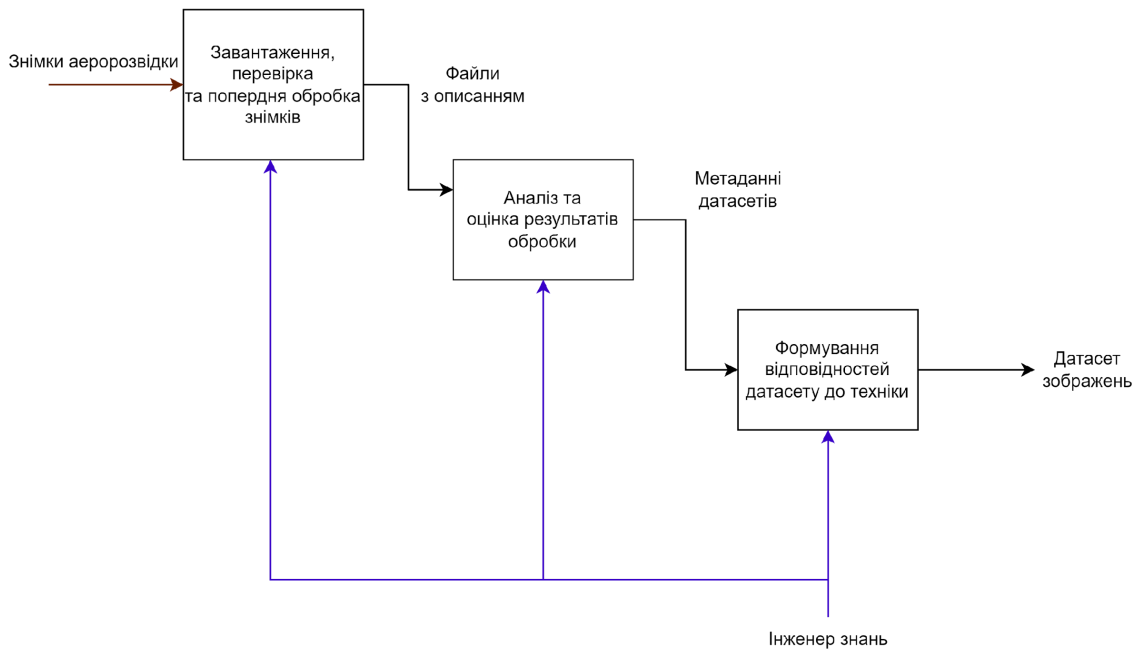


Рис. 2.6. Діаграма *IDEF0* рівень 1 процесу завантаження та аналізу датасету

На рис. 2.7. представлено діаграму у форматі *IDEF0* рівень 1 процесу навчання нейронної мережі розпізнавання знімків аерозвідки.

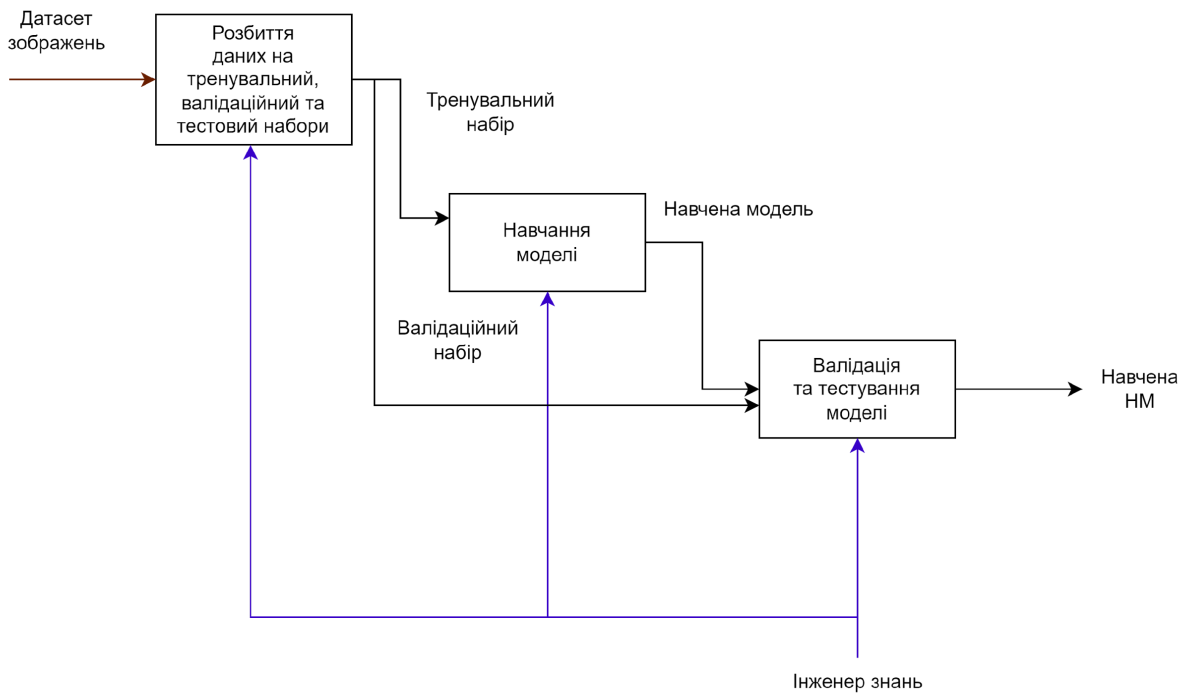


Рис. 2.7. Діаграма IDEF0 рівень 1 процесу навчання нейронної мережі розпізнавання знімків аерозвідки

На основі декомпозиції процесів і функціональних вимог до програмного модуля можна візуалізувати *use-case* діаграму (рис. 2.8).

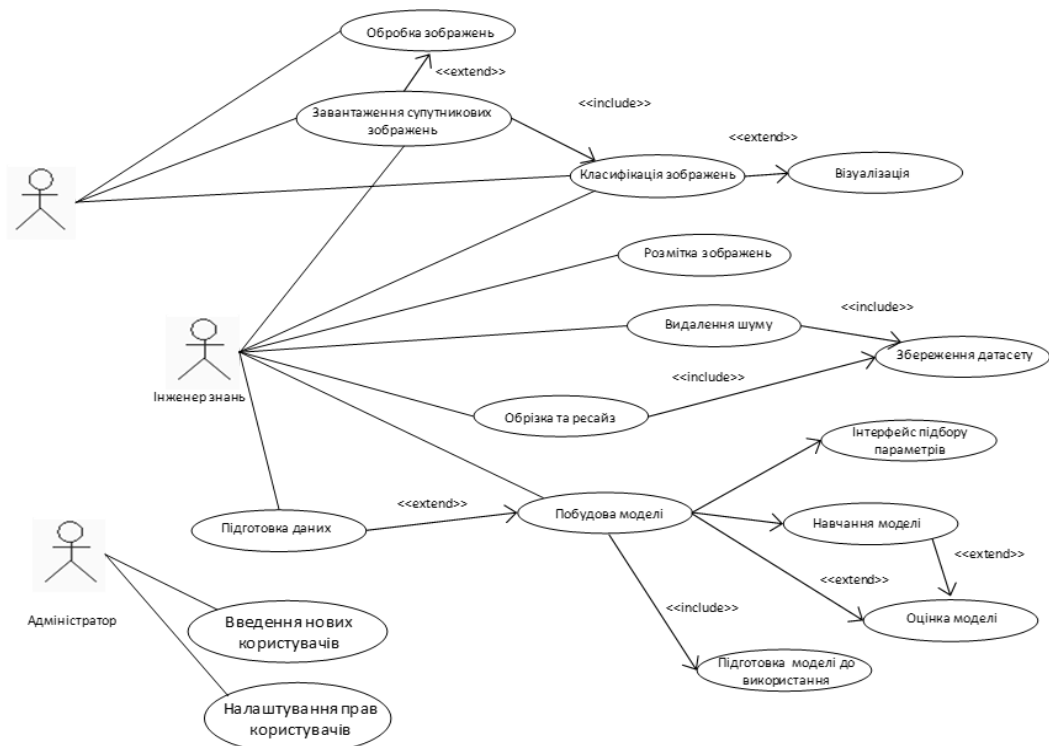


Рис. 2.8. Діаграма використання модуля

В діаграмі передбачено три актори, але функції в ході розробки системи є сенс винести функції адміністратора за межі програмного моду інтелектуального розпізнавання військової техніки за знімками аерозвідки.

Архітектура програмного модуля є фундаментальним аспектом його розробки та визначає структуру, взаємозв'язки компонентів і принципи роботи системи. У випадку інтелектуального програмного модуля для розпізнавання військової техніки на знімках аерозвідки, правильно спроектована архітектура є важливою для досягнення високої продуктивності та точності.

Монолітна архітектура є традиційним підходом до розробки програмних модулів і полягає в тому, що весь функціонал розміщений у єдиному блоку. У випадку інтелектуального програмного модуля для розпізнавання військової техніки це означає, що всі етапи обробки зображення, включаючи сегментацію та класифікацію, виконуються в одному компоненті.

Переваги монолітної архітектури:

– Простота: Монолітна архітектура виглядає простою і зрозумілою для розробників.

– Висока продуктивність: Оскільки всі компоненти об'єднані, немає накладних витрат на комунікацію між ними.

Недоліки монолітної архітектури:

– Складність масштабування: Важко додавати новий функціонал або змінювати існуючий без ризику зламу іншого функціоналу.

– Низька резервованість: Якщо один компонент виводиться з ладу, це може вплинути на всю систему.

– Складність тестування: Важко тестувати окремі частини функціоналу.

Розподілена архітектура передбачає, що різні функції або компоненти програмного модуля розміщені на різних вузлах мережі і взаємодіють за допомогою мережевого протоколу. У випадку інтелектуального програмного модуля, це означає, що окремі компоненти, такі як сегментація, витягнення ознак та класифікація, можуть виконуватися на окремих вузлах.

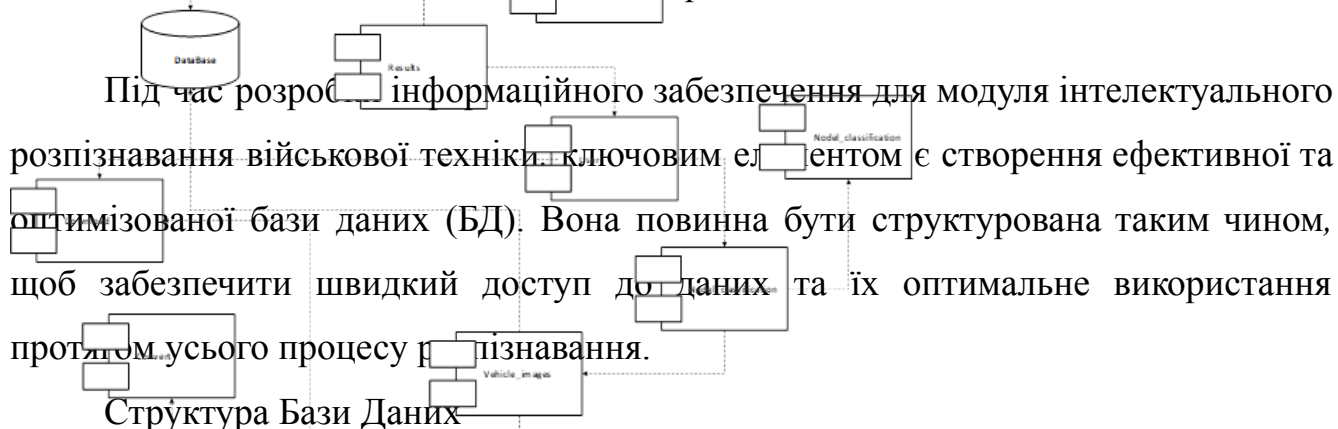
Переваги розподіленої архітектури:

– Масштабованість: Легше додавати нові вузли або масштабувати існуючі для забезпечення високої продуктивності.

– Вища резервованість: При виході з ладу одного вузла, інші можуть продовжувати працювати

Наступним етапом є розробка архітектури програмного модуля (рис. 2.9). Дана схема відображає структуру і взаємозв'язок між компонентами.

Рис. 2.9. Діаграма компонентів модуля інтелектуального розпізнавання об'єктів на знімках аерозвідки



1. Концептуальна модель (рис. 2.10):

– Визначення елементів: Ідентифікація всіх необхідних даних, таких як типи військової техніки, характеристики, зображення, метадані тощо.

– Визначення зв'язків: Встановлення зв'язків між різними елементами даних для моделювання відносин та ієрархій.

– Візуалізація: Графічне представлення елементів і зв'язків для чіткого розуміння структури та потоків даних.

2. Логічна модель БД (рис. 2.11):

– Таблиці та поля: Перетворення концептуальної моделі у таблиці з відповідними полями, що відображають необхідні типи даних.

– Визначення ключів: Встановлення первинних та зовнішніх ключів для забезпечення цілісності даних та ефективного пошуку.

– Нормалізація: Оптимізація структури для уникнення дублювання даних та забезпечення швидкодії.

3. Реалізація БД:

– Вибір СУБД: Визначення системи управління базами даних, яка найкраще відповідає технічним та функціональним вимогам проекту.

– Створення схеми: Реалізація логічної моделі у вигляді фізичної схеми БД з усіма таблицями, полями та індексами.

– Заповнення даними: Імпорт або введення необхідних даних, включаючи зображення, характеристики та іншу важливу інформацію.

Процес Розробки

1. Аналіз вимог:

– Збір та аналіз вимог до інформаційного забезпечення, включаючи типи даних, частоту доступу, операції з даними тощо.

2. Проєктування:

– Розробка концептуальної та логічної моделі на основі аналізу вимог.

3. Вибір технологій:

– Вибір підходящих технологій та інструментів для реалізації БД.

4. Реалізація та тестування:

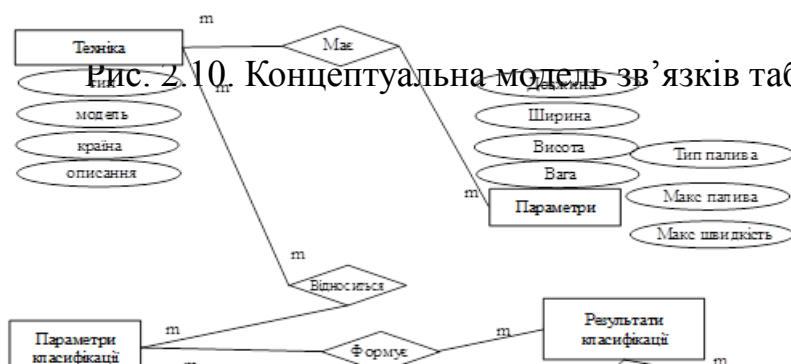
– Створення та оптимізація БД, наповнення її даними та проведення тестування для забезпечення її функціональності та ефективності.

5. Інтеграція та деплоймент:

– Інтеграція БД з іншими компонентами системи інтелектуального розпізнавання та її впровадження у робоче середовище.

Створення ефективної бази даних є важливим кроком у проєктуванні інформаційного забезпечення програмного модуля інтелектуального розпізнавання військової техніки. Це забезпечує основу для зберігання, обробки та аналізу даних, що є необхідними для ефективної роботи системи. Кожен етап від проєктування до впровадження вимагає ретельного планування та виконання для забезпечення, що інформаційна система є надійною, швидкою та масштабованою.

Концептуальна модель на рис. 2.10.



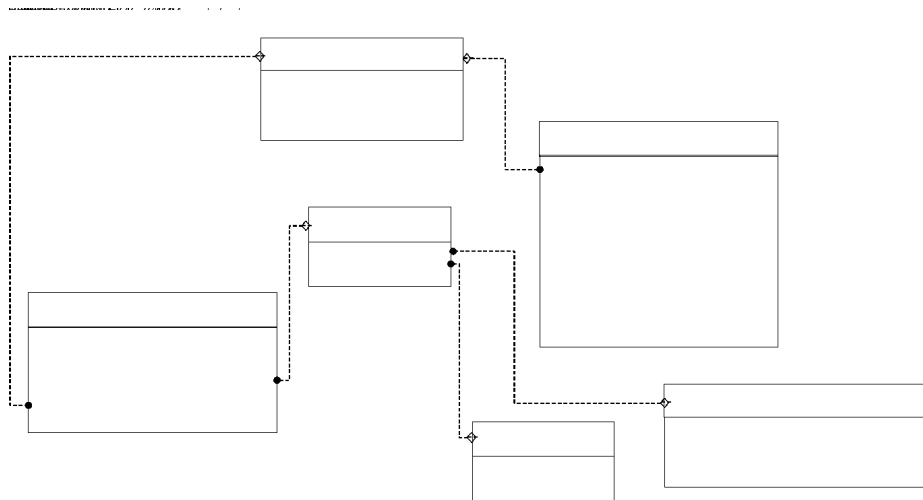


Рис. 2.11. Логічна модель зв'язків таблиці БД

Проектування інформаційного забезпечення програмного модуля для інтелектуального розпізнавання військової техніки є критичним етапом розробки, оскільки воно визначає структуру даних, взаємозв'язки між компонентами, а також способи зберігання, обробки та передачі інформації. У цьому розділі ми розглянемо ключові аспекти проектування інформаційного забезпечення для програмного модуля інтелектуального розпізнавання військової техніки.

Об'єктно-орієнтований дизайн (ООД) є підходом до проектування програмного забезпечення, де система розглядається як набір об'єктів, які взаємодіють між собою. В контексті інтелектуального розпізнавання військової техніки, об'єктами можуть бути зображення, моделі нейронних мереж, результати класифікації та інші.

ООД спрощує розробку та підтримку програмного модуля, оскільки код може бути розділений на невеликі, самостійні об'єкти, кожен з яких відповідає за конкретну функціональність. Наприклад, можна створити об'єкти для роботи зі зображеннями, обробки даних, взаємодії з користувачем тощо. Кожен об'єкт має власний стан і методи, які дозволяють йому взаємодіяти з іншими об'єктами.

Вибір мови програмування та технологій є важливим аспектом проектування інформаційного забезпечення. Для програмного модуля інтелектуального розпізнавання військової техніки, де обробка зображень та нейронні мережі грають важливу роль, можливі варіанти включають *Python*, *TensorFlow*, *Keras*, *PyTorch* та *OpenCV*.

– *Python* є популярною мовою для роботи зі штучним інтелектом, оскільки в ній доступні багато бібліотек та фреймворків для глибокого навчання.

– *TensorFlow* та *PyTorch* - це основні фреймворки для розробки та навчання нейронних мереж. Вони надають інструменти для побудови, навчання та впровадження нейронних мереж різної складності.

– *Keras* є вищорівневим інтерфейсом для *TensorFlow* та *PyTorch*, що спрощує розробку нейронних мереж.

– *OpenCV* - це бібліотека для обробки зображень, включаючи сегментацію та витягнення ознак. Вона допомагає ефективно обробляти та аналізувати зображення.

Вибір мови та технологій повинен бути зроблений на основі потреб конкретного проекту, доступного обладнання та рівня експертизи розробників.

Сховище даних визначає, де та як будуть зберігатися дані, необхідні для роботи програмного модуля. У випадку інтелектуального розпізнавання військової техніки, це може бути сховище для зображень, параметрів моделей нейронних мереж, результатів класифікації та інших даних.

Варіанти сховищ даних включають:

– Локальні файлові системи: Для невеликих обсягів даних можна використовувати локальні файли та каталоги на сервері.

– Реляційні бази даних: Якщо довідкові дані обширні або потрібна складна структура даних, реляційні бази даних, такі як *MySQL* або *PostgreSQL*, можуть бути вибором.

– Безсховищові бази даних: Для зберігання структурованих або неструктурованих даних можна розглядати *NoSQL* бази даних, такі як *MongoDB* або *Cassandra*.

– Хмарні сховища: Популярні хмарні рішення, такі як *Amazon S3* або *Google Cloud Storage*, надають масштабованість і доступність для зберігання даних.

Вибір сховища даних повинен враховувати обсяг даних, швидкість доступу, потреби в резервному копіюванні та безпеці даних.

Інтерфейс користувача визначає, як користувачі будуть взаємодіяти з програмним модулем. У випадку інтелектуального розпізнавання військової техніки, інтерфейс може включати в себе веб-інтерфейс, десктоп-додаток, *API* для інтеграції з іншими системами тощо.

Важливо враховувати потреби користувачів та забезпечити зручний та інтуїтивний інтерфейс. Розробка інтерфейсу повинна включати в себе проектування віджетів, створення графічних елементів та забезпечення відповідності дизайну стандартам і бест-практикам.

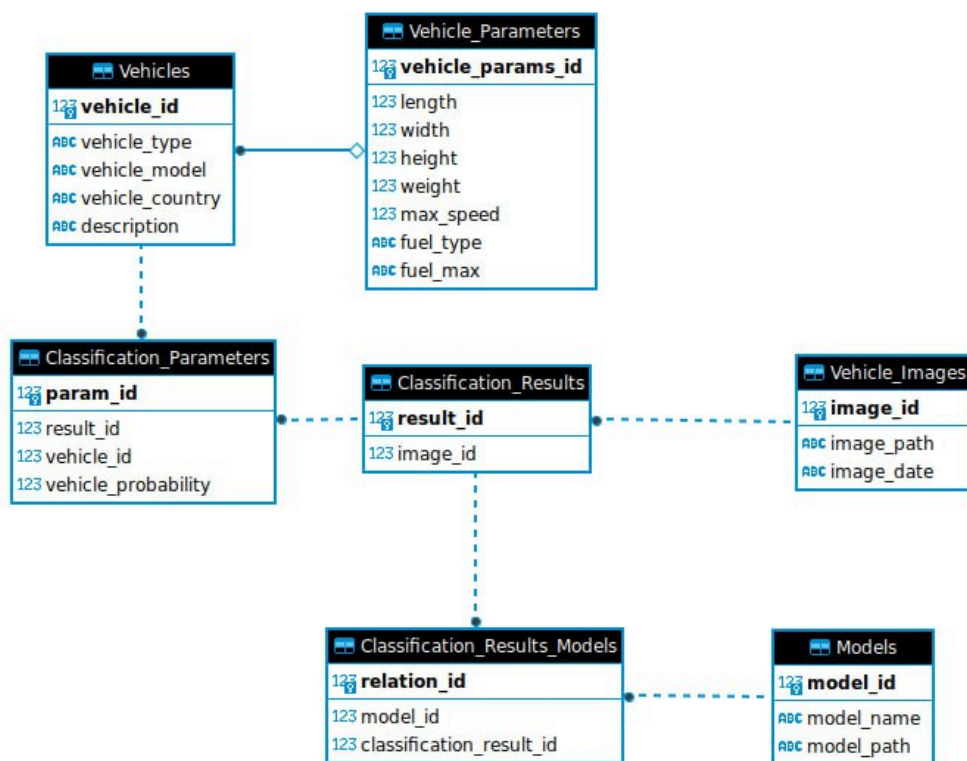


Рис. 2.12. Фізична модель БД

2.3. Проектування інтерфейсу програмного модуля інтелектуального розпізнавання військової техніки

При проектуванні інтерфейсу програмного модуля для інтелектуального розпізнавання військової техніки, важливо забезпечити зручність, інтуїтивність та

функціональність для користувачів. Ось деталізація основних вікон інтерфейсу та їхніх прототипів:

1. Вікно завантаження знімків аеророзвідки (рис. 2.13)

Функціональність:

– Завантаження знімків: Користувачі можуть завантажувати знімки аеророзвідки через діалогове вікно або перетягуванням файлів.

– Формати зображень: Підтримка різних форматів зображень (*JPEG, PNG, TIFF* тощо).

– Попередній перегляд: Відображення мініатюр завантажених зображень для підтвердження та перевірки користувачем.

– Параметри обробки: Налаштування параметрів обробки (наприклад, рівень деталізації, специфічні фільтри).

Дизайн інтерфейсу:

– Чіткий та зрозумілий лейаут з кнопками та інструкціями для завантаження.

– Візуальні керівництва або пояснення доступних опцій та форматів.

2. Вікно результатів розпізнавання знімків аеророзвідки (рис. 2.14)

Функціональність:

– Відображення результатів: На знімках позначаються виявлені об'єкти з вказівкою класу військової техніки.

– Деталізація: Кожен об'єкт може мати додаткову інформацію: тип, опис, можливі дії.

– Фільтрація результатів: Можливість сортувати або фільтрувати результати за різними параметрами (наприклад, тип техніки, рівень впевненості).

Дизайн інтерфейсу:

– Вікно з чітким розділенням на зону перегляду зображення та зону деталей і команд.

– Інтерактивні елементи для взаємодії з результатами (наприклад, збільшення зображення, вибір об'єктів).

3. Вікно додавання класів військової техніки (рис. 2.15)

Функціональність:

– Управління класами: Можливість додавати нові або редагувати існуючі класи військової техніки.

– Категоризація: Впорядкування класів за категоріями або типами для легшого доступу та управління.

– Імпорт/експорт класів: Функції для масового імпорту або експорту класів для використання в інших системах чи проектах.

Дизайн інтерфейсу:

– Просте та інтуїтивно зрозуміле меню для додавання та редагування класів.

– Візуальні елементи для категоризації та організації класів.

– Вбудовані інструкції або підказки для керування процесом.

Загальні рекомендації для проектування:

– Інтуїтивність та зручність: Інтерфейс повинен бути легким для розуміння та використання, навіть для користувачів без глибоких технічних знань.

– Масштабованість: Інтерфейс має бути готовим до розширення та інтеграції з іншими системами або модулями.

– Адаптивність: Підтримка різних розмірів екранів та пристроїв, особливо якщо очікується мобільне використання.

– Безпека: Забезпечення конфіденційності та захисту даних, особливо важливо для військових або розвідувальних даних.

Проектування інтерфейсу цього програмного модуля є ключовим етапом, що вимагає уваги до деталей, розуміння потреб користувачів та технічних можливостей сучасних технологій. Від ефективності та зручності інтерфейсу значною мірою залежить успіх та корисність всього програмного продукту.

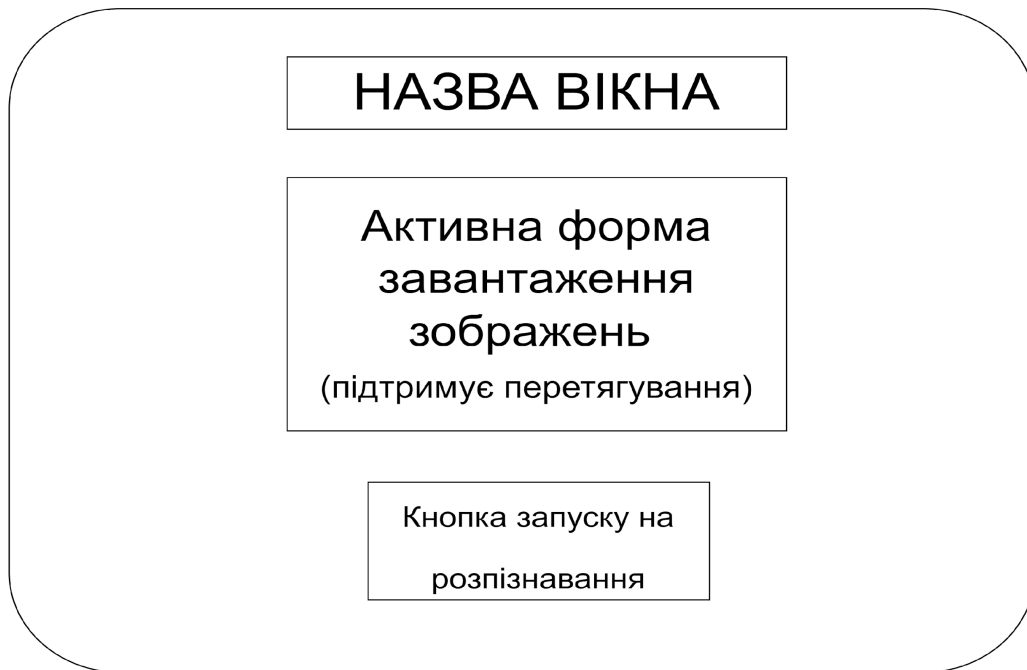


Рис. 2.13. Прототип вікна завантаження знімків аерозвідки

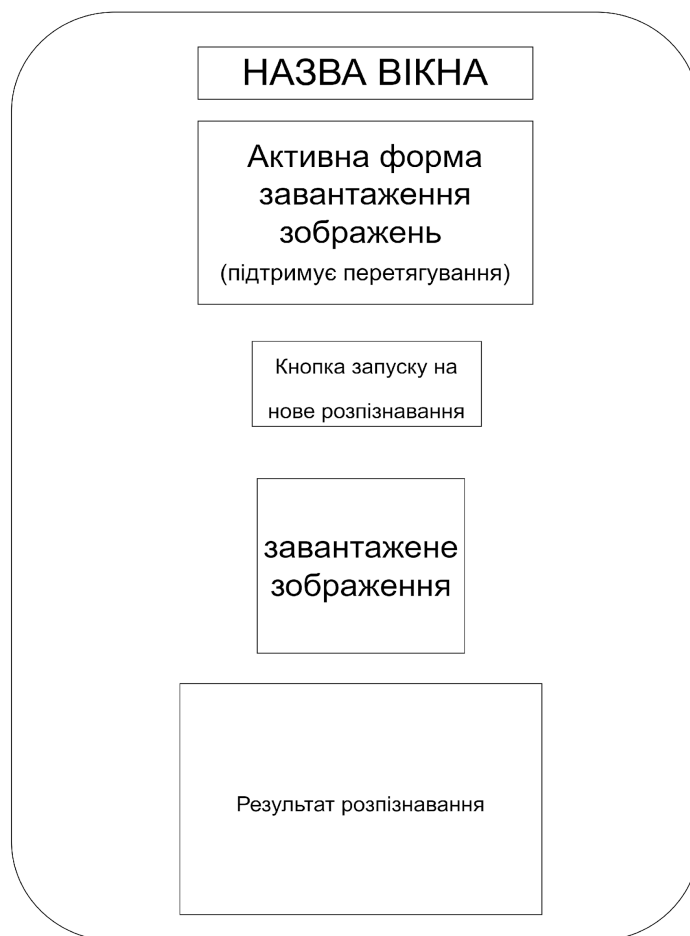


Рис. 2.14. Прототип вікна результатів розпізнавання знімків аерозвідки

The image shows a web form for adding military equipment classes. At the top, there is a button labeled "To main page". Below it, the form is organized into two columns. The left column contains labels for "Vehicle Type:", "Vehicle Model:", "Vehicle Country:", and "Description:". The right column contains labels for "Length:", "Width:", "Height:", "Weight:", "Max Speed:", "Fuel Type:", and "Fuel Max:". Each label is followed by an empty text input field. At the bottom center of the form, there is a "Submit" button.

Рис. 2.15. Прототип вікна для додавання класів військової техніки

2.4. Висновки до розділу

У ході розробки програмного модуля ідентифікації військової техніки було виконано аналіз функцій та створено дерево функцій, що відображає основні бізнес-процеси системи. Додатково були створені діаграми *Event-Driven Process Chain*, що відображають логіку дій бота та його реакції на різні події.

Для візуалізації майбутніх процесів були використані діаграми *IDEFO* "Майбутнє стан", які демонструють етапи обробки та видачі результатів на основі аналізу зображень.

Також була розроблена бізнес-архітектура модуля, включаючи опис внутрішніх компонентів та їх взаємодії. Це дозволило чітко визначити структуру та функціональність системи. В результаті цієї роботи була також створена оптимізована база даних, призначена для зберігання та обробки даних, що використовуються програмним модулем для розпізнавання військової техніки.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ, ТЕСТОВІ ПРИКЛАДИ

3.1. Вибір програмного інструментарію для розробки програмного модуля

Вибір програмного інструментарію є одним з перших та найважливіших кроків у процесі розробки програмного модуля. Правильно підібрані інструменти можуть значно спростити розробку, підвищити ефективність та якість кінцевого продукту. Ось основні аспекти, на які слід звернути увагу при виборі інструментарію:

Мова програмування:

– *Python*: Часто використовується у розробці систем штучного інтелекту та машинного навчання через велику кількість бібліотек (*TensorFlow*, *Keras*, *PyTorch*) та спільноту.

– *C++/C#*: Вибір для високопродуктивних систем та тих, що вимагають інтеграції з іншим програмним забезпеченням.

– *Java/JavaScript*: Підходить для веб-орієнтованих систем або універсальних рішень.

Фреймворки та бібліотеки:

– *TensorFlow* та *Keras*: Для розробки та тренування нейронних мереж.

– *PyTorch*: Вибір багатьох дослідників у галузі штучного інтелекту за зручність та потужні можливості.

– *Scikit-learn*: Для класичного машинного навчання.

– *OpenCV*: Для обробки зображень та відео.

Інструменти розробки:

– *Integrated Development Environment (IDE)*: Вибір середовища розробки, такого як *PyCharm*, *Visual Studio*, або *Eclipse*, в залежності від мови програмування.

– *Version Control Systems (VCS)*: Використання систем контролю версій, як-от *Git*, для керування кодом та співпраці в команді.

– *Containers/Docker*: Для ізоляції та легкого розгортання проекту.

Інструменти тестування та валідації:

– *Unit Testing Frameworks*: Наприклад, *pytest* для *Python*.

– *Continuous Integration/Continuous Deployment (CI/CD) tools*: Автоматизація тестування та розгортання, наприклад, *Jenkins* або *Travis CI*.

Вибір інструментарію на основі специфіки проекту:

– Специфіка задачі: Вибір залежить від типу задачі, обсягу даних, вимог до швидкодії, та інших технічних характеристик.

– Бюджет та ресурси: Вибір інструментів з огляду на наявність фінансових та обчислювальних ресурсів.

– Досвід команди: Предпочтєння та досвід команди також впливають на вибір мов програмування та інструментів.

Вибір відповідних інструментів є важливим для забезпечення гладкого процесу розробки та високої якості кінцевого продукту. Важливо забезпечити баланс між потужністю, гнучкістю, зручністю використання та вартістю інструментарію.

3.2. Розроблені алгоритми для підтримки роботи модуля

Рис. 3.1. представляє схему алгоритму попередньої обробки і тренування моделі, яка включає наступні етапи:

Етап 1: Попередня обробка

1. Завантаження даних:

– Зчитування зображень або даних з вказаного джерела.

2. Передобробка зображень:

– Масштабування: Зміна розмірів зображень до однакових розмірів.

– Нормалізація: Приведення значень пікселів до заданого діапазону (наприклад, 0-1).

– Покращення зображень: Застосування методів покращення зображень, як-от фільтрація шуму або корекція контрасту.

3. Аугментація даних (за потреби):

– Застосування технік аугментації зображень для збільшення та різноманітності набору тренувальних даних.

Етап 2: Розбиття на набори

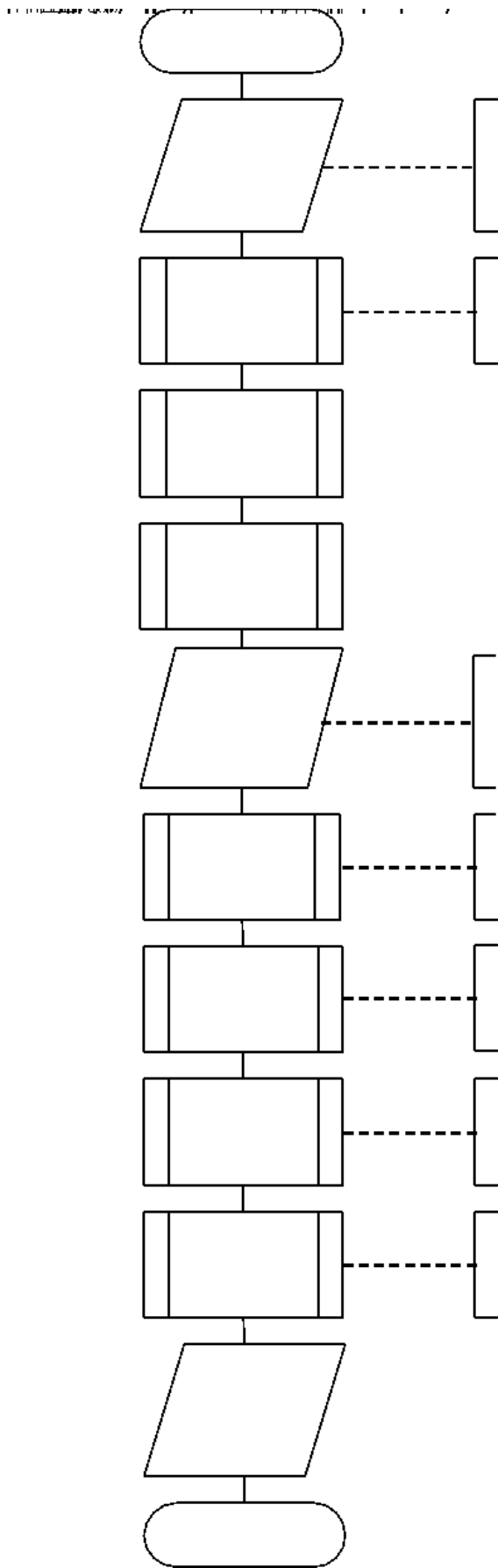


Рис. 3.1. Схема алгоритму попередньої обробки і тренування моделі

1. Розділення даних:

– Розділення даних на тренувальний, валідаційний, та тестовий набори.

Етап 3: Підготовка моделі

1. Вибір архітектури моделі:

– Визначення структури та параметрів нейронної мережі, які будуть використовуватися для навчання.

2. Ініціалізація параметрів:

– Встановлення початкових значень вагів і змінних.

Етап 4: Тренування моделі

1. Навчання:

– Подача тренувальних даних у мережу та корекція вагів на основі обчислення помилки.

2. Валідація:

– Оцінка продуктивності моделі на валідаційному наборі даних та тонке налаштування гіперпараметрів.

Етап 5: Оцінка моделі

1. Тестування:

– Перевірка кінцевої продуктивності моделі на тестовому наборі даних.

2. Аналіз результатів:

– Оцінка точності, повноти, F -міри, та інших метрик для визначення ефективності моделі.

Етап 6: Збереження та розгортання

1. Збереження моделі:

– Збереження навченої моделі для подальшого використання або розгортання.

2. Розгортання:

– Інтеграція моделі в потрібне середовище або додаток для реального використання.

На рисунку 3.1 можуть бути візуально представлені всі ці кроки у вигляді потокової діаграми або блок-схеми, що демонструє послідовність операцій та

переходи між різними стадіями обробки та навчання. Зрозуміле графічне представлення може сприяти кращому розумінню процесу та полегшити спілкування між розробниками, аналітиками та іншими зацікавленими сторонами.

Для оптимізації процесу розпізнавання військової техніки в зображеннях аеророзвідки був розроблений спеціалізований алгоритм попереднього аналізу зображень. Цей алгоритм має на меті виявити та виконати необхідні корекції на зображенні або відхилити його, якщо воно не відповідає певним критеріям для подальшого аналізу.

Основні етапи алгоритму включають:

1. Перевірка якості зображення:

- Оцінка рівня шуму, чіткості та контрастності.
- Визначення чи зображення достатньо якісне для розпізнавання об'єктів.

2. Корекція зображення:

- Автоматична корекція освітлення, контрасту та кольорового балансу.
- Фільтрація шуму та покращення чіткості за необхідності.

3. Перевірка відповідності формату:

- Переконалися, що розміри та формат зображення відповідають встановленим вимогам.

4. Відмова або прийняття зображення:

- Якщо зображення не відповідає критеріям якості або формату, алгоритм відхиляє його з подальшою індикацією причини відмови.

– Якщо всі перевірки пройдені успішно, зображення приймається для детального аналізу та класифікації.

Цей підхід дозволяє значно підвищити ефективність та точність подальшого аналізу зображень, забезпечуючи, що система працює тільки з якісними даними. Водночас, автоматична корекція дозволяє виправляти помилки зображення та покращувати їх аналітичну цінність.

3.3. Навчання нейронної мережі

Навчання нейронної мережі — це процес адаптації ваг нейронів на основі даних для вирішення специфічної задачі. Мета навчання — мінімізувати помилку мережі, зазвичай визначену як різниця між очікуваним виходом та виходом, сгенерованим мережею.

3.3.4 Оцінка результатів навчання нейронної мережі

Визначення: Оцінка результатів навчання нейронної мережі — це процес перевірки, наскільки добре мережа виконує задану задачу після навчання. Це включає оцінювання її продуктивності, стійкості та здатності узагальнювати на нових даних.

Ключові аспекти оцінки:

1. Тестування проти невидимих даних: Оцінка зазвичай включає перевірку на даних, які не використовувалися під час навчання (тестовий набір даних), щоб оцінити здатність моделі узагальнювати.
2. Метрики оцінки: Залежно від задачі, можуть використовуватися різні метрики, такі як точність, відгук, $F1$ -оцінка для класифікації або середня квадратична помилка для регресії.
3. Перевірка стійкості: Оцінюється, наскільки добре мережа справляється зі зміною або шумом у вхідних даних.
4. Аналіз помилок: Розглядається, в яких випадках мережа зазвичай робить помилки, щоб зрозуміти її обмеження та потенційно покращити архітектуру або дані для тренування.

Методи оцінки:

1. Тестовий набір даних: Найпростіший і найпоширеніший метод — це оцінка на тестовому наборі даних, який був відокремлений від тренувального набору.
2. Перехресне затвердження: Як вже згадувалося, перехресне затвердження дозволяє більш об'єктивно оцінити здатність моделі узагальнювати, використовуючи кілька різних тренувальних і тестових розбиттів.

3. *Bootstrap* методи: Іноді використовуються для оцінки невизначеності метрик оцінки.

Практичні кроки для оцінки:

1. Вибір метрик: Визначте, які метрики найкраще відображають цілі вашої задачі.
2. Розділення даних: Забезпечте чесне розділення даних на тренувальний, валідаційний та тестовий набори.
3. Тестування та аналіз: Проведіть тестування моделі та аналізуйте результати, виявляючи слабкі сторони та можливості для покращення.
4. Ітераційне покращення: На основі аналізу, зробіть необхідні коригування в моделі або даних і повторіть оцінку.

Оцінка результатів навчання нейронної мережі є критичною частиною процесу розробки. Вона не тільки підтверджує ефективність моделі на заданій задачі, але й виявляє потенційні проблеми та допомагає удосконалити модель. Якісна оцінка вимагає чіткого розуміння задачі, вибору правильних метрик та методичного аналізу результатів. Завдяки цьому можна не тільки досягти високої продуктивності поточної моделі, але й забезпечити її здатність ефективно працювати на нових, невідомих даних.

3.3 Навчання нейронної мережі

Навчання нейронних мереж — це процес налаштування ваг нейронів з метою мінімізувати різницю між фактичним виходом мережі та очікуваним результатом. Це досягається шляхом ітеративного коригування ваг на основі даних навчання.

3.3.3 Використання методу перехресного затвердження (*Cross-Validation*)

Визначення: Перехресне затвердження — це техніка валідації моделі, яка має на меті оцінити, як статистична аналізована модель буде узагальнюватися на незалежний датасет. Вона широко використовується у задачах, де кінцевий цільовий набір даних може не бути ідентичним до набору даних, на якому відбулося навчання.

Принцип роботи:

1. Розділення даних: Найперше, весь датасет розбивається на k взаємно виключних піднаборів зазвичай зроблених однакового розміру.

2. Навчання та валідація: На кожному з k етапів, один з піднаборів використовується як тестовий набір (або валідаційний), а решта - як тренувальні дані.

3. Ітерація та агрегація: Процес повторюється k разів, кожного разу з новим тестовим піднабором. Після завершення, результати з усіх ітерацій усереднюються або іншим чином комбінуються для отримання одного оцінювання.

Переваги:

– Зменшення перенавчання (*Overfitting*): Оскільки кожен інстанс даних має шанс бути у тестовій і тренувальній вибірці, це зменшує ризик перенавчання моделі на одному наборі даних.

– Надійна оцінка: Використання різних наборів для тренування та тестування забезпечує більш об'єктивну оцінку ефективності моделі.

– Гнучкість: Метод може бути застосований до будь-якого типу моделі, і не залежить від специфіки даних.

Недоліки:

– Високі вимоги до обчислювальних ресурсів: Для кожного з k "складів" даних потрібно провести навчання та оцінку моделі, що може бути дуже ресурсомістким, особливо для великих наборів даних або складних моделей.

– Вибір k : Неправильний вибір k може призвести до перенавчання або недонавчання моделі. Зазвичай, $k=10$ використовується як стандартне значення, але для конкретних задач може знадобитися інше значення.

Застосування: Перехресне затвердження є важливим інструментом в машинному навчанні, особливо в задачах класифікації та регресії. Воно застосовується для вибору оптимальних параметрів моделі, оцінки ефективності моделей, та уникнення перенавчання.

Використання перехресного затвердження є ключовим для створення ефективних і надійних моделей нейронних мереж, що володіють здатністю

узагальнювати здобуті знання на нові, невідомі дані. З його допомогою можна значно підвищити якість та надійність прогнозів, що роблять нейронні мережі. Однак, важливо збалансувати між обчислювальними вимогами та потребами конкретної задачі для досягнення найкращих результатів.

Першим кроком є завантаження даних набору (рис. 3.3) за класами техніки.

```
Found 6672 files belonging to 8 classes.  
Using 5338 files for training.  
Found 6672 files belonging to 8 classes.  
Using 1334 files for validation.  
Found 747 files belonging to 8 classes.
```

Рис. 3.3. Завантаження даних набору техніки

3.3.1. Архітектура нейронної мережі

На основі відкритої моделі *Keras* проведено налаштування архітектури нейронної мережі для розпізнавання зображень. За документацією схожої системи розпізнавання об'єктів на зображеннях [26] було підібрано наступне налаштування (рис. 3.4). Використовуючи цю інформацію, була розроблена архітектура, що включає кілька ключових компонентів, спрямованих на підвищення точності та ефективності процесу розпізнавання:

1. Вибір базової моделі: Визначення підходящої передтренованої моделі з *Keras*, такої як *VGG16*, *ResNet*, або *Inception*, яка слугуватиме як основа для подальшого навчання та адаптації під конкретні задачі розпізнавання.
2. Модифікація шарів: Додавання, видалення або заміна певних шарів для досягнення більшої специфічності та ефективності архітектури під конкретні види зображень та об'єктів.
3. Налаштування гіперпараметрів: Тонке налаштування кількості нейронів, швидкості навчання, розміру пакетів (*batch size*), та інших гіперпараметрів для оптимізації процесу навчання.

4. Регуляризація та запобігання перенавчанню: Застосування методів, таких як *Dropout*, *Batch Normalization*, або введення штрафів за великі ваги, для покращення узагальнюючої здатності моделі.

5. Вибір функції втрат та оптимізатора: Визначення найбільш ефективної функції втрат (наприклад, крос-ентропія) та алгоритму оптимізації (наприклад, *Adam*, *SGD*), що відповідають задачі класифікації.

6. Тренування та валідація: Поділ даних на тренувальний та валідаційний набори для ефективного навчання моделі та оцінки її точності на невидимих даних.

7. Тестування та фінальна оцінка: Перевірка здатності навченої моделі розпізнавати об'єкти на нових зображеннях, аналіз помилок та оцінка загальної ефективності системи.

На рисунку 3.4 представлено схематичне відображення всієї архітектури, включаючи кожен шар, його роль та взаємозв'язки між шарами. Цей візуальний засіб допомагає зрозуміти структуру та механізми роботи моделі, а також сприяє більш ефективному налаштуванню та оптимізації архітектури.

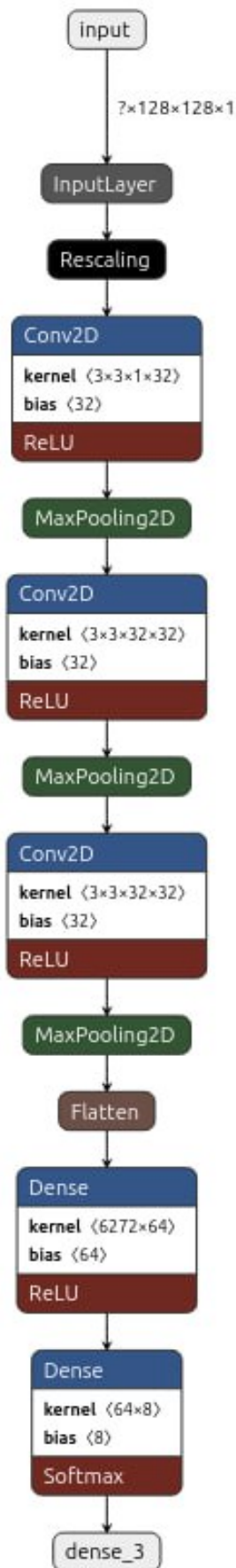


Рис. 3.4. Налаштування НМ для розпізнавання зображень

3.3.2. Підбір гіперпараметрів нейронної мережі

Для того, щоб підібрати гіперпараметри НМ, використано функцію *GridSearch* класу *Tuner* з бібліотеки *Keras*. *GridSearch* дозволяє автоматизувати процес вибору і тестування різних комбінацій гіперпараметрів, що в свою чергу сприяє вибору оптимального набору параметрів для конкретної архітектури мережі.

Процес вибору гіперпараметрів включає такі кроки:

1. Визначення діапазонів гіперпараметрів:
 - Визначення списку значень для кожного гіперпараметра, який потребує налаштування. Це можуть бути різні рівні швидкості навчання, кількості нейронів в шарі, розміру пакетів (*batch size*) тощо.
2. Використання *GridSearch*:
 - Застосування *GridSearch* для систематичного обходу всіх комбінацій гіперпараметрів. Для кожної комбінації, модель навчається та оцінюється за допомогою вибраної метрики ефективності, такої як точність або втрати.
3. Оцінка результатів:
 - Аналіз результатів, отриманих для кожної комбінації гіперпараметрів. Вибір найкращого набору на основі вибраної метрики.
4. Валідація та тестування:
 - Проведення подальших валідацій та тестувань обраної комбінації гіперпараметрів на різних даних або за допомогою крос-валідації для забезпечення стійкості та узагальнюваності моделі.
5. Тонке налаштування:
 - Подальше тонке налаштування вибраних гіперпараметрів на основі отриманих знань про їх вплив на ефективність моделі.
6. Застосування в обраній моделі:
 - Інтеграція оптимального набору гіперпараметрів в архітектуру нейронної мережі для досягнення найкращих результатів.

Використання *GridSearch* та інших методів тонкого налаштування гіперпараметрів є ключовим для оптимізації роботи нейронних мереж. Правильний вибір гіперпараметрів може значно підвищити ефективність та точність моделі, забезпечуючи краще розуміння даних і більш точні прогнози.

3.3.3. Використання методу перехресного затвердження

Для навчання мережі було використано підхід перехресного затвердження (*cross-validation*) – метод *K-Fold* бібліотеки *Sklearn* (рис. 3.5). Перехресне затвердження методом *K-Fold* є одним з найпоширеніших методів оцінки ефективності моделей машинного навчання. Використовуючи цей метод, доступний набір даних розбивається на K рівних частин, що називаються фолдами. Кожен фолд використовується раз як тестовий набір, тоді як інші $K-1$ фолди об'єднуються та використовуються як тренувальний набір. Процес повторюється K разів, кожного разу з новим фолдом як тестовим набором. Результати з усіх ітерацій потім усереднюються для отримання загальної оцінки моделі.

Процес перехресного затвердження *K-Fold* включає наступні кроки:

1. Розділення даних: Набір даних розділяється на K рівних (або приблизно рівних) частин.
2. Ітерації навчання та тестування:
 - На кожному кроці один з фолдів використовується як тестовий набір, а інші $K-1$ фолди об'єднуються для формування тренувального набору.
 - Модель навчається на тренувальному наборі та оцінюється на тестовому наборі.
 - Записується точність або інша метрика ефективності для кожного ітераційного тесту.
3. Аналіз результатів:
 - Після завершення усіх K ітерацій результати (наприклад, точність або помилка) усереднюються, щоб дати загальну оцінку ефективності моделі.
 - Можуть бути також проаналізовані дисперсія та стандартне відхилення результатів для оцінки стійкості моделі.

Використання перехресного затвердження методом *K-Fold* має кілька переваг:

– Зменшення перенавчання: Оскільки кожен елемент даних має шанс бути у тестовому наборі, це допомагає зменшити перенавчання моделі.

– Більш точна оцінка: Модель оцінюється на різних наборах даних, що забезпечує більш об'єктивну та стійку оцінку її ефективності.

– Гнучкість: Метод є універсальним і може бути використаний для оцінки різних типів моделей.

Однак, важливо відмітити, що метод *K-Fold* вимагає більше часу та обчислювальних ресурсів, оскільки модель потрібно навчати K разів. Тому, вибір кількості фолдів (K) часто є компромісом між точністю оцінки та витратами на обчислення.



Рис. 3.5. Візуалізація методу перехресного затвердження (*cross-validation*)

Таким чином для знімків аерозвідки налаштовано НМ, в якій використано гіперпараметри наступних значень (рис. 3.7).

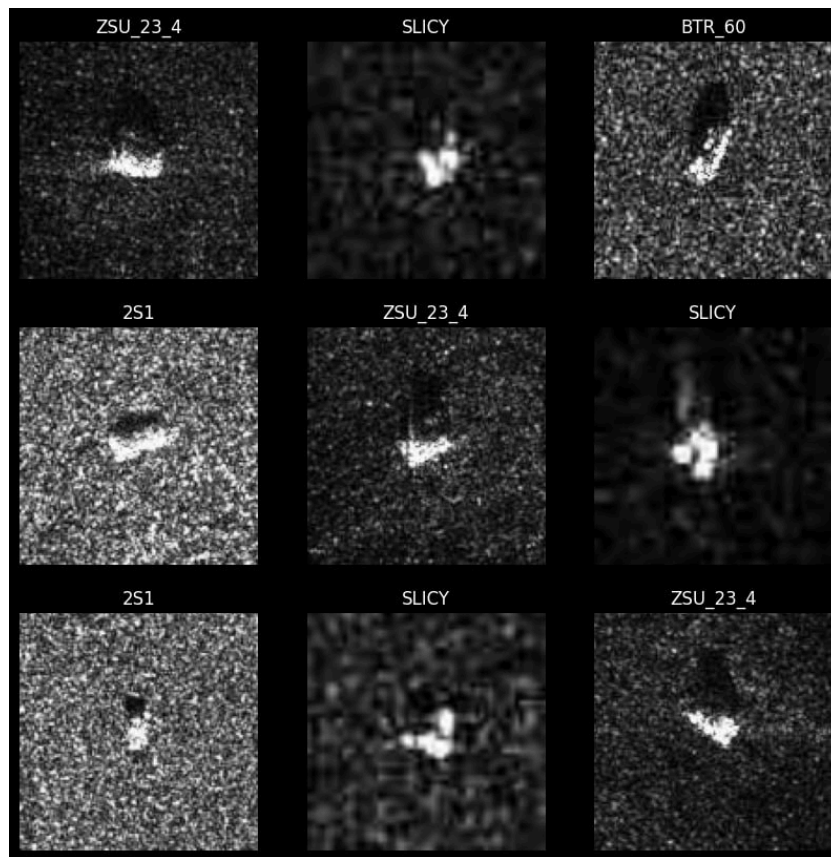


Рис. 3.6. Приклади знімків аерозвідки військової техніки

```

Model: "sequential_1"
-----
Layer (type)                Output Shape                Param #
-----
rescaling_1 (Rescaling)     (None, 128, 128, 1)        0
conv2d_3 (Conv2D)           (None, 126, 126, 32)       320
max_pooling2d_3 (MaxPooling  (None, 63, 63, 32)         0
2D)
conv2d_4 (Conv2D)           (None, 61, 61, 32)       9248
max_pooling2d_4 (MaxPooling  (None, 30, 30, 32)         0
2D)
conv2d_5 (Conv2D)           (None, 28, 28, 32)       9248
max_pooling2d_5 (MaxPooling  (None, 14, 14, 32)         0
2D)
flatten_1 (Flatten)         (None, 6272)               0
dense_2 (Dense)              (None, 64)                 401472
dense_3 (Dense)              (None, 8)                  520
-----
Total params: 420,808
Trainable params: 420,808
Non-trainable params: 0
-----

```

Рис. 3.7. Налаштування моделі НМ

3.3.4. Оцінка результатів навчання нейронної мережі

Результати навчання моделі було оцінено в наступних метриках: точність класифікації, *MCC (Matthews Correlation Coefficient)*, *F-міра (F-Measure)* та інші. Ці метрики оцінюють продуктивність моделі і демонструють її здатність до правильної класифікації військової техніки на зображеннях (рис. 3.8).

Ось більш детальний огляд кожної метрики:

1. Точність класифікації (*Accuracy*):
 - Відсоток випадків, коли модель правильно класифікує військову техніку.
 - Вона розраховується як відношення кількості правильно класифікованих зразків до загальної кількості зразків.

2. *MCC (Matthews Correlation Coefficient):*

– Оцінка якості бінарних та багатокласових класифікацій.
– *MCC* є більш інформативною метрикою, ніж проста точність, особливо для незбалансованих наборів даних, оскільки враховує істинні та хибні позитивні та негативні результати.

– Її значення лежить в діапазоні від -1 до 1, де 1 вказує на ідеальну класифікацію, 0 — на випадкову та -1 на інверсну класифікацію.

3. *F-міра (F-Measure) або F1 Score:*

– Гармонійне середнє точності та повноти (*recall*).
– Ця метрика дає більш збалансоване оцінювання, коли неважливо, який з типів помилок є критичнішим, та корисна в ситуаціях, коли нерівномірний розподіл класів.

– *F1 Score* є корисним, коли потрібно взяти до уваги як точність, так і повноту.

4. Додаткові метрики:

– Залежно від специфіки задачі та доступних даних, можуть використовуватися інші метрики, такі як *AUC-ROC* для бінарної класифікації, мульти-класова точність, чутливість, специфічність тощо.

Використання цих метрик дозволяє отримати детальне розуміння продуктивності моделі, її сильних та слабких сторін, та є важливим для коригування архітектури та параметрів моделі. Це також допомагає у визначенні, як модель може бути поліпшена та які аспекти потребують додаткової уваги при подальшому розвитку.

На рисунку 3.8 надано візуальне представлення цих метрик, ілюструючи, як кожна з них впливає на загальну оцінку моделі та як вони можуть бути використані для порівняння різних моделей або ітерацій навчання.

```

Train - 5338
Val - 1334
Epoch 1/5
167/167 [=====] - 37s 212ms/step - loss: 1.1395 - categorical_accuracy: 0.6620 - MCC: 0.6016 - F2: 0.6510 - auc: 0.9375 - prc: 0.7807
Epoch 2/5
167/167 [=====] - 35s 209ms/step - loss: 0.2913 - categorical_accuracy: 0.9110 - MCC: 0.8955 - F2: 0.9108 - auc: 0.9939 - prc: 0.9672
Epoch 3/5
167/167 [=====] - 35s 209ms/step - loss: 0.1151 - categorical_accuracy: 0.9655 - MCC: 0.9595 - F2: 0.9656 - auc: 0.9987 - prc: 0.9931
Epoch 4/5
167/167 [=====] - 35s 209ms/step - loss: 0.0587 - categorical_accuracy: 0.9826 - MCC: 0.9795 - F2: 0.9826 - auc: 0.9995 - prc: 0.9978
Epoch 5/5
167/167 [=====] - 35s 209ms/step - loss: 0.0246 - categorical_accuracy: 0.9942 - MCC: 0.9932 - F2: 0.9942 - auc: 0.9999 - prc: 0.9996
validation_accuracy for 1334 images - 0.9917541146278381
----- Fold 5 -----
Train - 5338
Val - 1334
Epoch 1/5
167/167 [=====] - 32s 182ms/step - loss: 1.1245 - categorical_accuracy: 0.6700 - MCC: 0.6106 - F2: 0.6618 - auc: 0.9399 - prc: 0.7879
Epoch 2/5
167/167 [=====] - 30s 182ms/step - loss: 0.2882 - categorical_accuracy: 0.9133 - MCC: 0.8983 - F2: 0.9130 - auc: 0.9942 - prc: 0.9691
Epoch 3/5
167/167 [=====] - 31s 184ms/step - loss: 0.0965 - categorical_accuracy: 0.9768 - MCC: 0.9728 - F2: 0.9768 - auc: 0.9990 - prc: 0.9950
Epoch 4/5
167/167 [=====] - 31s 185ms/step - loss: 0.0719 - categorical_accuracy: 0.9813 - MCC: 0.9780 - F2: 0.9813 - auc: 0.9992 - prc: 0.9970
Epoch 5/5
167/167 [=====] - 33s 199ms/step - loss: 0.0289 - categorical_accuracy: 0.9940 - MCC: 0.9930 - F2: 0.9940 - auc: 0.9999 - prc: 0.9995
validation_accuracy for 1334 images - 0.9962518811225891
Average validation accuracy over 5 folds: 0.9892104506492615

```

Рис. 3.8. Інформація про навчену нейронну мережу

Зведемо на графіку (рис. 3.9) метрики *loss* та *auc* для тестевих і валідаційних зображень.

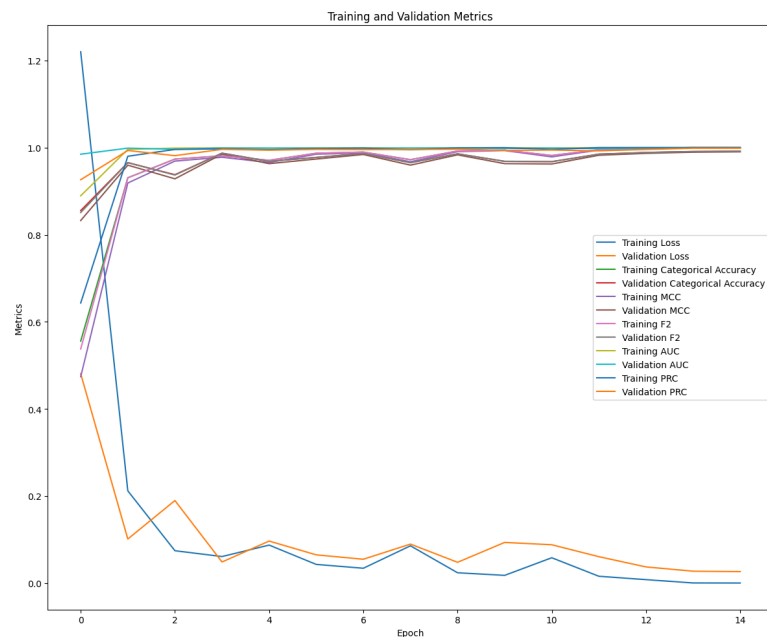


Рис. 3.9. Метрики навчання моделі: тестові і валідаційні зображення

Після навчання нейронної мережі програма звітує графіком та частот вдалої класифікації технік по основній діагоналі (рис. 3.10).

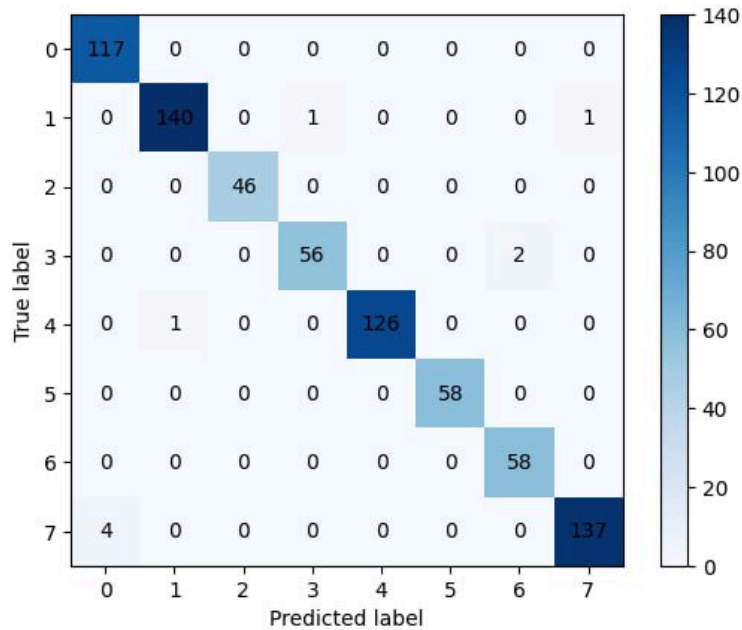


Рис. 3.10. Інформація про вірну класифікацію техніки

Після підрахунку, програма видає метрики нашої моделі на данні які ми ніколи не бачили, точність нашої моделі становить приблизно 98% з похибкою менше 10% (рис. 3.11).

```
24/24 [=====] - 1s 47ms/step - loss: 0.0541 - categorical_accuracy: 0.9880 - MCC: 0.9859 - F2: 0.9879 - auc: 0.9984 - prc: 0.9970
{'loss': 0.054056040942668915,
 'categorical_accuracy': 0.9879518151283264,
 'MCC': 0.9859123229880469,
 'F2': 0.9879298886190491,
 'auc': 0.9984369277954102,
 'prc': 0.9969944357872009}
```

Рис. 3.11. Результат сканування

Перевірялись наступні ознаки перенавчання:

- дуже висока точність на навчальних даних, але низьку точність на тестових або невідомих даних;
- велику різницю між результатами навчання та результатами валідації або тестування;
- занадто складені або "розряжені" моделі з великою кількістю параметрів, що можуть "запам'ятовувати" навчальні дані.

3.4. Описання інтерфейсу програмного модуля інтелектуального розпізнавання військової техніки за знімками аерозвідки

Інтерфейс програмного модуля інтелектуального розпізнавання військової техніки розроблений для аналізу, ідентифікації та класифікації видів військової техніки на знімках з аерозвідки. Цей модуль інтегрує алгоритми машинного навчання та штучного інтелекту для точного виявлення та класифікації об'єктів.

Основні компоненти інтерфейсу:

1. Вхідні дані:

– Зображення: Модуль приймає знімки аеророзвідки у різних форматах (наприклад, *JPEG*, *PNG*, *TIFF*).

– Параметри: Можливість вказати параметри роботи алгоритму, такі як рівень чутливості, специфічні фільтри чи режими роботи.

2. Процес обробки:

– Попередня обробка: Видалення шуму, корекція освітлення, інші трансформації для підготовки зображення.

– Детекція та сегментація: Виділення потенційних об'єктів військової техніки на знімку.

– Класифікація: Визначення типу військової техніки за допомогою нейронної мережі або інших алгоритмів машинного навчання.

3. Вихідні дані:

– Результати розпізнавання: Класифіковані об'єкти з інформацією про тип техніки, можливо, з додатковими атрибутами, як-от координати на зображенні.

– Візуалізація: Показ результатів на знімку, включаючи позначення розташування та класу військових об'єктів.

– Звіти: Можливість генерації детальних звітів про виявлені об'єкти та ефективність розпізнавання.

4. Контроль та налаштування:

– Інтерфейс користувача: Графічний користувацький інтерфейс для налаштування параметрів роботи, завантаження знімків, перегляду результатів.

– *API*: Програмний інтерфейс для інтеграції з іншими системами або для автоматизації процесів.

Основні функції:

– Швидка та точна ідентифікація: Використання сучасних алгоритмів для швидкого розпізнавання та високої точності.

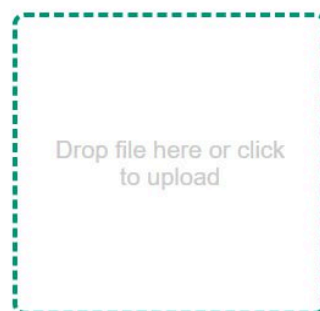
– Адаптивність: Можливість навчання та адаптації до нових типів військової техніки або змін у зображеннях.

– Багатозадачність: Підтримка одночасної роботи з кількома знімками або навіть потоковою обробкою даних.

При створенні інтерфейсу для інтелектуального розпізнавання необхідно враховувати низку викликів, включаючи забезпечення високої точності, швидкодії, а також здатності масштабування та інтеграції з іншими системами. Крім того, важливим є забезпечення безпеки та конфіденційності даних, особливо у випадку використання військових або чутливих знімків.

Інтерфейс програмного модуля інтелектуального розпізнавання військової техніки за знімками аеророзвідки є комплексним рішенням, що поєднує в собі передові технології обробки зображень, машинного навчання та штучного інтелекту. Це дозволяє ефективно вирішувати задачу ідентифікації та класифікації військової техніки, надаючи важливу інформацію для військових та розвідувальних цілей. Завдяки своїй гнучкості та розширюваності, такий модуль може бути адаптований до різноманітних умов та потреб користувачів.

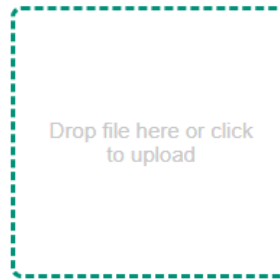
Vehicle Classification



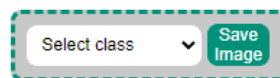
Classify

Рис. 3.12. Інтерфейс завантаження зображення

Vehicle Classification

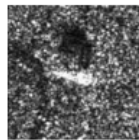


Classify



Add class

Photo:



Prediction:

Class	Probability
БРДМ BRDM_2 (СРСР)	96.547
БТР BTR_60 (СРСР)	2.77
САУ 2S1 (СРСР)	0.682

TX:

Тип: BRDM_2

Довжина: 575
Ширина: 235
Висота: 239
Вага: 7000
Макс. швидкість: 100
Тип палива: А-76
Об'єм баку: 280л

Тип: BTR_60

Довжина: 756
Ширина: 283
Висота: 223
Вага: 9900
Макс. швидкість: 80
Тип палива: Бензин
Об'єм баку: 300л

Тип: 2S1

Довжина: 726
Ширина: 285
Висота: 272
Вага: 15700
Макс. швидкість: 60
Тип палива: Дизель
Об'єм баку: 550л

Рис. 3.13. Інтерфейс виведення результату розпізнавання військової техніки на зображеннях

В результаті роботи нейронної мережі отримується класифікація зображення військової техніки на основі навчених важелів і параметрів моделі. Ця класифікація вказує на найбільш відповідну техніку з бази, до якої належить зображення, та видає коротку характеристику цієї техніки з бази даних.

Також була дана можливість додавання нового класу техніки та визначення його параметрів.

[To main page](#)

Vehicle Type:	Length:
<input type="text"/>	<input type="text"/>
Vehicle Model:	Width:
<input type="text"/>	<input type="text"/>
Vehicle Country:	Height:
<input type="text"/>	<input type="text"/>
Description:	Weight:
<input type="text"/>	<input type="text"/>
	Max Speed:
	<input type="text"/>
	Fuel Type:
	<input type="text"/>
	Fuel Max:
	<input type="text"/>

Existing classes:

- 2S1
- BRDM_2
- BTR_60
- D7
- SLICY
- ZIL131
- ZSU_23_4
- T62

Рис. 3.14. Інтерфейс додавання нового класу військової техніки

У роботі можуть бути використані різні класи військової техніки, такі як танки, бронетранспортери, військові літаки тощо. Нейронна мережа навчається розпізнавати ці класи на основі тренувального набору зображень, де кожне зображення має відповідну мітку класу.

Після навчання моделі та застосування її до нового зображення військової техніки, нейронна мережа видає прогнозований клас, який найбільше відповідає

зображенню. Наприклад, якщо вхідне зображення представляє танк, модель може класифікувати його як "танк" і надати відповідний результат.

Отримання найбільш відповідної техніки з бази є результатом процесу класифікації, де модель визначає найбільш ймовірний клас військової техніки для заданого зображення. Цей результат може бути використаний для подальшого аналізу, ідентифікації та прийняття рішень в контексті військових та розвідувальних операцій.

На рис. 3.15 представлено приклади збережених даних в таблицях (використовується *SQLite*).

vehicle id	vehicle type	vehicle model	vehicle country	description
1	САУ	2S1	СРСР	Радянська 122-мм полкова самохідна артилерійська установка
2	БРДМ	БРДМ_2	СРСР	Основна бойова броньована машина розвідувальні
3	БТР	BTR_60	СРСР	Радянський бронетранспортер (БТР) БТР-60П рс
4	Трактор	D7	США	Середній гусеничний трактор виробництва Caterp
5	Танк	T62	СРСР	T-62 - радянський середній танк, вперше предста
6	Вантажівка	ZIL131	СРСР	ЗИЛ-131 - 3,5-тонна армійська вантажівка загальн
7	ЗСУ	ZSU_23_4	СРСР	ЗСУ-23-4 "Шилка" - легкоброньована радянська са
8	Будівля	SLICY	-	SLICY ціль - це кілька простих геометричних форм

а)

vehicle params id	length	width	height	weight	max speed	fuel type	fuel max
1	726	285	272	15,700	60	Дизель	550л
2	575	235	239	7,000	100	А-76	280л
3	756	283	223	9,900	80	Бензин	300л
4	410	97	96	14,460	30	Дизель	100л
5	934	330	240	37,000	50	Дизель	960л
6	704	250	248	6,700	80	Бензин	90л
7	653	312	2,576	19,000	50	Дизель	515л
8	600	900	400	5,800	0	Немає	0л

б)

image id	image path	image date
1	pred_images/image1.png	12/06/2023 16:12:27
2	pred_images/image2.png	12/06/2023 16:15:23
3	pred_images/image3.png	12/06/2023 16:15:50
4	pred_images/image4.png	12/06/2023 16:16:05
5	pred_images/image5.png	12/06/2023 16:16:30

в)

Рис. 3.15. Приклад збережених даних: а) таблиця техніки, б) таблиця параметрів техніки, в) таблиця завантажених зображень

3.5. Висновки до розділу

У роботі використані різні класи військової техніки, такі як танки, бронетранспортери, військові літаки тощо. Нейронна мережа навчається розпізнавати ці класи на основі тренувального набору зображень, де кожне

зображення має відповідну мітку класу. Після навчання моделі та застосування її до нового зображення військової техніки, нейронна мережа видає прогнозований клас, який найбільше відповідає зображенню. Наприклад, якщо вхідне зображення представляє танк, модель може класифікувати його як "танк" і надати відповідний результат. Отримання найбільш відповідної техніки з бази є результатом процесу класифікації, де модель визначає найбільш ймовірний клас військової техніки для заданого зображення. Цей результат може бути використаний для подальшого аналізу, ідентифікації та прийняття рішень в контексті військових та розвідувальних операцій.

Покращення точності: Застосування різних технік оптимізації та підвищення точності, включаючи розширення даних, тонке налаштування гіперпараметрів, або використання ансамблів моделей.

Контекстуальний аналіз: Врахування додаткової інформації, такої як локація знімка, час доби, історичні дані про використання певних типів техніки в регіоні, що може покращити інтерпретацію результатів.

Наприклад, інтеграція з геоінформаційними системами, системами управління боєм, або іншими джерелами даних для комплексного використання інформації.

Розробка функцій для автоматичного формування звітів, аналітичних записок або рекомендацій на основі результатів класифікації.

Таким чином, нейронна мережа не лише класифікує зображення, але й слугує фундаментом для широкого спектру аналітичних і операційних застосувань, підвищуючи ефективність та оперативність військових та розвідувальних дій.

ВИСНОВКИ

У рамках даної дипломної роботи було реалізовано програмний комплекс для ідентифікації та аналізу військової техніки, заснований на зображеннях, отриманих із аерофотозйомки. Цей комплекс може застосовуватися для моніторингу та контролю військових об'єктів.

В ході розробки програмного комплексу було здійснено аналітичний огляд і створено ієрархічну структуру функцій, що відображає ключові бізнес-процеси системи. Були розроблені та впроваджені діаграми ланцюгів подій, що відображають логічну послідовність реакцій системи на зовнішні події. Діаграми *IDEF0*, які візуалізують майбутні процеси в системі, демонструють логіку роботи інтелектуального модуля видачі рішень.

Архітектура бізнесу програмного комплексу була розроблена з детальним описом внутрішніх компонентів та їх взаємодії, що дозволило чітко визначити структуру та функціональні можливості системи. Окрім того, була реалізована база даних, призначена для зберігання та обробки необхідних даних для функціонування програмного комплексу.

Розробка базувалася на принципах обробки зображень та використання алгоритмів машинного навчання, що дозволило ефективно виявляти і класифікувати різні типи військової техніки. Були визначені оптимальні структури програмного забезпечення, мови програмування та необхідні бібліотеки.

Метод перехресної валідації був застосований для навчання нейронної мережі, що забезпечило об'єктивну оцінку її ефективності. Навчання нейронної мережі було проведено з використанням різних метрик для оцінки точності, включаючи класифікаційну точність, *MCC* і *F*-міру. Результати навчання підтвердили високу ефективність моделі.

Розділ 3 описує процес розробки програмного комплексу, включаючи вибір програмних та технічних компонентів, а також детальний опис алгоритмів, використаних для аналізу та класифікації зображень. Окрім того, були

представлені можливі сценарії застосування програмного комплексу та приклади його функціонування.

У кваліфікаційній роботі було реалізовано комплексний програмний модуль для інтелектуального ідентифікування та розпізнавання військової техніки з використанням зображень аеророзвідки. Розробка модуля включала кілька ключових етапів та завдань, кожен з яких забезпечував важливу складову для функціональної та ефективної системи. Ось що було зроблено:

1. Аналіз бізнес-процесів та вимог:

Декомпозиція функцій: Було виконано ретельний аналіз функцій, які повинен виконувати програмний модуль, що включав визначення основних бізнес-процесів системи. Це дозволило розбити загальну задачу на менші, керовані частини.

Створення дерева функцій: Розроблено дерево функцій, яке ілюструвало ієрархічну структуру функціональності модуля та його взаємозв'язки.

Діаграми *Event-Driven Process Chain*: Сформовано діаграми, які відображають логіку роботи системи та її реакції на різні події.

2. Розробка та архітектура модуля:

Вибір технічних рішень: Визначено необхідні технології, мови програмування та бібліотеки. Це включало вибір алгоритмів машинного навчання та методів обробки зображень для ефективного розпізнавання.

Бізнес-архітектура: Розроблено архітектуру модуля з чітким описом внутрішніх компонентів та їх взаємодії.

База даних: Створено оптимізовану базу даних для зберігання та обробки необхідних даних.

3. Розробка алгоритмів та методів навчання:

Машинне навчання: Вибрано та налаштовано методи машинного навчання для розпізнавання військової техніки з використанням навчального набору зображень.

Перехресна валідація: Застосовано метод перехресного затвердження для оцінки та оптимізації моделі.

Оцінка моделі: Визначено ключові метрики для оцінки продуктивності моделі, включаючи точність, *MCC*, *F*-міру та інші.

4. Тестування та валідація:

Точність моделі: Проаналізовано результати навчання та визначено рівень точності моделі на тестовому наборі зображень.

Режими роботи: Розроблено різні режими роботи програмного модуля для різних сценаріїв використання.

5. Документація та результати:

Опис процесу: Детально описано процес розробки, включаючи алгоритми, методи, використані інструменти та досягнуті результати.

Приклади роботи: Надано приклади роботи програмного модуля, ілюструючи його здатність ефективно розпізнавати військову техніку.

Таким чином, було створено комплексний та функціональний програмний модуль, спроможний ідентифікувати та класифікувати різні види військової техніки, що надає цінні дані для військових та розвідувальних цілей.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойченко С.В., Іванченко О.В. Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017. – 63 с.
2. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – 39 с.
3. Коваленко Н. П., Лисенко Ю. І. та інші. Застосування нейронних мереж у задачі автоматичної класифікації зображень // Матеріали II Міжнародної науково-технічної конференції "Проблеми та перспективи розвитку сучасної техніки і технологій". – Київ, 2017. – С. 111-114.
4. Родіонов О. С., Черкаський В. Ю. Методи обробки та аналізу знімків аерозвідки для задач військового застосування / Родіонов О. С., Черкаський В. Ю. // Військова техніка, 2019. – № 1 (21). – С. 43-49.
5. Березін, А. В., Гудков, О. М., Кіпріянов, І. В. Автоматизована система розпізнавання військової техніки на зображеннях / Березін, А. В., Гудков, О. М., Кіпріянов, І. В. // Збірник наукових праць Харківського національного університету імені В. Н. Каразіна. Серія: Радіофізика та електроніка, випуск 23. – ХНУ, 2018. – С. 122-131.
6. Абрамов С.В., Лупаленко О.В., Манжай О.В. Аналіз автоматизованих систем виявлення та розпізнавання об'єктів збройних сил Російської Федерації. – *Scientific Collection «InterConf» 95. Scientific goals and purposes in XXI century.* – 2022. – С. 893–905.
7. *Mustafin R., Isakov I., Kussainov T. Recognition of military equipment on aerial photographs using convolutional neural networks.* – *International Journal of Engineering & Technology*, 7(4.6). – 2018. – P. 29-33.
8. *Ouyang W., Luo P., Zeng W., Zhong Z., Li, X. Convolutional neural networks for pedestrian detection: A survey.* – *IEEE Transactions on Circuits and Systems for Video Technology*, 28(4). – 2018. – P. 969-982.

9. *Chen L.C., Papandreou G., Kokkinos I., Murphy K., Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. – IEEE Transactions on Pattern Analysis and Machine Intelligence, 40(4). – 2018. – P. 834-848.*
10. *Li, P., Qi, Y., Zhang, S., Wang, C. Object detection based on deep learning: A review. – Journal of Sensors, 2017. – P. 1-13.*
11. Железняк О.О. Космічна фотограмметрія / Железняк О.О., Чубко Л.С. – Київ: Міністерство освіти і науки, молоді та спорту України, 2012. – 216 с.
12. Бурштинська Х. В. Аерокосмічні знімальні системи / Бурштинська Х.В., Станкевич С.А. – Львів: Львівської політехніки, 2010. – 292 с.
13. Байрак Г.Р., Муха Б.П. Дистанційні дослідження Землі: навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка. – 2010. – 712 с.
14. *Atiqur Rahman. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation / Atiqur Rahman, Wang Y. // Режим доступу: <https://www.cs.umanitoba.ca/~ywang/papers/isvc16.pdf>.*
15. *Christian Szegedy. Rethinking the Inception Architecture for Computer Vision //Режим доступу: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016.pdf.*
16. *Rayed Bin Wahed. Comparative Analysis between Inception-v3 and Other Learning Systems using Facial Expressions Detection // Режим доступу: http://dSPACE.Bracu.ac.bd/xmlui/bitstream/handle/10361/6397/12201020%20%26%2016141024_CSE.pdf.*
17. Цифрова обробка зображень [Текст] : метод, рекомендації до викон. лаборатор. робіт для студ. спеціальності 7.05080302, 8.05080302 «Аудіо-, відео- та кінотехніка» усіх форм навчання / Уклад.: В. С. Лазебний, П. В. Попович. – К.: НТУУ «КПІ», 2016. – 73 с.
18. *Kaiming He. Deep Residual Learning for Image Recognition / Xiangyu Zang, Shaoqing Ren, Jian Sun // Режим доступу: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.*

19. Nguyen H. *Fast object detection framework based on mobilenetv2 architecture and enhanced feature pyramid*. *Journal of Theoretical and Applied Information Technology*, 98 (05) – 2020. – P. 145-156.
20. Ojala T., Pietikainen M., Maenpaa T. *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*. – *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7). – 2002. – P. 971-987.
21. Simard P.Y. *A Best practices for convolutional neural networks applied to visual document analysis* / Steinkraus D., J.C. Platt // Режим доступа: <https://ieeexplore.ieee.org/document/1227801>.
22. Matusugu M., Katsuhiko M., Yusuke M., Yuji K. *Designing Convolutional Neural Network Architecture Using Genetic Algorithms*. – *International Journal of Advanced Network, Monitoring and Controls, Volume & Issue: Volume 6*. – 2021 – P. 555–559.
23. Van den Oord A., Dieleman S., Schrauwen B. *Factoring variations in natural images with deep Gaussian mixture models*. – *Advances in Neural Information Processing Systems, Vol. (4)*. – 2014. – P. 3518-3526.
24. Collobert R., Weston J. *Deep learning via semi-supervised embedding*. – *Proceedings of the 25th International Conference on Machine Learning. ICML '08*. – New York, NY, USA: ACM. – 2008. – P. 160–167.
25. *DeepLearning 0.1. LISA Lab*. Посилання: <https://github.com/lisa-lab/DeepLearningTutorials>
26. Habibi A.H., Elnaz J.H. *Guide to convolutional neural networks: a practical application to traffic-sign detection and classification*. – Cham: Springer International Publishing. – 2017. – 348 P.
27. *Military Vehicles Tracking*. Посилання: https://github.com/Lin-Sinorodin/Military_Vehicles_Tracking.
28. *Military Vehicles Image Recognition*. Посилання: <https://github.com/AlexandreSajus/Military-Vehicles-Image-Recognition>.

29. Пелех А.В. Принципи визначення військової техніки на знімках аеророзвідки / Пелех А.В., Дехтяренко Д.Т., Рябчук Н.А. // Тези доповідей міжн. наук.-техн. конф. “Інтелектуальні технології лінгвістичного аналізу” (м. Київ, 24-25 жовтня 2023 р.) – К.: НАУ, 2023. – С. 68.

30. Пелех А.В. Програмний модуль визначення військової техніки на знімках аеророзвідки / Апенько Н.В., Пелех А.В.// Тези доповідей наук.-практ. конф. “Сучасні тенденції розвитку системного програмування” (23-24 листопада 2023 р.). – К.: НАУ, 2023. – С. 48-50.

Лістинг програмного модулю розпізнавання військової техніки


```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import image_dataset_from_directory
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import *
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import tensorflow_addons as tfa
from sklearn.metrics import matthews_corrcoef, fbeta_score, roc_auc_score,
precision_recall_curve, auc
import os
import shutil
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
import keras_tuner as kt
import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.image as mpimg

directory = './mstar_images/'
# Вкажіть шлях до ваших зображень (абсолютний або відносний)
src_dir = './mstar_images/'

# Створюємо нові папки для розбиття на
# тестування/тренування/валідацію
os.makedirs('./test', exist_ok=True)
os.makedirs('./train', exist_ok=True)
```

```

class_names = os.listdir(src_dir)

for class_name in class_names:
    os.makedirs(f./test/{class_name}', exist_ok=True)
    os.makedirs(f./train/{class_name}', exist_ok=True)

    # Список всіх файлів у цій папці класу
    image_files = os.listdir(f{src_dir}/{class_name}')

    # Випадково перемішуємо файли
    np.random.shuffle(image_files)

    # Виділяємо 10% вибірки як тестову
    train_val_files, test_files = train_test_split(image_files, test_size=0.10,
random_state=42)

    # Копіюємо файли до відповідних папок
    for file in test_files:
        shutil.copy(f{src_dir}/{class_name}/{file}', f./test/{class_name}/{file}')

    for file in train_val_files:
        shutil.copy(f{src_dir}/{class_name}/{file}', f./train/{class_name}/{file}')

image_size = (128, 128)
batch_size = 32

datagen = tf.keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2,
)

```

data for GridSearch

```
train_ds = tf.keras.utils.image_dataset_from_directory('./train',  
                                                    image_size=image_size,  
                                                    label_mode='categorical',  
                                                    color_mode='grayscale',  
                                                    batch_size=batch_size,  
                                                    validation_split=0.2,  
                                                    subset='training',  
                                                    seed=10)
```

```
val_ds = tf.keras.utils.image_dataset_from_directory('./train',  
                                                    image_size=image_size,  
                                                    label_mode='categorical',  
                                                    color_mode='grayscale',  
                                                    batch_size=batch_size,  
                                                    validation_split=0.2,  
                                                    subset='validation',  
                                                    seed=10)
```

```
test_ds = tf.keras.utils.image_dataset_from_directory('./test',  
                                                    image_size=image_size,  
                                                    label_mode='categorical',  
                                                    color_mode='grayscale',  
                                                    batch_size=batch_size,  
                                                    seed=10)
```

```
params = {"ytick.color" : "w",  
          "xtick.color" : "w",  
          "axes.labelcolor" : "w",  
          "axes.edgecolor" : "w",
```

```

"figure.figsize" : (10,10),
    "axes.titlecolor" : 'w',
    "axes.facecolor" : 'w',
    "figure.facecolor" : 'k'}

colors = plt.rcParams['axes.prop_cycle'].by_key()['color']

%matplotlib inline

class_names = train_ds.class_names

with plt.rc_context(params):
    plt.figure(figsize=(10, 10))
    for images, labels in train_ds.take(1):
        for i in range(9):
            ax = plt.subplot(3, 3, i + 1)
            image = images[i].numpy().astype('uint8')
            image = image[:, :, 0]
            plt.imshow(image, cmap='gray')
            plt.title(class_names[np.argmax(labels[i])], )
            plt.axis("off")

print('Number of validation batches: %d % tf.data.experimental.cardinality(val_ds))
print('Number of test batches: %d % tf.data.experimental.cardinality(test_ds))

AUTOTUNE = tf.data.AUTOTUNE

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

```

```
num_classes = 8
```

```
metrics = [  
    keras.metrics.CategoricalAccuracy(name='categorical_accuracy'),  
    tfa.metrics.MatthewsCorrelationCoefficient(num_classes=8, name='MCC'),  
    tfa.metrics.FBetaScore(num_classes=8, average='weighted', beta=2.0,  
name='F2'),  
    keras.metrics.AUC(name='auc'),  
    keras.metrics.AUC(name='prc', curve='PR'),  
]
```

```
def holdout_results(model):  
    result = model.evaluate(test_ds)  
    return dict(zip(model.metrics_names, result))
```

```
x_train = []  
y_train = []
```

```
for images, labels in train_ds.as_numpy_iterator():  
    x_train.append(images)  
    y_train.append(labels)
```

```
x_train = np.array(x_train)  
y_train = np.array(y_train)
```

```
class CNNHyperModel(kt.HyperModel):  
    def __init__(self, input_shape, num_classes):  
        self.input_shape = input_shape  
        self.num_classes = num_classes  
    def build(self, hp):  
        model = tf.keras.Sequential()
```

```

model.add(tf.keras.layers.InputLayer(input_shape=self.input_shape))
model.add(tf.keras.layers.Rescaling(1./255))
for i in range hp.Int('conv_blocks', 3, 5, default=3)):
    filters = hp.Int('filters_' + str(i), 32, 256, step=32)
    model.add(tf.keras.layers.Conv2D(filters, kernel_size=(3,3),
activation='relu'))
    model.add(tf.keras.layers.MaxPooling2D())
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(units=hp.Int('dense_units', 32, 512,
step=32), activation='relu'))
model.add(tf.keras.layers.Dense(self.num_classes, activation='softmax'))

model.compile(optimizer=tf.keras.optimizers.Adam(hp.Float('learning_rate',
1e-4, 1e-2, sampling='log')),
loss='categorical_crossentropy', metrics=metrics)
return model

hypermodel = CNNHyperModel(input_shape=image_size + (1,),
num_classes=num_classes)

tuner = kt.GridSearch(
    hypermodel,
    objective=kt.Objective("val_categorical_accuracy", direction="max"),
    max_trials=20, # максимальна кількість спроб
    directory='./tuner_params',
    project_name='Deplomka')

tuner.search(train_ds, validation_data=val_ds)
best_hps = tuner.get_best_hyperparameters(num_trials=1)[0]
print(f"Найкращі гіперпараметри: {best_hps.values}")

```

```

# Побудова моделі з найкращими гіперпараметрами
model = tuner.hypermodel.build(best_hps)
history = model.fit(train_ds, validation_data=val_ds, epochs=15)

val_accuracy_per_fold = []
n_splits = 5
kf = KFold(n_splits=n_splits, shuffle=True, random_state=10)
num_fold = 1

train_ds = tf.keras.utils.image_dataset_from_directory('./train/',
                                                    image_size=image_size,
                                                    label_mode='categorical',
                                                    color_mode='grayscale',
                                                    batch_size=batch_size,
                                                    seed=10)

# Convert the dataset to numpy arrays for use with KFold
images, labels = [], []
for x, y in train_ds:
    images.append(x.numpy())
    labels.append(y.numpy())
images = np.concatenate(images)
labels = np.concatenate(labels)

print(f"Total images for training - {len(images)}")

from tensorflow.keras import backend as K

```