

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВОЛИНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ЛЕСІ
УКРАЇНКИ

Кафедра комп'ютерних наук та кібербезпеки

На правах рукопису

ЛАШУК МАКСИМ ТАРАСОВИЧ
РЕАЛІЗАЦІЯ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ
Спеціальність: 122 Комп'ютерні науки
Освітньо-професійна програма: Комп'ютерні науки та інформаційні
технології
Кваліфікаційна робота на здобуття освітнього ступеня «бакалавр»

Науковий керівник:
МАМЧИЧ ТЕТЯНА ІВАНІВНА,
кандидат фізико-математичних наук,
доцент кафедри комп'ютерних наук та
кібербезпеки

РЕКОМЕНДОВАНО ДО ЗАХИСТУ
Протокол No _____
засідання кафедри комп'ютерних наук
та кібербезпеки
від _____ 2024 р.
Завідувач кафедри
(_____) Гришанович Т. О.

Луцьк – 2024

ЗМІСТ

| | |
|---|-----------|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ..... | 3 |
| ВСТУП..... | 4 |
| РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ..... | 6 |
| 1.1 Комп’ютерний зір і розпізнавання образів | 6 |
| 1.2 Моделі глибокого навчання у розпізнаванні тварин | 19 |
| 1.3 Аналіз мов програмування | 32 |
| 1.4 Аналіз бібліотек для роботи з комп’ютерним зором | 33 |
| РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ | 37 |
| 2.1 Розробка програми | 37 |
| 2.2 Випробування розробленого рішення..... | 41 |
| ВИСНОВКИ | 49 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 51 |
| ДОДАТОК А. ЛІСТИНГ КОДУ ПРОГРАМИ..... | 55 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AI - Artificial Intelligence (Штучний інтелект)

ML - Machine Learning (Машинне навчання)

DL - Deep Learning (Глибинне навчання)

CNN - Convolutional Neural Network (Згорткова нейронна мережа)

RNN - Recurrent Neural Network (Рекурентна нейронна мережа)

YOLO - You Only Look Once (Модель для виявлення об'єктів)

SSD - Single Shot MultiBox Detector (Одношаровий багатоблоковий детектор)

R-CNN - Region-based Convolutional Neural Network (Згорткова нейронна мережа на основі регіонів)

Faster R-CNN - Faster Region-based Convolutional Neural Network (Швидша згорткова нейронна мережа на основі регіонів)

TP - True Positive (Істинно позитивне визначення)

FP - False Positive (Хибнопозитивне визначення)

FN - False Negative (Хибнонегативне визначення)

TN - True Negative (Істинно негативне визначення)

API - Application Programming Interface (Інтерфейс програмування додатків)

RGB - Red, Green, Blue (Червоний, зелений, синій - модель кольорів)

ROI - Region of Interest (Область інтересу)

ВСТУП

Актуальність теми. Актуальність теми дипломної роботи "розпізнавання об'єктів" є надзвичайно високою в сучасному світі завдяки швидкому розвитку технологій штучного інтелекту та машинного навчання. Розпізнавання об'єктів має широкий спектр застосувань у різних галузях, включаючи безпеку, медицину, транспорт, промисловість і розваги. Зокрема, системи розпізнавання об'єктів використовуються для автоматичного виявлення і класифікації об'єктів на зображеннях і відео, що допомагає у створенні безпечніших і ефективніших систем, таких як системи безпілотного транспорту, системи відеоспостереження, автоматизовані виробничі лінії та багато інших.

Розпізнавання тварин, як окремий аспект розпізнавання об'єктів, також має значну актуальність і потенціал. Ця технологія може бути використана для охорони навколишнього середовища, дослідження біорізноманіття, моніторингу дикої природи та збереження рідкісних видів. Наприклад, автоматизовані системи розпізнавання тварин можуть допомогти в ідентифікації та підрахунку популяцій тварин у природних резерватах, що сприяє кращому розумінню екосистем і прийняттю ефективних заходів для їх захисту. Крім того, такі технології можуть бути корисними в сільському господарстві для моніторингу здоров'я та поведінки домашніх тварин, що може підвищити ефективність фермерських господарств і поліпшити умови утримання тварин.

Об'єкт дослідження - процеси та методи автоматичного розпізнавання об'єктів.

Предмет дослідження - алгоритми та моделі глибокого навчання для розпізнавання тварин на зображеннях.

Мета дослідження - розробка та впровадження ефективної системи автоматичного розпізнавання тварин.

Наукова новизна - розробка нових методів та підходів до попередньої обробки зображень для покращення якості розпізнавання тварин.

Практична новизна - створення робочої системи автоматичного розпізнавання тварин, яка може бути інтегрована в існуючі системи моніторингу та управління.

Відповідно до мети було поставлено наступні **завдання**:

- Проаналізувати проблематику комп'ютерного зору і виявлення об'єктів в реальному часі
- Розглянути питання розпізнавання тварин та технології, які при цьому використовуються
- Проаналізувати мови програмування
- Проаналізувати бібліотеки для роботи з комп'ютерним зором
- Розробити програмне рішення
- Протестувати розроблену програму

Структура роботи. Робота складається з двох розділів, шести підрозділів, висновків та списку використаних джерел. Загальний обсяг роботи – 70 сторінок.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Комп'ютерний зір і розпізнавання образів

Сфера виявлення об'єктів у реальному часі стала незамінним аспектом комп'ютерного зору, охоплюючи швидку та точну ідентифікацію об'єктів у зображеннях і відеопотоках. Ця можливість знаходить широке застосування в різноманітних програмах, включаючи автономні транспортні засоби, робототехніку, спостереження та доповнену реальність. Здатність швидко розрізняти й точно визначати місцезнаходження об'єктів у потоках візуальних даних має потенціал для революції в процесах прийняття рішень і підвищення ефективності найсучасніших технологій. Виявлення об'єктів у реальному часі стало свідком трансформаційного зрушення з появою глибокого навчання, зокрема з появою згорткових нейронних мереж (CNN). Ці нейронні мережі зробили революцію в задачах комп'ютерного зору, автономно виділяючи складні представлення функцій з необроблених піксельних даних, дозволяючи розпізнавати складні візерунки та функції, необхідні для надійного розпізнавання об'єктів.

На перший план вийшов набір моделей виявлення об'єктів на основі глибокого навчання, кожна з яких відрізняється своєю унікальною архітектурою та сильними сторонами. Видатні приклади включають YOLO (You Only Look Once), SSD (Single Shot Multibox Detector) і Faster R-CNN (Region-based Convolutional Neural Networks). Ці моделі використовують різноманітні стратегії для досягнення балансу між точністю та швидкістю, задовольняючи динамічні вимоги додатків у реальному часі. Процес виявлення об'єктів у режимі реального часу включає ряд етапів, таких як збір даних, анотацію, навчання моделі та висновок. Моделі глибокого навчання навчаються на великих наборах даних, що містять позначені екземпляри об'єктів, що дозволяє їм набути здатності точно виявляти об'єкти. Під

час висновку ці моделі оперативно аналізують прямі відеопотоки або послідовності зображень, генеруючи обмежувальні прямокутники навколо ідентифікованих об'єктів разом із відповідними мітками класів. Прагнення до можливостей реального часу вимагає впровадження методів оптимізації. Такі стратегії, як квантування моделі, яке зменшує розмір моделі, зберігаючи продуктивність, у поєднанні з використанням апаратного прискорення через GPU або TPU (блоки обробки тензорів) разом сприяють підвищенню швидкості обробки та загальної ефективності систем виявлення об'єктів у реальному часі.

Хоч і було досягнуто значного прогресу, виявлення об'єктів у реальному часі залишається активною областю досліджень. Дослідники наполегливо опрацьовують нові алгоритми, архітектури та вдосконалюють апаратне забезпечення, щоб підвищити точність, ефективність і адаптивність моделі, створюючи таким чином більш надійні та практичні системи виявлення об'єктів у реальному часі, здатні ефективно працювати в реальних сценаріях [1].

Виявлення об'єктів у реальному часі ґрунтується на фундаментальному завданні ідентифікації та локалізації об'єктів у зображеннях або відеопотоках. Мета полягає в тому, щоб класифікувати об'єкти, окреслюючи їх точне положення за допомогою обмежувальних рамок. Траєкторія виявлення об'єктів зазнала суттєвої еволюції, спричиненої появою методів глибокого навчання та доступністю ретельно анотованих наборів даних. Центральними для цих досягнень є згорткові нейронні мережі (CNN), які автономно розрізняють ієрархічні характеристики даних. Такі моделі, як YOLO і Faster R-CNN, служать зразками надзвичайної точності та продуктивності в режимі реального часу, які визначали виявлення об'єктів у різноманітних програмах.

Мотивація виявлення об'єктів у реальному часі впливає зі зростаючої потреби в ефективних і точних системах візуального розуміння в різних сферах реального світу. Традиційні методи виявлення об'єктів, незважаючи на ефективність, часто не справлялися з проблемами обробки в реальному часі, що

має вирішальне значення в динамічних середовищах, де своєчасна відповідь є важливою. Поява підходів на основі глибокого навчання, таких як YOLO і SSD, забезпечила вирішення цієї проблеми, спричинивши значний прогрес у можливостях виявлення об'єктів у реальному часі

Виявлення об'єктів є фундаментальним завданням у комп'ютерному зорі, яке передбачає ідентифікацію та локалізацію об'єктів інтересу в зображеннях або відеопотоках. Мета полягає в тому, щоб не тільки класифікувати об'єкти за попередньо визначеними категоріями, але й намалювати навколо них обмежувальні прямокутники, вказуючи їх точне розташування. Протягом багатьох років було досягнуто значних успіхів у виявленні об'єктів завдяки появі методів глибокого навчання та наявності великих анотованих наборів даних. Згорткові нейронні мережі (CNN) відіграли ключову роль у революції у виявленні об'єктів шляхом автоматичного вивчення ієрархічних характеристик із даних. Кілька найсучасніших моделей, таких як YOLO і Faster R-CNN, продемонстрували вражаючу точність і продуктивність у реальному часі. Ці моделі знайшли застосування в різних сферах, включаючи автономні транспортні засоби, системи спостереження, медичну візуалізацію тощо, демонструючи важливість виявлення об'єктів у створенні широкого спектру практичних та інноваційних рішень.

Виявлення об'єктів — це фундаментальне завдання комп'ютерного зору, яке передбачає ідентифікацію та локалізацію конкретних об'єктів, що представляють інтерес, у зображеннях або відеокадрах. Основна мета — виявити наявність об'єктів і намалювати навколо них обмежувальні рамки, вказавши їх точне розташування та протяжність. Крім того, виявлення об'єктів часто включає класифікацію виявлених об'єктів за попередньо визначеними категоріями або класами, що забезпечує повне розуміння сцени.

Важливість виявлення об'єктів:

Виявлення об'єктів відіграє вирішальну роль у різних прикладних програмах реального світу та стало основним компонентом сучасних систем комп'ютерного зору.

Виявлення об'єктів дозволяє машинам сприймати та розуміти зміст зображень або відеопотоків. Ідентифікуючи та локалізуючи об'єкти, системи отримують глибше розуміння візуальних даних, сприяючи розширеному аналізу та ухваленню рішень.

У таких сферах, як автономні транспортні засоби та робототехніка, виявлення об'єктів має важливе значення для виявлення пішоходів, транспортних засобів, перешкод та інших відповідних об'єктів у навколишньому середовищі. Ця інформація має вирішальне значення для забезпечення безпечної навігації та взаємодії з оточенням.

Виявлення об'єктів є життєво важливим у системах спостереження для виявлення потенційних загроз, зловмисників або підозрілих дій. Виявлення об'єктів у режимі реального часу дозволяє негайно реагувати на порушення безпеки, покращуючи заходи безпеки.

У медичній візуалізації виявлення об'єктів використовується для ідентифікації анатомічних структур, уражень і аномалій. Це допомагає в діагностиці та плануванні лікування, сприяючи покращенню результатів лікування.

Виявлення об'єктів використовується в додатках доповненої реальності для взаємодії та накладання віртуальних об'єктів на реальний світ. Це забезпечує повну інтеграцію віртуального та фізичного середовищ.

Виявлення об'єктів використовується для розпізнавання жестів і відстеження поз людини, що забезпечує більш інтуїтивну та природну взаємодію з комп'ютерами та пристроями.

Виявлення об'єктів полегшує розпізнавання та локалізацію продукту, що робить його цінним у програмах, таких як автоматизовані системи оформлення замовлення та керування запасами.

Виявлення об'єктів можна використовувати для моніторингу дикої природи, ідентифікації видів рослин і відстеження змін у природному середовищі, сприяючи зусиллям по збереженню природи та екологічним дослідженням.

Загалом виявлення об'єктів має величезне значення для того, щоб машини могли розуміти візуальні дані та ефективно взаємодіяти з реальним світом. Його універсальність і широке застосування роблять його основним інструментом у різних галузях промисловості та наукових областях, сприяючи прогресу в технологіях і покращуючи наше повсякденне життя

Проблеми, які виникають під час виявлення об'єктів у реальному часі:

Ландшафт виявлення об'єктів у реальному часі представляє низку проблем, що кореняться в імперативі швидкої та точної обробки візуальних даних. Ці проблеми випливають із складності реальних сцен, потреби в продуктивності в реальному часі та делікатної рівноваги між швидкістю та точністю в алгоритмах виявлення об'єктів.

Основні проблеми:

Обчислювальна складність: моделі виявлення об'єктів на основі глибокого навчання, прикладом яких є YOLO та SSD, висувають значні обчислювальні вимоги через свою складну архітектуру та конфігурацію параметрів. Досягнення продуктивності в реальному часі на платформах з обмеженими обчислювальними ресурсами вимагає ретельної оптимізації моделі та використання методів апаратного прискорення.

Компроміс між швидкістю та точністю: системи виявлення об'єктів у реальному часі часто стикаються з компромісом між швидкістю та точністю. Прискорена обробка може призвести до спрощення моделі або зниження просторової роздільної здатності, що потенційно впливає на точність виявлення.

Встановлення оптимальної рівноваги між швидкістю та точністю є обов'язковим для виконання вимог у режимі реального часу при збереженні прийнятної продуктивності виявлення.

Багатомасштабне виявлення об'єктів: сцени реального світу містять об'єкти різного масштабу, що вимагає одночасного виявлення об'єктів різного розміру. Ефективне звернення до багатомасштабних об'єктів має важливе значення для повного розуміння сцени.

Оклюдія та безлад: ефективному виявленню об'єктів заважають закриті об'єкти та захарашений фон. Надійні алгоритми необхідні для виявлення частково видимих об'єктів і керування примірниками з характеристиками, що перекриваються [3].

Адаптація до динамічного середовища: сценарії реального світу за своєю суттю є динамічними, вимагаючи від систем виявлення об'єктів у реальному часі для швидкої адаптації до змін навколишнього середовища, змін умов освітлення та рухомих об'єктів для підтримки точності та надійності.

Виявлення малих об'єктів: виявлення мініатюрних об'єктів, особливо тих, що розташовані на відстані або мають низьку роздільну здатність є проблемою. Моделі виявлення об'єктів у режимі реального часу повинні бути чутливими до малих об'єктів без шкоди для загальної продуктивності.

Анотовані дані та маркування: підготовка великих анотованих наборів даних для виявлення об'єктів у реальному часі може бути трудомісткою. Наявність точних анотацій, що охоплюють різні класи об'єктів, є ключовою для ефективного навчання моделі.

Вирішення цих проблем потребує постійних досліджень та інновацій, що охоплює розробку складних алгоритмів, стратегій оптимізації та підтримку апаратних компонентів. Системи виявлення об'єктів у реальному часі, здатні швидко та точно аналізувати візуальні дані, мають потенціал для революції в

додатках, що охоплюють галузі, сприяючи інтелектуальній та безпечній взаємодії між машинами та світом.

Застосування виявлення об'єктів у реальному часі:

Автономні транспортні засоби: виявлення об'єктів у реальному часі є основоположним елементом систем автономного водіння, що дозволяє автомобілям сприймати пішоходів, транспортні засоби та перешкоди та реагувати на них. Технологія відіграє ключову роль в уникненні зіткнень, відстеженні смуги руху та загальному розумінні ситуації.

Спостереження та безпека: виявлення об'єктів у реальному часі полегшує моніторинг у реальному часі та оцінку загроз у системах спостереження. Це дозволяє ідентифікувати підозрілу поведінку, багаж без нагляду або несанкціонований доступ, сприяючи підвищеним заходам безпеки.

Робототехніка: роботи, наділені можливостями виявлення об'єктів у режимі реального часу, орієнтуються в середовищі, маніпулюють об'єктами та взаємодіють з навколишнім середовищем. Це покращує співпрацю людини та робота та розширює автономність роботозованих систем.

Доповнена реальність: виявлення об'єктів у реальному часі покращує досвід доповненої реальності шляхом накладання цифрового вмісту на об'єкти реального світу. Підтримуються такі програми, як розпізнавання об'єктів, інтерактивні ігри та захоплююча візуалізація.

Медична візуалізація: виявлення об'єктів у реальному часі знаходить застосування в медичній візуалізації, допомагаючи радіологам ідентифікувати та локалізувати анатомічні структури та аномалії. Це прискорює діагностику та планування лікування, сприяючи покращенню догляду за пацієнтами. Оскільки штучний інтелект та комп'ютерний зір продовжують розвиватися, системи виявлення об'єктів у реальному часі готові змінити ще безліч галузей, сприяючи безпечнішій та ефективнішій взаємодії між людьми та машинами.

Виявлення об'єктів у реальному часі є наріжним каменем сучасного комп'ютерного зору, що дозволяє швидко й точно ідентифікувати об'єкти в потоках візуальних даних. Поєднання алгоритмів глибокого навчання, апаратного прискорення та стратегій оптимізації стало каталізатором розробки систем виявлення об'єктів у реальному часі. Незважаючи на видатні успіхи, проблеми залишаються, включаючи обчислювальну складність, багатомасштабну обробку об'єктів і делікатні компроміси між точністю і швидкістю. Дослідники та практики повинні співпрацювати у вирішенні цих проблем, культивуючи інноваційні рішення та надійні системи, здатні процвітати у складних та динамічних сценаріях реального світу. Траєкторія виявлення об'єктів у реальному часі продовжує розгортатися і майбутні дослідження готові дати нові алгоритми, архітектури та вдосконалення апаратного забезпечення.

Традиційні підходи до виявлення об'єктів

До появи підходів, заснованих на глибокому навчанні, традиційні методи виявлення об'єктів спиралися на створені вручну функції та спеціальні алгоритми [7].

Деякі з відомих традиційних підходів до виявлення об'єктів:

Гістограма орієнтованих градієнтів (HOG): HOG — це дескриптор функції, який використовується для представлення інформації про локальну текстуру та форму зображення. Він фіксує інформацію про орієнтацію градієнта та обчислює гістограми напрямків градієнта для виявлення країв і меж об'єктів. HOG широко використовується для виявлення пішоходів та інших завдань виявлення об'єктів.

Хаар-функції : Хаар-подібні функції — це прості прямокутні фільтри, які використовуються в алгоритмі Віоли-Джонса для виявлення об'єктів. Ці функції вловлюють варіації інтенсивності в певних областях зображення та є обчислювально ефективними для додатків у реальному часі. Алгоритм Віоли-Джонса відомий своїми можливостями швидкого виявлення облич. .

Відповідність функцій: методи зіставлення функцій, такі як Scale-Invariant Feature Transform (SIFT) і Speeded-Up Robust Features (SURF), виявляють характерні локальні особливості на зображенні та зіставляють їх між кадрами для розпізнавання та відстеження об'єктів. Ці методи використовуються для виявлення об'єктів і вирівнювання зображення. .

Моделі деформованих частин (DPM): DPM — це класичний фреймворк для виявлення об'єктів, який представляє об'єкти як набір деформованих частин. Він моделює просторове співвідношення між частинами та фіксує варіації зовнішнього вигляду об'єкта для підвищення точності виявлення. DPM використовувався для виявлення об'єктів з шарнірними конструкціями. .

Вибірковий пошук: вибірковий пошук — це метод генерації пропозицій, який використовується під час виявлення об'єктів для створення регіонів-кандидатів, які ймовірно містять об'єкти. Він сегментує зображення на основі кольору, текстури та розміру, щоб отримати потенційні області об'єкта для подальшої обробки [9].

Хоча традиційні підходи до виявлення об'єктів були ефективними в певних сценаріях, методи глибокого навчання, такі як YOLO та SSD, перевершили їх з точки зору точності та ефективності, особливо в задачах виявлення об'єктів у реальному часі.

Методи на основі ковзного вікна

Методи на основі ковзних вікон були одними з ранніх традиційних підходів до виявлення об'єктів. Ці методи включають переміщення вікна фіксованого розміру по зображенню в різних масштабах для виявлення об'єктів у різних місцях і розмірах. Хоча підходи ковзного вікна були значною мірою замінені методами глибокого навчання, вони дають цінну інформацію про еволюцію цих методів.

Ось кілька методів ковзного вікна:

1. Гістограми орієнтованих градієнтів для виявлення людини. Було представлено дескриптор функцій, який став ключовим компонентом багатьох

детекторів об'єктів на основі ковзного вікна. Дескриптор HOG фіксує інформацію про орієнтацію локальних градієнтів для представлення країв об'єктів і широко використовується для виявлення пішоходів [10].

2. Виявлення об'єктів за допомогою дискримінаційно навчених моделей на основі частин. Ця робота представила структуру моделей деформованих частин (DPM) для виявлення об'єктів. DPM використовує підхід ковзного вікна для пошуку частин об'єкта, моделюючи просторові відносини між частинами для кращої точності виявлення .

3. Відмінні риси зображення від масштабно-інваріантних ключових моментів. Масштабно-інваріантне перетворення ознак (SIFT), широко використовується для завдань зіставлення ознак і розпізнавання об'єктів. Ковзні вікна часто використовуються в методах на основі SIFT для виявлення ключових точок і виконання зіставлення ознак у різних масштабах зображення .

4. Швидке виявлення об'єктів за допомогою розширеного каскаду простих функцій. Алгоритм Віоли-Джонса, який є одним із найперших і успішних методів виявлення об'єктів у реальному часі, заснованих на функціях, подібних до Хаара. Техніка ковзного вікна використовується в алгоритмі Віоли-Джонса для сканування всього зображення на потенційні місця розташування об'єктів .

Методи на основі ковзних вікон були обмежені їхньою обчислювальною складністю, оскільки вони включали вичерпну оцінку ковзних вікон у кількох масштабах, що призводило до тривалого обчислення.

Розробка підходів на основі глибокого навчання, таких як YOLO та SSD, значно підвищила швидкість і точність виявлення об'єктів шляхом впровадження наскрізного навчання та нових архітектур. Ці сучасні методи значною мірою замінили підходи на основі ковзних вікон у практичних застосуваннях, оскільки вони досягають продуктивності в реальному часі без необхідності явного сканування вікон.

Підходи на основі функцій

Підходи, засновані на функціях відіграли значну роль у ранньому розвитку методів виявлення об'єктів і були основоположними в історії комп'ютерного зору. Ці методи спиралися на ручне виділення ознак і спеціалізовані алгоритми для ідентифікації об'єктів на зображеннях. Хоча їх значною мірою перевершили методи, засновані на глибокому навчанні, підходи, засновані на функціях, проклали шлях до більш просунутих методів.

Ключові методи, засновані на функціях:

Лоу. Д. Г. Відмінні риси зображення від масштабно-інваріантних ключових точок: масштабно-інваріантне перетворення ознак (SIFT), стало одним із найбільш широко використовуваних дескрипторів ознак. Він важливий для задач розпізнавання об'єктів, зіставлення зображень і виявлення об'єктів завдяки своїй здатності отримувати незмінні від масштабу ключові точки та дескриптори.

Бухта . SURF : прискорені надійні функції. SURF — ще один впливовий метод, заснований на функціях, відомий тим, що забезпечує надійні та ефективні дескриптори локальних функцій. Він використовує наближення матриці Гессе для виділення ключових точок і зазвичай використовується в задачах розпізнавання об'єктів і зіставлення зображень.

Dalal. N. Гістограми орієнтованих градієнтів для виявлення людей. Дескриптор функції гістограми орієнтованих градієнтів (HOG), допомагає у задачах виявлення пішоходів і об'єктів. HOG фіксує локальну інформацію про орієнтацію градієнта, що робить її придатною для визначення країв і меж об'єктів.

Felzenszwalb, P. F. Виявлення об'єктів за допомогою дискримінаційно навчених моделей на основі частин. Структура моделей деформованих частин (DPM), є підходом на основі функцій, який представляє об'єкти як набір деформованих частин. Він фіксує просторові співвідношення між частинами для підвищення точності виявлення об'єктів.

Віола. П.. Швидке виявлення об'єктів за допомогою розширеного каскаду простих функцій. Алгоритм Віоли-Джонса є одним із найперших і успішних

методів виявлення об'єктів у реальному часі, заснованих на функціях, подібних до Хаара. Він ефективно виявляє об'єкти, вибираючи підмножину функцій, подібних до Хаара, за допомогою AdaBoost.

Ці методи, засновані на функціях дали цінну інформацію та заклали основу для дослідження виявлення об'єктів. Однак у них були обмеження, такі як потреба в функціях ручної роботи та великих обчислювальних ресурсах. Розвиток підходів до глибокого навчання та наскрізного навчання призвів до суттєвих покращень у точності та ефективності виявлення об'єктів, завдяки чому методи, засновані на функціях, рідше використовуються в сучасних програмах. Перехід до методів, заснованих на глибокому навчанні, дозволив автоматично вивчати функції, зменшуючи залежність від створених вручну функцій і створюючи більш складні та точні системи виявлення об'єктів [11].

Складнощі у виявленні об'єктів у режимі реального часу

Досягнення функціональності в реальному часі потребує швидкої обробки кадрів або зображень, що створює складність через обчислювальну інтенсивність методологій глибокого навчання, таких як Faster R-CNN і YOLO. Щоб вирішити цю проблему, дослідники розробили полегшені архітектури, такі як SSD і YOLOv3-tiny, що знижує точність для швидшої обробки. Збільшення швидкості обчислень можливе за допомогою механізмів апаратного прискорення, таких як GPU або TPU.

Підтримка точності виявлення має вирішальне значення, але прискорена обробка може призвести до зниження точності порівняно з повільнішими, але більш точними методами. Щоб пом'якшити зниження точності, можна використовувати розширені архітектури, складні магістралі, як-от ResNet, і оптимізацію гіперпараметрів під час навчання.

Розглядання різноманітності розміру об'єкта та співвідношення сторін у сценах реального світу потребує таких методів, як мережа пірамід функцій (FPN) і

блоки прив'язки. FPN фіксує багатомасштабні об'єкти, тоді як блоки прив'язки полегшують прогнозування для об'єктів різного розміру .

Часткова оклюзія та безлад у реальних сценах ускладнюють виявлення. Можна розробити стійкі моделі виявлення об'єктів, які обробляють перекриття та перешкоди, використовуючи контекстну інформацію або часову узгодженість між кадрами для підвищення точності.

Обмеження ресурсів у периферійних пристроях або вбудованих системах можна вирішити за допомогою спрощених архітектур і методів квантування моделей.

Навчання моделей виявлення об'єктів у реальному часі вимагає значної кількості анотованих даних. Ефективність можна підвищити завдяки передачі навчання та доповненню даних, використовуючи попередньо навчені моделі та синтетичні дані для зменшення вимог до анотацій.

Адаптація моделей з одного середовища до різних з різними умовами вимагає асиміляції різноманітних навчальних даних і застосування методів адаптації домену для адаптації.

Конвергенція вдосконалених алгоритмів, оптимізації апаратного забезпечення та підібраних наборів даних має ключове значення для подолання цих багатогранних викликів. Вчені та практики постійно досліджують інноваційні методи вдосконалення виявлення об'єктів у режимі реального часу для додатків, що охоплюють робототехніку, спостереження, автономні транспортні засоби тощо.

Висновки та майбутні напрямки та виклики

Виявлення об'єктів у реальному часі є наріжною технологією в сферах комп'ютерного зору та штучного інтелекту, наділяючи машини здатністю миттєво сприймати та розуміти оточення. Об'єднання передових алгоритмів глибокого навчання, спрощена архітектура та механізми апаратного прискорення породили парадигматичну трансформацію, катапультируючи виявлення об'єктів у реальному

часі в різноманітні програми, що охоплюють автономні транспортні засоби, спостереження, робототехніку тощо.

Еволюція виявлення об'єктів у реальному часі супроводжувалася низкою проблем, кожна з яких зустрічалася з інноваційними рішеннями. Від появи одноетапних архітектур, таких як YOLO та SSD, до оркестровки периферійних обчислювальних парадигм і апаратних прискорювачів, шлях виявлення об'єктів у реальному часі характеризується постійним удосконаленням і прогресом [16].

Майбутні напрямки включають виявлення 3D-об'єктів у реальному часі, ефективні апаратні архітектури, мультимодальне злиття та поступове навчання. Проблеми охоплюють обробку складних сцен, обмеження ресурсів і зміщення даних, що підкреслює необхідність постійних досліджень і розробок для підвищення точності, ефективності та надійності виявлення об'єктів у реальному часі в програмах і доменах, що розвиваються.

1.2 Моделі глибокого навчання у розпізнаванні тварин

Традиційні методи виявлення тварин у відеоданих покладаються на методи комп'ютерного зору та алгоритми обробки зображень. Ці підходи включають вилучення ознак з окремих кадрів або послідовностей кадрів, а потім використання класифікаторів для ідентифікації присутності тварин. Це включає покадровий аналіз, віднімання фону, відстеження об'єктів і застосування класифікатора. Однак ці методи стикаються з кількома проблемами. Їм часто потрібна допомога, щоб вловити часовий контекст, що ускладнює відмінність тварин від схожих на вигляд об'єктів. Вони можуть бути чутливими до змін умов освітлення та складного фону, що призводить до помилкових спрацьовувань або пропусків виявлення. Крім того, широка мінливість зовнішнього вигляду тварин створює проблему для створення універсального набору ознак. Крім того, через

свої обчислювальні вимоги традиційні методи можуть погано масштабуватися при застосуванні до великомасштабних наборів відеоданих.

Глибоке навчання, зокрема згорткові нейронні мережі (CNN), революціонізувало виявлення тварин у відеоданих. Ці моделі можуть вивчати ієрархічні функції безпосередньо з необроблених піксельних даних, усуваючи потребу в розробці функцій вручну. Це дозволяє мережі адаптуватися до широкого діапазону зовнішнього вигляду тварин. Крім того, повторювані нейронні мережі (RNN) і тривимірні згорткові нейронні мережі (3D CNN) дозволяють фіксувати тимчасові залежності, що веде до кращого розуміння моделей руху в часі. Трансферне навчання, яке передбачає тонке налаштування попередньо навчених моделей на великих наборах даних, таких як ImageNet, використовує знання з різноманітних наборів даних для конкретних завдань виявлення тварин. Моделі глибокого навчання також вправно відрізняють тварин від складного фону шляхом автоматичного вилучення відповідних ознак. У рамках глибокого навчання та апаратному забезпеченні ефективна обробка великих обсягів відеоданих стала можливою, що ще більше покращує масштабованість завдань виявлення тварин. Загалом, методи глибокого навчання, особливо CNN, значно підвищили точність і ефективність виявлення тварин у відеоданих, усунувши обмеження традиційних методів.

Виявлення тварин — це важливий додаток комп'ютерного зору та глибокого навчання, метою якого є автоматичне визначення присутності тварин у відео. Протягом багатьох років для вирішення цього завдання було розроблено різні архітектури глибокого навчання, кожна з яких має сильні та обмежені сторони. Для виявлення тварин у відеоданих було використано кілька архітектур глибокого навчання. Мета полягає в тому, щоб розпізнати тварин на відео, щоб розрізнити типову поведінку від незвичної.

Нейронні мережі прямого зв'язку (FNN) представляють основоположну парадигму штучних нейронних мереж. Її відмітна характеристика полягає в

односпрямованому потоці інформації, що починається на вхідних вузлах, проходить через низку прихованих шарів і завершується на вихідних вузлах. Ця систематична прогресія даних дозволяє FNN досягати успіхів у завданнях, що охоплюють регресію, де мета полягає в прогнозуванні безперервних числових значень і класифікації, яка передбачає призначення дискретних міток вхідним даним.

Вони отримали широке визнання за свою адаптивність і універсальність, що робить їх незамінними інструментами в машинному навчанні. Їхні знання поширюються на різні сфери, від комп'ютерного зору та обробки природної мови до фінансів та охорони здоров'я. Цей архітектурний проект є ключовим каменем для розробки більш складних моделей нейронних мереж. Він залишається центром безперервного дослідження та вдосконалення в динамічному ландшафті досліджень і застосування глибокого навчання [17].

Згорткові нейронні мережі (CNN) — це спеціалізовані архітектури, призначені для обробки сітчастих даних, які особливо добре підходять для завдань із зображеннями. Ці мережі відрізняються використанням згорткових шарів, що дозволяє їм автономно отримувати ієрархічні характеристики з вхідних даних. Ця можливість вивчення ієрархічних функцій є перевагою, коли властива структура даних містить критичну інформацію.

CNN продемонстрували надзвичайну ефективність у різних програмах, особливо в обробці зображень і комп'ютерному зорі. Вони відмінно справляються із завданнями, починаючи від класифікації зображень, де метою є призначення мітки вхідному зображенню, до виявлення об'єктів, що передбачає точне визначення розташування об'єктів в ідеї, та сегментації, яка передбачає поділ зображення на окремі області чи об'єкти. Адаптивність і майстерність CNN зміцнили їхню позицію як незамінного інструменту глибокого навчання з широким застосуванням у таких сферах, як медична візуалізація, автономні транспортні

засоби тощо. Їх унікальний архітектурний дизайн продовжує залишатися центром інновацій та удосконалення в еволюції моделей нейронних мереж.

Повторювані нейронні мережі (RNN) — це спеціалізована архітектура, яка точно налаштована на нюанси послідовних даних, охоплюючи такі області, як аналіз часових рядів і обробка природної мови. Що відрізняє RNN від інших, то це їх властива здатність зберігати внутрішню пам'ять, що дозволяє їм ефективно обробляти послідовності, зберігаючи контекст із минулих вхідних даних. Ця повторювана структура наділяє їх динамічною адаптивністю, яка особливо добре підходить для завдань, де порядок і взаємозв'язок елементів у послідовності є вирішальними.

RNN знайшли надзвичайний успіх у різних програмах, зокрема в завданнях, таких як моделювання мови, де метою є передбачення ймовірності заданої послідовності слів, і машинний переклад, який передбачає перетворення тексту з однієї мови на іншу. Крім того, RNN незамінні в аналізі часових рядів, де важливо розуміти часові закономірності та робити прогнози на основі історичних даних.

Універсальний характер RNN позиціонує їх як наріжний камінь у глибокому навчанні, із застосуваннями, що виходять за межі аналізу мови та часових рядів у такі сфери, як розпізнавання мови, аналіз настроїв тощо. Їх відмінна архітектурна структура є центром інновацій і постійних досліджень, що сприяє прогресу в послідовній обробці даних.

Мережі розширеної короткочасної пам'яті (LSTM) представляють удосконалену ітерацію рекурентних нейронних мереж (RNN), створених для пом'якшення проблеми зникнення градієнта. Це вдосконалення дозволяє LSTM досягати успіхів у захопленні складних довгострокових залежностей у послідовних даних, що є важливою можливістю в завданнях, де розуміння контексту має першочергове значення. Їхня архітектура містить спеціалізовані механізми, які сприяють збереженню інформації протягом тривалих періодів, що

робить їх потужним інструментом для таких завдань, як обробка природної мови та прогнозування часових рядів.

Gated Recurrent Units (GRU) є ще одним варіантом RNN, схожим на LSTM, у своїй здатності керувати послідовними даними. Їх обчислювальна ефективність відрізняє GRU, пропонуючи більш спрощений підхід до послідовностей обробки, але збереження аналогічного рівня ефективності. Така ефективність досягається шляхом інтеграції механізмів стробування, які регулюють потік інформації в мережі. Такий контроль над інформаційним потоком покращує адаптивність GRU та робить їх добре придатними для обмежених ресурсів або сценаріїв великомасштабних застосувань.

І LSTM, і GRU є прикладами ітеративного вдосконалення та інновацій у рекурентних архітектурах нейронних мереж. Їх дизайн вирішує конкретні проблеми, пов'язані з послідовними даними, прокладаючи шлях до прогресу в різних програмах, від розуміння природної мови та аналізу настроїв до розпізнавання мови тощо. Постійне дослідження та розробка цих архітектур продовжує сприяти прогресу в глибокому навчанні [18].

Автокодері представляють ключову категорію нейронних мереж, розроблених спеціально для завдань, пов'язаних із навчанням без нагляду та зменшенням розмірності. Ця відмінна архітектура охоплює два невід'ємних компоненти: кодер і декодер. Основна мета автокодувальника — отримати стисло, але дуже інформативне представлення вхідних даних. Перед кодером поставлено завдання стиснути вхідну інформацію в більш компактну та абстрактну форму, тоді як декодер згодом намагається відновити вихідні дані з цього стислого представлення. Цей двонаправлений процес змушує мережу виділяти найбільш помітні характеристики та істотні закономірності, властиві даним.

Універсальність автокодерів є далекосяжною, вони знаходять застосування в різноманітних сферах, починаючи від усунення шумів зображення та виявлення аномалій до навчання репрезентації тощо. Їх ефективність у неконтрольованих

умовах, де позначені навчальні дані можуть бути дефіцитними або недоступними, робить їх безцінними інструментами в машинному навчанні. Крім того, автокодери відіграють ключову роль у задачах зменшення розмірності, де вони допомагають зменшити складність і обчислювальне навантаження на обробку даних великої розмірності, зберігаючи при цьому критичну інформацію. Ця подвійна здатність позиціонує автокодери як незамінні засоби комп'ютерного зору, обробки природної мови та обробки сигналів. Постійні дослідження та інновації в цій галузі продовжують удосконалювати та розширювати можливості автокодерів, сприяючи вдосконаленню методів неконтрольованого навчання.

Варіаційні автокодери (VAE) являють собою складну варіацію традиційних автокодерів, вводячи важливий імовірнісний елемент у процес кодування. На відміну від звичайних автокодерів, які виробляють детерміновані кодування, VAE кодують дані в розподіл ймовірностей. Це означає, що замість отримання єдиного фіксованого представлення VAE надають діапазон потенційних моделей, кожна з яких має відповідну ймовірність появи. Це імовірнісне кодування надає VAE можливість стискати дані та генерувати абсолютно нові зразки даних, які відповідають вивченому розподілу [21].

VAE особливо вправні в генеративних завданнях, де метою є створення нових точок даних, які мають схожість з даними навчання. Це робить їх чудовим інструментом у генеративному моделюванні, із різними застосуваннями від синтезу зображень до генерації тексту. Можливість генерувати нові дані з вивченого розподілу мають далекосяжні наслідки, впливаючи серед інших на такі галузі, як комп'ютерна графіка, обробка природної мови та медична візуалізація.

Інновація, запропонована VAE, підкреслює їх ключову роль у розвитку можливостей методів неконтрольованого навчання. Їхня здатність як вивчати складні представлення, так і генерувати нові зразки даних із цих представлень є потужним інструментом для широкого спектру програм. Поточні дослідження в цій галузі продовжують вдосконалювати та розширювати потенціал VAE,

позиціонуючи їх як наріжний камінь у ландшафті генеративного моделювання та імовірного машинного навчання.

Генеративні змагальні мережі (GAN) представляють новаторську парадигму глибокого навчання, що характеризується унікальною подвійною мережевою архітектурою. GAN складаються з двох різних нейронних мереж, генератора та дискримінатора, які беруть участь у змагальному навчальному процесі. Ця суперницька динаміка виділяє GAN як потужний інструмент для завдань генеративного моделювання.

Компонент генератора GAN має завдання створювати абсолютно нові екземпляри даних, ефективно синтезуючи зразки, які імітують характеристики навчальних даних. Одночасно дискримінатор відповідає за розрізнення фактичних точок даних від вихідного набору даних і згенерованих моделей, створених генератором. Ця суперечлива взаємодія між двома мережами породжує безперервний цикл вдосконалення, коли кожна ітерація змушує генератор створювати все більш реалістичні дані, а дискримінатор стає більш вправним у відрізненні справжніх від сфабрикованих екземплярів.

Універсальність GAN вражає, оскільки програми охоплюють широкий спектр доменів. Вони використовувалися для синтезу зображень, уможливлуючи генерацію фотореалістичних зображень, а також у таких завданнях, як суперроздільна здатність, передача стилю та трансляція зображення в зображення. Окрім додатків, пов'язаних із зображеннями, GAN знайшли корисність у синтезі тексту в зображення, генерації голосу та навіть у створенні реалістичного середовища відеоігор.

Інновації GAN зробили революцію в генеративному моделюванні, запропонувавши надійну структуру для створення нових зразків даних високої точності. Постійні дослідження та розробки в GAN продовжують удосконалювати та розширювати їхні можливості, ще більше зміцнюючи їхню позицію як наріжного каменю в ландшафті глибокого навчання [23].

Мережі глибокої віри (DBN) — це особливий клас генеративних моделей, що характеризується своєю багаторівневою архітектурою, що включає стохастичні приховані змінні. Ця унікальна структура дозволяє DBN досягти успіху в завданнях неконтрольованого навчання, головним чином навчання функцій і зменшення розмірності.

За своєю суттю DBN складаються з кількох шарів прихованих блоків, кожен з яких взаємодіє з шаром над ним. Ця ієрархічна структура дає змогу DBN фіксувати складні шаблони та взаємозв'язки всередині даних, що робить їх особливо вправними у задачах, де розуміння складних високорівневих функцій має вирішальне значення.

Можливості неконтрольованого навчання DBN є цінними в сценаріях, коли позначені дані обмежені або недоступні. Використовуючи притаманну структуру даних, DBN можуть автономно виявляти значущі представлення, ефективно зменшуючи розмірність простору введення. Ця здатність має глибоке значення для комп'ютерного зору, природної мови та обробки сигналів.

DBN продемонстрували надзвичайну ефективність у різноманітних програмах, включаючи, але не обмежуючись, розпізнаванням зображень, аналізом мовлення. Їх здатність до адаптації та вміння виконувати завдання навчання без нагляду робить їх життєво важливим інструментом в арсеналі практиків машинного навчання. Постійні дослідження та розробки в цій галузі продовжують удосконалювати та розширювати можливості DBN, зміцнюючи їх значення в ландшафті моделей глибокого навчання.

Спочатку задуманий для завдань обробки природної мови NLP представляє новаторську архітектуру, яка використовує механізми самоконтролю для захоплення складних глобальних залежностей у даних. Цей інноваційний підхід зробив революцію в галузі, дозволивши розпаралелювати послідовності, роблячи їх високоефективними для завдань, що вимагають розуміння довготривалих залежностей. NLP знайшло широке застосування в багатьох функціях, охоплюючи

мовний переклад, резюмування тексту, аналіз настроїв тощо. Ця універсальність виникає завдяки їх адаптованості до завдань, пов'язаних зі структурованими даними, де першочерговим є визначення зв'язків між віддаленими елементами.

Однією з найвпливовіших похідних архітектури Transformer є Bidirectional Encoder Representations from Transformers (BERT), акронім від Bidirectional Encoder Representations from Transformers. BERT — це попередньо навчена модель трансформатора, спеціально розроблена для завдань розуміння природної мови. Що відрізняє BERT від інших, так це його здатність генерувати контекстуалізовані представлення слів, тобто він розуміє слова на основі їхнього контексту в реченні. Таке усвідомлення контексту значно покращує його здатність розуміти тонкі мовні нюанси, дозволяючи йому досягати успіхів у різноманітних завданнях NLP, включаючи аналіз настроїв, розпізнавання іменованих сутностей і відповіді на запитання [25].

Transformers і BERT започаткували нову еру в NLP, докорінно змінивши те, як машини сприймають і створюють людську мову. Їх вплив поширюється на різні додатки, від автоматизованих систем підтримки клієнтів до створення контенту. Постійні дослідження та вдосконалення в цій області продовжують розвивати можливості Transformers і BERT, розсуваючи межі розуміння природної мови та віку.

Капсульні мережі (CapsNets) являють собою новаторський відхід від звичайних згорткових нейронних мереж (CNN), які відрізняються своїм наголосом на захопленні детальної інформації про складові частини об'єктів і складні просторові зв'язки між ними. Цей унікальний архітектурний підхід знаменує значний крок вперед у розпізнаванні об'єктів, усуваючи помітні обмеження CNN. Капсульні мережі особливо вправно розуміють, як різні елементи в об'єкті взаємодіють і пов'язані, що робить їх надзвичайно цінними в сценаріях, де розпізнавання дрібних деталей і обробка варіацій точки огляду є критичними.

Основною концепцією капсульних мереж є використання капсул, спеціалізованих блоків нейронної мережі, які кодують інформацію про конкретні характеристики об'єкта та їх відносне розташування. На відміну від традиційних нейронних мереж, які можуть боротися з перетвореннями, такими як обертання або спотворення, капсульні мережі мають потенціал для збереження просторових зв'язків між компонентами об'єктів, дозволяючи їм обробляти варіації точки зору з більшою ефективністю.

Capsule Networks показали надзвичайну перспективу в розпізнаванні об'єктів, оцінці пози та завданнях реконструкції зображень. Їхня здатність сприймати ієрархічну структуру об'єктів та їхніх складових частин має значення для таких різноманітних галузей, як комп'ютерне бачення, робототехніка та медичне зображення. Постійне вдосконалення та дослідження Capsule Networks продовжують розширювати їхній потенціал, висуваючи їх на передовий край передових досліджень глибокого навчання та комп'ютерного зору.

Нейроеволюція доповнювальних топологій (NEAT) є ключовим досягненням у розвитку штучного інтелекту та нейронної мережі. Він являє собою еволюційний алгоритм, ретельно розроблений для розвитку штучних нейронних мереж (ШНМ). Що відрізняє NEAT, так це його здатність динамічно адаптувати структуру та топологію нейронних мереж протягом еволюційного процесу.

NEAT використовує принциповий підхід, вводячи нові нейрони та зв'язки протягом поколінь, дозволяючи мережі зростати у складності та адаптуватися до мінливих вимог завдання. Ця унікальна методологія пом'якшує загальні проблеми, пов'язані з нейронними мережами з фіксованою топологією, такі як пошук оптимальної архітектури для певної проблеми.

Застосування NEAT є далекосяжними, з основним акцентом на завданнях, що включають навчання з підкріпленням. Шляхом динамічного налаштування мережевої архітектури та з'єднань NEAT забезпечує появу нейронних мережевих структур, адаптованих до конкретних вимог складних динамічних середовищ. Це

робить NEAT особливо потужним у сценаріях, де адаптивність, надійність і оптимізація продуктивності є найважливішими.

NEAT мав трансформаційний вплив на штучний інтелект, за допомогою додатків, що охоплюють робототехніку, ігри та системи керування. Його адаптивність і універсальність позиціонують NEAT як основний інструмент для дослідників і практиків, які прагнуть використовувати потужність еволюційних алгоритмів у розробці штучних нейронних мереж. Поточні дослідження та вдосконалення в цій області продовжують розширювати потенціал NEAT, сприяючи прогресу в нейроеволюції та адаптивному навчанні.

Цей комплексний огляд досліджує поширені архітектури глибокого навчання, які використовуються для виявлення тварин, зосереджуючись на методах, заснованих на згорткових нейронних мережах (CNN). Досліджуючи та аналізуючи застосування методів, заснованих на CNN, цей огляд намагається забезпечити повне розуміння їх ефективності та потенціалу в розвитку галузі виявлення тварин [28].

Згорткові нейронні мережі (CNN) — це методи глибокого навчання, які добре підходять для завдань на основі зображень. Вони розроблені, щоб імітувати візуальну обробку людського мозку та збираються з кількох шарів згорткових фільтрів і операцій об'єднання. Ці шари дозволяють CNN вивчати ієрархічні характеристики з вхідних зображень, дозволяючи виявляти складні візерунки та об'єкти.

Згорткові нейронні мережі (CNN) є спеціальним типом штучної нейронної мережі прямого зв'язку. На їх структурну організацію впливає розташування клітин зорової кори тварин. У зоровій корі невеликі кластери клітин виявляють чутливість до певних ділянок поля зору. Нейрональні клітини в мозку реагують вибірково, запускаючи лише за наявності крайових орієнтацій. Наприклад, деякі нейрони активуються у випадку вертикальних країв, тоді як інші роблять це для горизонтальних або діагональних країв. CNN, які використовуються в глибокому

навчанні, призначені для оцінки візуальної інформації. Вони можуть вирішувати широкий спектр завдань, включаючи обробку зображень, звуків, текстів, відео та різних інших форм медіа.

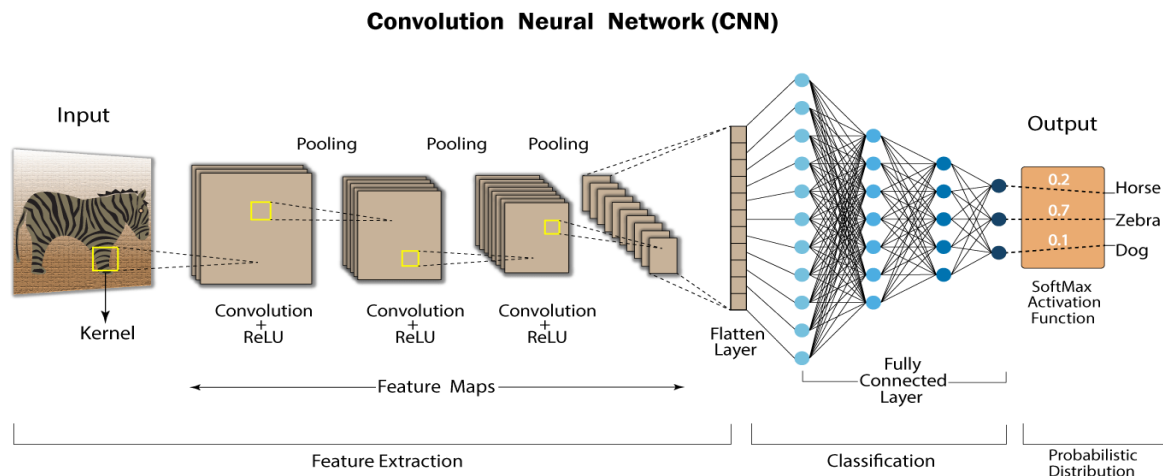


Рисунок 1.1 – Графічне представлення згорткової нейронної мережі

Згорткові нейронні мережі (CNN) мають вхідний рівень, кілька прихованих шарів, вихідний рівень і багато параметрів, що дозволяє їм вміло розпізнавати складні об'єкти та заплутані візерунки. Ці мережі використовують процеси згортання та об'єднання для зменшення вибірки вхідних даних перед застосуванням функції активації. Ці операції переважно виконуються в межах частково з'єднаних прихованих шарів, кульмінацією яких є повністю з'єднаний шар, який дає результат, показаний на рис. 1.1.

Результуючий вихід із CNN зберігає просторову розмірність, як вихідне вхідне зображення. Згортка, у цьому контексті, передбачає об'єднання двох функцій для створення результату останньої функції. У CNN вхідне зображення піддається згортці шляхом застосування фільтрів, у результаті чого створюється карта функцій. Ці фільтри містять випадково згенеровані вектори, що охоплюють ваги та зміщення в мережі. Для кожного нейрона, CNN використовують однакові

ваги та зміщення для всіх нейронів. Кілька фільтрів можуть бути створені, захоплюючи окремі аспекти вхідних даних. Фільтри також називаються ядрами.

У виявленні тварин на відеоданих CNN відіграють вирішальну роль у виділенні та розпізнаванні ознак. Початкові рівні CNN ідентифікують скромні особливості зі схожими краями та текстурями, тоді як глибші рівні вивчають величезні абстрактні та розрізнявальні особливості, які мають значення для виявлення тварин. CNN можна налаштувати за допомогою мічених зображень тварин, щоб адаптувати модель для завдань виявлення тварин. Широко використовуються згорткові нейронні мережі (CNN), включаючи такі варіанти, як YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), R-CNN, Mask R-CNN і RetinaNet, які пропонують об'єкти в реальному часі або з високою точністю. виявлення. Тимчасові аспекти розглядаються за допомогою I3D (Inflated 3D ConvNet) і T-CNN (Temporal CNN), які фіксують сигнали руху. Наприклад, для сегментації та відстеження застосовуються Mask R-CNN і Tube-CNN. Просторово-часові взаємодії підкреслюються STSN (Spatio-Temporal Snippet Network) і STAM (Spatio-Temporal Механізм уваги). Більш сучасні підходи використовують зіставлення меж (BMN) і прогнозування траєкторії (TrajNet++) для покращеного відстеження .

Ці архітектури використовують здатність глибокого навчання вивчати складні шаблони та часові залежності, забезпечуючи точне й ефективне виявлення тварин у відеоданих [25].

Ці дослідження, висвітлюють різноманітні застосування глибокого навчання для виявлення тварин, досягнення високих показників точності та переваг автоматизації, але стикаються з проблемами щодо вимог до даних, обчислювальної інтенсивності та узагальнення. Вибір архітектури залежить від конкретних вимог програми, таких як продуктивність у реальному часі, точність виявлення та обчислювальні ресурси.

1.3 Аналіз мов програмування

Python - це потужна та динамічна мова програмування, яка широко використовується у різних галузях, включаючи розробку веб-додатків, обробку даних, штучний інтелект, наукові дослідження та багато іншого. Однією з головних переваг Python є його простота та легкість вивчення. Він має зрозумілу та читабельну синтаксичну конструкцію, що робить його ідеальним вибором для початківців у програмуванні.

Python є високорівневою мовою програмування, що означає, що вона наближена до природної мови та абстрагує від деталей машинного коду. Це дозволяє розробникам швидко створювати програми та витратити менше часу на написання коду, що робить їх більш продуктивними.

Однією з сильних сторін Python є його велика та активна спільнота розробників. У Python існує велика кількість сторонніх бібліотек та фреймворків, які спрощують розробку програм та розширюють їх можливості. Наприклад, для обробки зображень є бібліотеки як OpenCV, для машинного навчання - TensorFlow, PyTorch, Scikit-learn та інші, а для веб-розробки - Django, Flask, FastAPI.

Ще однією важливою особливістю Python є його переносимість. Код, написаний на Python, може запускатися на різних операційних системах без необхідності внесення змін. Це дозволяє розробникам створювати програми, які працюють на різних платформах та пристроях без особливих зусиль.

Загалом, Python - це потужний та універсальний інструмент, який знаходить своє застосування в різних галузях та завданнях, починаючи від простих сценаріїв до складних систем та додатків.

C++ є потужною та універсальною мовою програмування, яка широко використовується для розробки швидких та ефективних програм у різних галузях, включаючи системне програмування, ігрову розробку, наукові обчислення, вбудовані системи та багато іншого. Вона є однією з найбільш впливових мов

програмування та має широкий набір функціональностей, які дозволяють розробникам створювати складні програми.

Однією з ключових особливостей C++ є його можливості маніпулювання пам'яттю, що дозволяє розробникам докладно керувати ресурсами та оптимізувати продуктивність програм. Вона має високопродуктивну систему управління пам'яттю через вказівники, динамічне виділення пам'яті та можливість роботи з низькорівневими операціями, такими як робота з бітовими операціями та маніпулювання побайтовою адресою.

C++ також володіє багатою стандартною бібліотекою, яка включає в себе різноманітні класи та функції для роботи з контейнерами даних, рядками, введенням-виведенням, алгоритмами та багато іншого. Ця бібліотека дозволяє розробникам швидко створювати програми та мінімізувати кількість написаного коду.

Ще однією важливою особливістю C++ є його можливості багатопотокового програмування. Вона має потужні функції для створення та управління потоками виконання, що дозволяє розробникам створювати паралельні програми та використовувати багатопроцесорні системи для покращення продуктивності.

Нарешті, C++ є високпереносимою мовою програмування, що означає, що програми, написані на C++, можуть запускатися на різних операційних системах без необхідності внесення змін у вихідний код. Це дозволяє розробникам створювати програми, які працюють на різних платформах та пристроях без особливих зусиль.

1.4 Аналіз бібліотек для роботи з комп'ютерним зором

PyTorch - це відкрите програмне забезпечення для машинного навчання та обробки даних, яке розробляється та підтримується командою Facebook. Вона

базується на мові програмування Python та надає зручний інтерфейс для розробки та навчання нейронних мереж.

Однією з ключових переваг PyTorch є її динамічна обчислювальна графіка. Це означає, що граф обчислень створюється "на льоту" під час виконання програми, що дозволяє розробникам з легкістю вносити зміни та відлагоджувати моделі нейронних мереж. Також PyTorch підтримує статичну обчислювальну графіку за допомогою модулю TorchScript, що робить його більш гнучким у порівнянні з іншими бібліотеками.

PyTorch має багатий набір функцій та модулів для розробки та навчання нейронних мереж. Вона містить різноманітні типи шарів, функції активації, оптимізатори, втрачені функції та інші компоненти, які допомагають розробникам створювати складні та потужні моделі нейронних мереж. Крім того, PyTorch має вбудовану підтримку для автоматичного диференціювання (autograd), що дозволяє автоматично обчислювати градієнти для оптимізації параметрів моделі.

PyTorch також має активну та швидко зростаючу спільноту розробників, яка надає різноманітні ресурси для навчання та підтримки. Існують офіційна документація, навчальні матеріали, блоги, форуми та інші ресурси, які допомагають розробникам вивчити та використовувати PyTorch для розвитку своїх проектів.

Узагальнюючи, PyTorch є потужним та гнучким інструментом для розробки нейронних мереж та застосування їх у різних областях, включаючи комп'ютерний зір, обробку мови, машинне навчання та багато іншого. Вона забезпечує широкі можливості для розробників та дослідників у сфері штучного інтелекту та машинного навчання.

TensorFlow - це відкрите програмне забезпечення для машинного навчання та глибинного навчання, розроблене компанією Google. Вона є однією з найпопулярніших бібліотек для розробки та навчання нейронних мереж і широко використовується в індустрії та академії.

Однією з ключових особливостей TensorFlow є її графова обчислювальна модель. Вся робота з обчисленнями в TensorFlow відбувається у вигляді графа обчислень, де вузлами є операції, а ребрами - тензори (масиви даних). Це дозволяє оптимізувати виконання обчислень та забезпечує високу продуктивність.

TensorFlow надає розробникам широкий набір інструментів для розробки та навчання нейронних мереж. Вона містить різноманітні вбудовані модулі для створення різних типів шарів, функцій активації, втрат, оптимізаторів та інших компонентів нейронних мереж. Крім того, TensorFlow має вбудовану підтримку для автоматичного диференціювання, що дозволяє автоматично обчислювати градієнти для оптимізації параметрів моделі.

TensorFlow також надає широкий набір інструментів для розгортання та моніторингу моделей. Вона підтримує різні платформи, такі як мобільні пристрої, вбудовані системи та хмарні сервіси, що дозволяє розробникам легко розгорнути та використовувати моделі нейронних мереж у різних середовищах.

TensorFlow має велику та активну спільноту розробників, яка надає різноманітні ресурси для навчання та підтримки. Існують офіційна документація, навчальні матеріали, блоги, форуми та інші ресурси, які допомагають розробникам вивчити та використовувати TensorFlow для розвитку своїх проєктів.

Узагальнюючи, TensorFlow є потужним та гнучким інструментом для розробки нейронних мереж та їх застосування у різних областях, включаючи комп'ютерний зір, обробку мови, машинне навчання та багато іншого. Вона забезпечує широкі можливості для розробників та дослідників у сфері штучного інтелекту та машинного навчання [33].

OpenCV (Open Source Computer Vision Library) - це бібліотека відкритого програмного забезпечення, яка надає інструменти для обробки зображень та комп'ютерного зору. Вона має великий набір функцій для розпізнавання облич, виявлення об'єктів, вимірювання відстаней, відстеження руху, розпізнавання шаблонів, обробки відео та багато іншого.

Однією з ключових переваг OpenCV є її крос-платформенність. Вона підтримує різні операційні системи, такі як Windows, Linux, macOS, а також може працювати на різних архітектурах, включаючи x86, ARM та CUDA. Це дозволяє розробникам створювати програми на різних платформах без переписування коду.

OpenCV має простий та зрозумілий інтерфейс програмування на різних мовах, включаючи C++, Python та Java. Це робить OpenCV доступним для широкого кола розробників з різними рівнями кваліфікації та досвідом програмування.

Однією з ключових функцій OpenCV є обробка зображень та відео. Вона надає широкий спектр алгоритмів для розпізнавання та виявлення об'єктів, вимірювання відстаней та кутів, фільтрації та покращення зображень, а також інші інструменти для обробки графічної інформації.

OpenCV також має багато вбудованих модулів для роботи з комп'ютерним зором. Наприклад, вона має модуль для розпізнавання облич, модуль для виявлення об'єктів, модуль для вимірювання відстаней та модуль для відстеження руху.

Узагальнюючи, OpenCV є потужним та універсальним інструментом для обробки зображень та комп'ютерного зору. Вона має великий набір функцій та алгоритмів, які дозволяють розробникам створювати різноманітні застосунки в областях від розпізнавання облич до автоматизованого контролю якості.

РОЗДІЛ 2. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ

2.1 Розробка програми

При проектуванні архітектури і принципів функціонування програми було розроблено наступні діаграми:

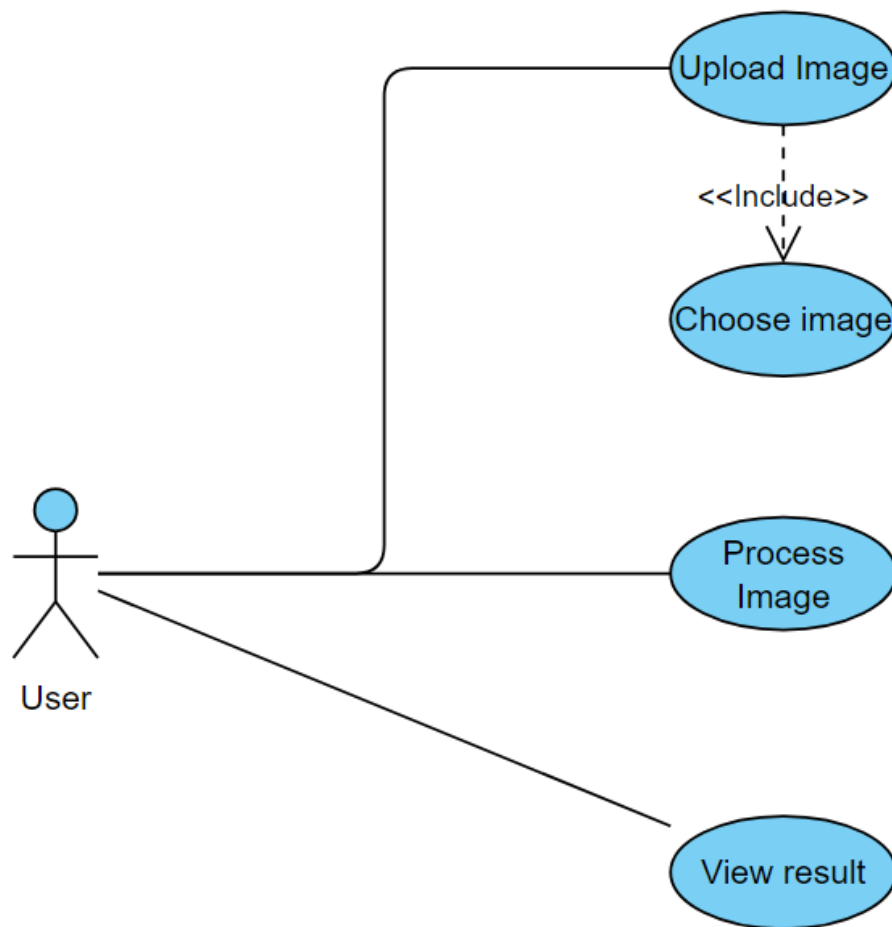


Рисунок 2.1 - Діаграма використання (Use-case diagram)

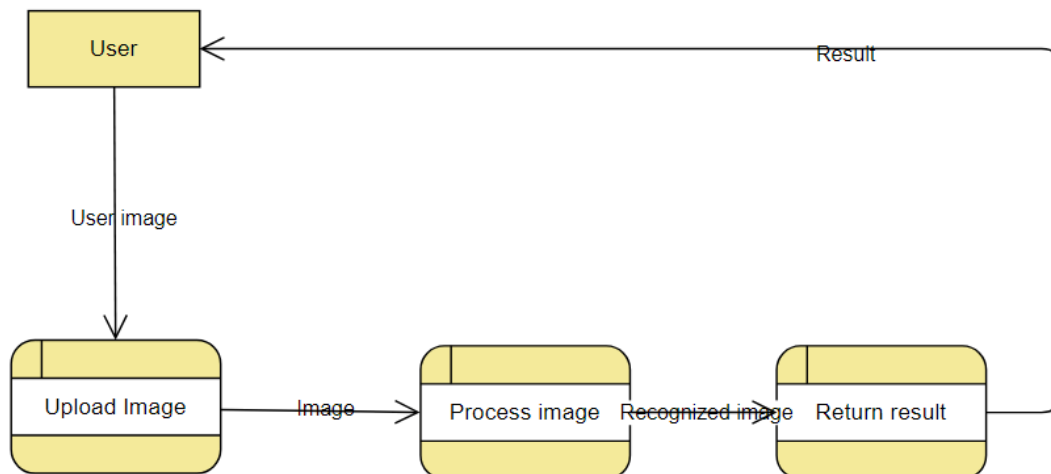


Рисунок 2.2 – Діаграма потоків даних (Data Flow diagram)

Повний лістинг коду додатку наведений у Додатку А.

```

import argparse
import io from PIL
import Image import torch from flask
import Flask, render_template, request, redirect
  
```

Лістинг 2.1 – Імпорт модулів

Даний блок коду містить імпорти необхідних бібліотек та модулів для роботи з Flask-додатком. Ось їх опис:

`import argparse`: цей модуль дозволяє створювати зручні інтерфейси командного рядка для програм. В даному випадку, він може використовуватися для передачі аргументів у командний рядок під час запуску програми Flask.

`import io`: цей модуль дозволяє працювати з потоками вводу/виводу (I/O). У даному випадку, він може використовуватися для роботи з об'єктами байтового потоку.

`from PIL import Image`: цей модуль надає функціонал для роботи з зображеннями. Використовується для завантаження, збереження та обробки зображень.

`import torch`: цей модуль є основою для працюючих з нейромережами та обчисленнями. У даному випадку, він використовується для роботи з моделями нейромереж, наприклад, YOLOv5.

`from flask import Flask, render_template, request, redirect`: Flask - це мікрофреймворк для веб-розробки на Python. У цьому рядку імпортуються основні функції та класи, необхідні для роботи з Flask:

Flask: клас для створення об'єкту додатка Flask.

`render_template`: функція для відображення HTML-шаблонів.

`request`: об'єкт, який містить дані про запит, що надходить від клієнта.

`redirect`: функція для перенаправлення клієнта на іншу сторінку.

```
TARGET_CLASSES = [ "bird", "cat", "dog", "horse", "cow", "elephant",
"bear", "zebra", "giraffe"]
```

Лістинг 2.2 – Перелік класів об'єктів для розпізнавання (видів тварин)

Цей рядок коду визначає перелік класів об'єктів, які потрібно виявляти за допомогою моделі. Список з дев'яти класів об'єктів, які повинні бути розпізнані моделлю. Це такі об'єкти, як птах, кіт, собака, кінь, корова, слон, ведмідь, зебра та жираф.

```
@app.route("/", methods=["GET", "POST"])
def predict():
    if request.method == "POST":
        if "file" not in request.files:
            return redirect(request.url)
        file = request.files["file"]
        if not file:
            return

        img_bytes = file.read()
        img = Image.open(io.BytesIO(img_bytes))
```

```

results = model(img)

# Filter results to include only target classes
filtered_preds = [pred for pred in results.pred[0] if
model.names[int(pred[5])] in TARGET_CLASSES]

# Update results with filtered predictions
results.pred[0] = torch.stack(filtered_preds) if
filtered_preds else torch.zeros((0, 6))

results.render() # updates results.imgs with boxes and labels
results.save(save_dir="static/", exist_ok=True)

return redirect("static/image0.jpg")

return render_template("index.html")

```

Лістинг 2.3 – Головна сторінка додатку

Цей код описує функцію `predict`, яка обробляє запити до кореневого маршруту веб-додатку. Функція може обробляти як GET, так і POST запити. Коли надходить POST запит, функція перевіряє наявність файлу у запиті. Якщо файл відсутній, користувача перенаправляють на ту ж сторінку. Якщо файл присутній, він зчитується в байти, після чого завантажується як зображення за допомогою бібліотеки PIL. Потім зображення передається у модель для отримання результатів детекції. Далі результати фільтруються так, щоб залишити тільки ті, що відповідають заданим класам об'єктів, які визначені у списку `TARGET_CLASSES`. Якщо після фільтрації залишаються результати, вони оновлюються, інакше створюється порожній тензор. Після цього результати візуалізуються (до зображень додаються рамки та мітки) та зберігаються у статичну директорію. Після збереження користувача перенаправляють на сторінку зі збереженим зображенням. Якщо ж запит був GET, функція повертає HTML шаблон для завантаження зображення.

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Flask app exposing
yolov5 models")
    parser.add_argument("--port", default=5000, type=int, help="port
number")

```



```
args = parser.parse_args()

model = torch.hub.load('ultralytics/yolov5', 'yolov5s',
pretrained=True) # force_reload = recache latest code
model.eval()
app.run(host="0.0.0.0", port=args.port) # debug=True causes
Restarting with stat
```

Лістинг 2.4 – Обробка запуску додатку

Цей код виконується, коли скрипт запускається безпосередньо. Спочатку створюється об'єкт `ArgumentParser` для обробки аргументів командного рядка, зокрема аргументу `--port`, який визначає номер порту для запуску додатку (за замовчуванням 5000). Далі ці аргументи зчитуються і зберігаються у змінній `args`. Потім завантажуються попередньо навчена модель YOLOv5 з бібліотеки `torch.hub` за допомогою команди `torch.hub.load`, причому використовується версія `yolov5s`. Модель переведена в режим оцінки (`eval`), щоб забезпечити її використання тільки для передбачень і не змінювати її параметри під час роботи. Нарешті, додаток Flask запускається на всіх мережевих інтерфейсах (`host="0.0.0.0"`) та на порту, вказаному в аргументах командного рядка (або за замовчуванням на порту 5000).

2.2 Випробування розробленого рішення

Після початку роботи з додатком потрапляємо на його головну сторінку:

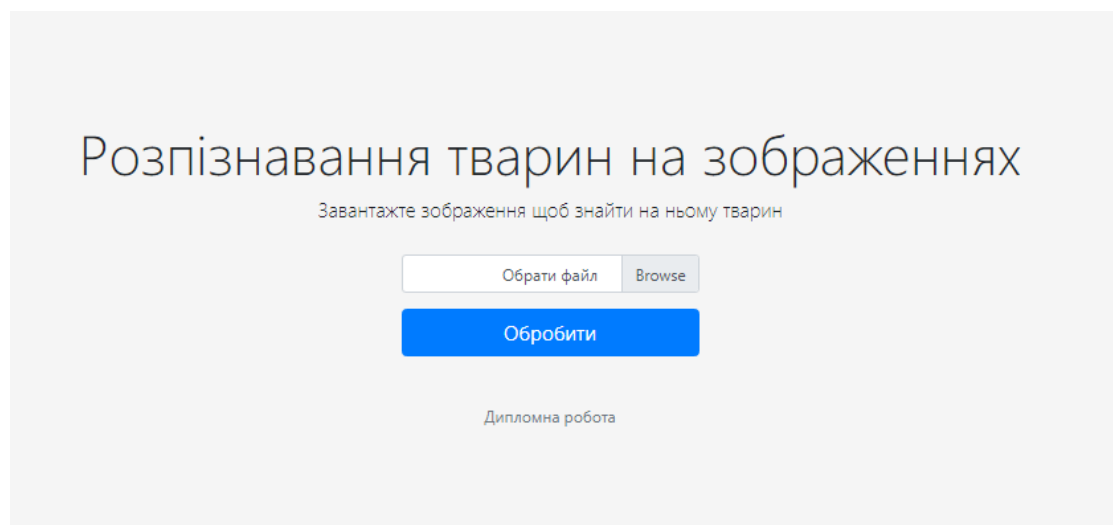


Рисунок 2.3 – Головна сторінка додатку

Тут можемо побачити заголовок, заклик завантажити зображення, а також форму для завантаження файлу з кнопкою відправлення зображення на обробку.

Для того, щоб протестувати роботу додатку, візьємо тестове зображення, яке містить тварин (собака та кіт):



Рисунок 2.4 – Тестове зображення

Натиснувши кнопку «Обробити», отримуємо наступний результат:

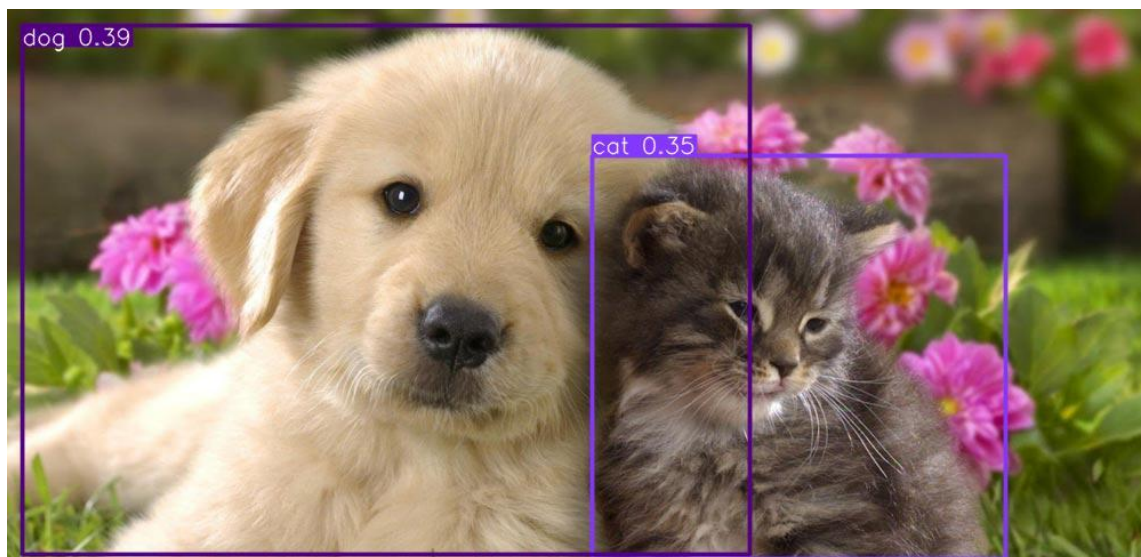


Рисунок 2.5 – Результат обробки

На отриманому зображенні видно, що програма виявила собаку і кота, і помітила їх відповідними рамочками.

Спробуємо інше зображення:



Рисунок 2.6 – Тестове зображення

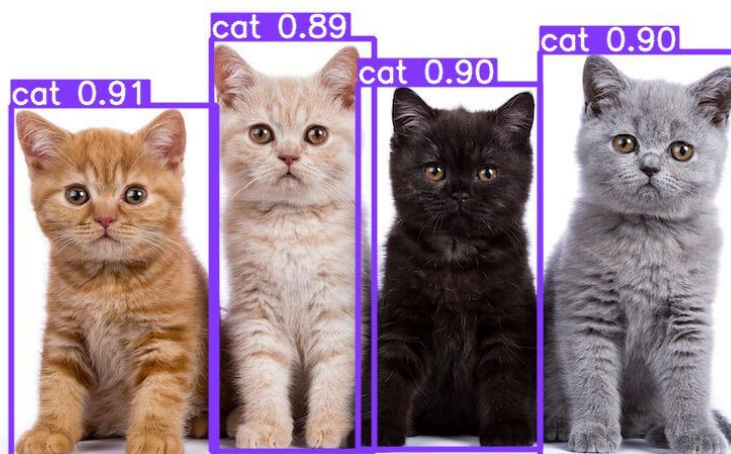


Рисунок 2.7 – Результат обробки

На отриманому вище зображенні бачимо, що система успішно розпінала одразу чотирьох кошенят, що є на зображенні.

Далі, проведемо випробування системи декількома зображеннями різних видів тварин:



Рисунок 2.8 – Тестове зображення



Рисунок 2.9 – Результат обробки



Рисунок 2.10 – Тестове зображення

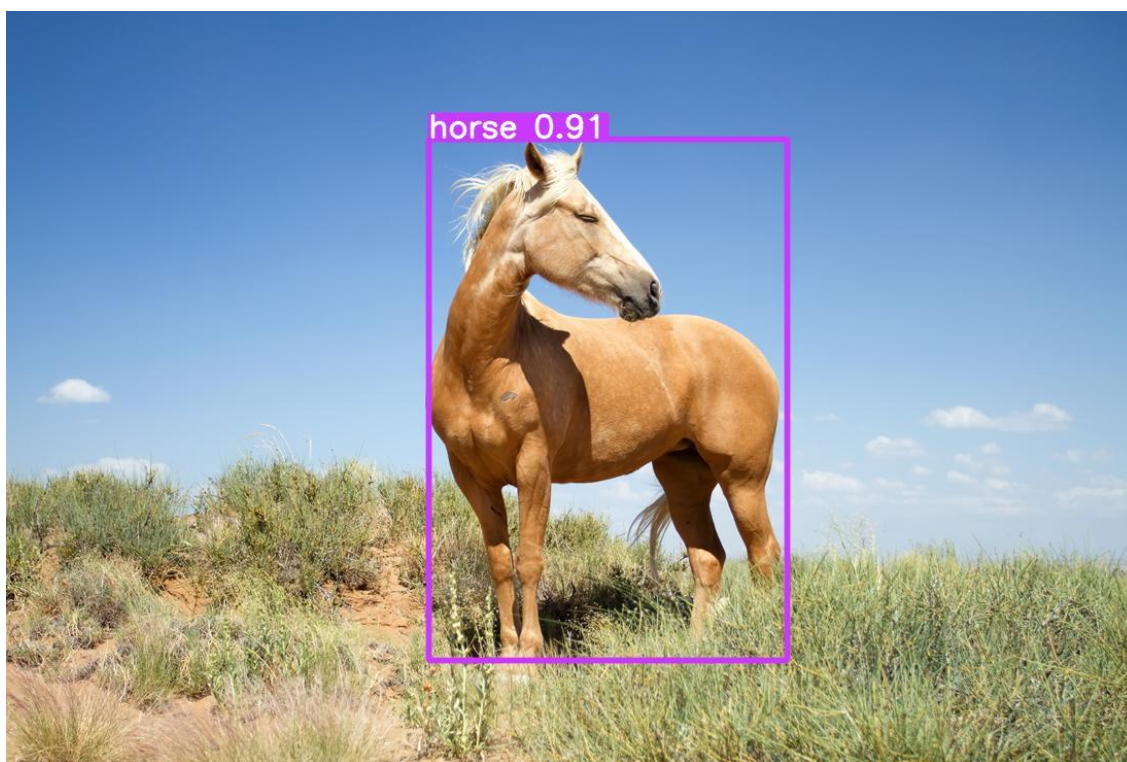


Рисунок 2.11 – Результат обробки

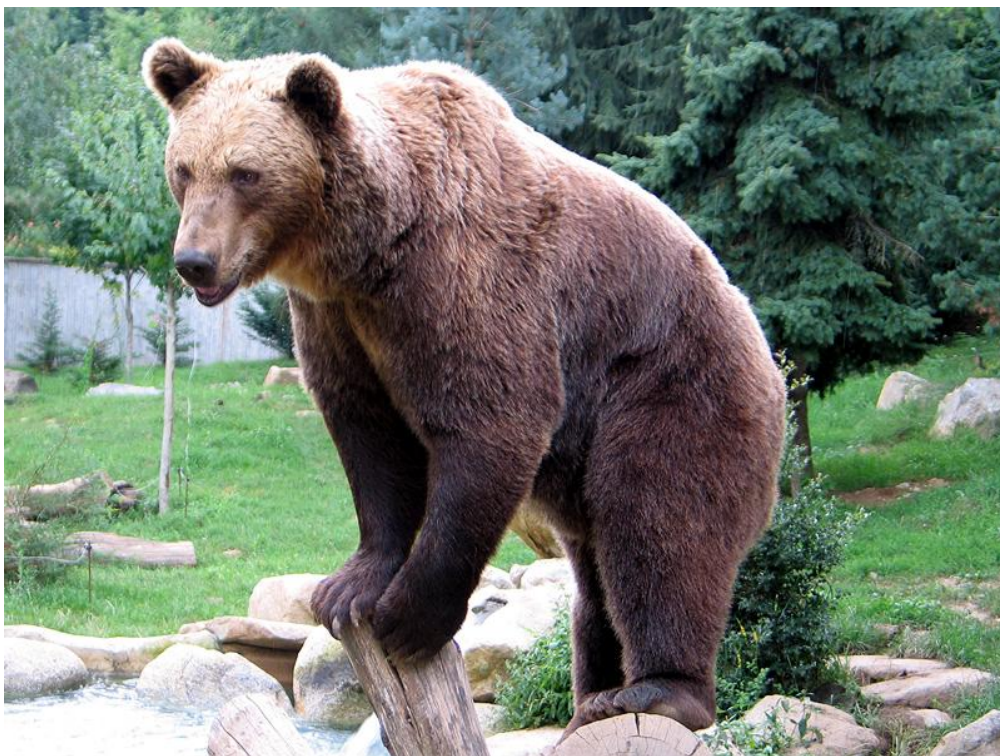


Рисунок 2.12 – Тестове зображення

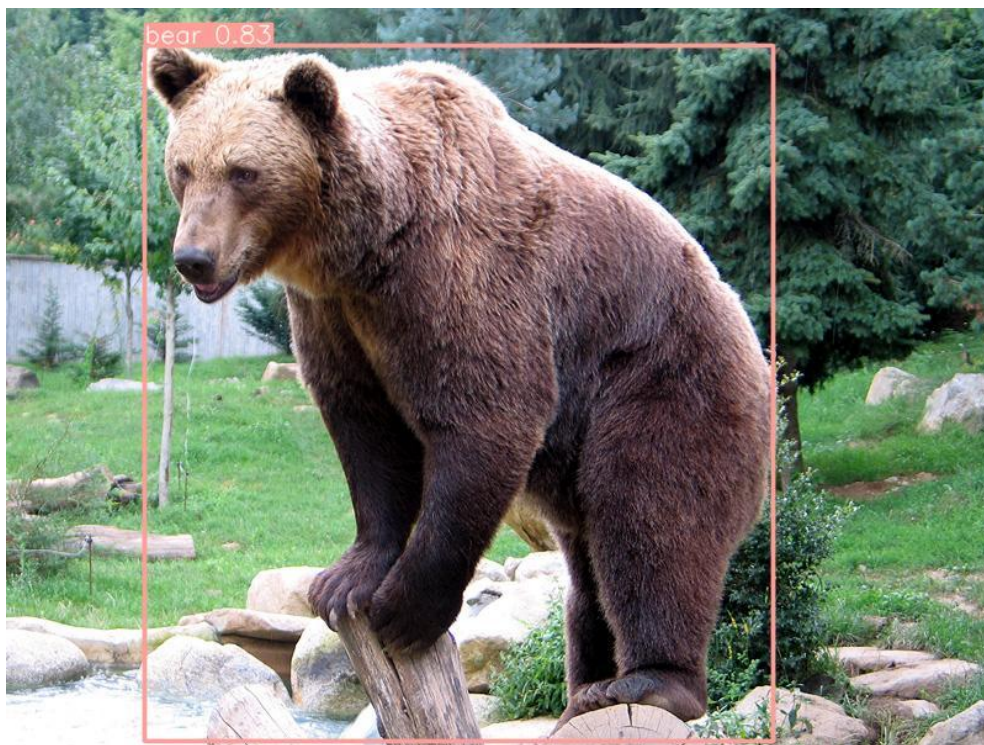


Рисунок 2.13 – Результат обробки



Рисунок 2.14 – Тестове зображення

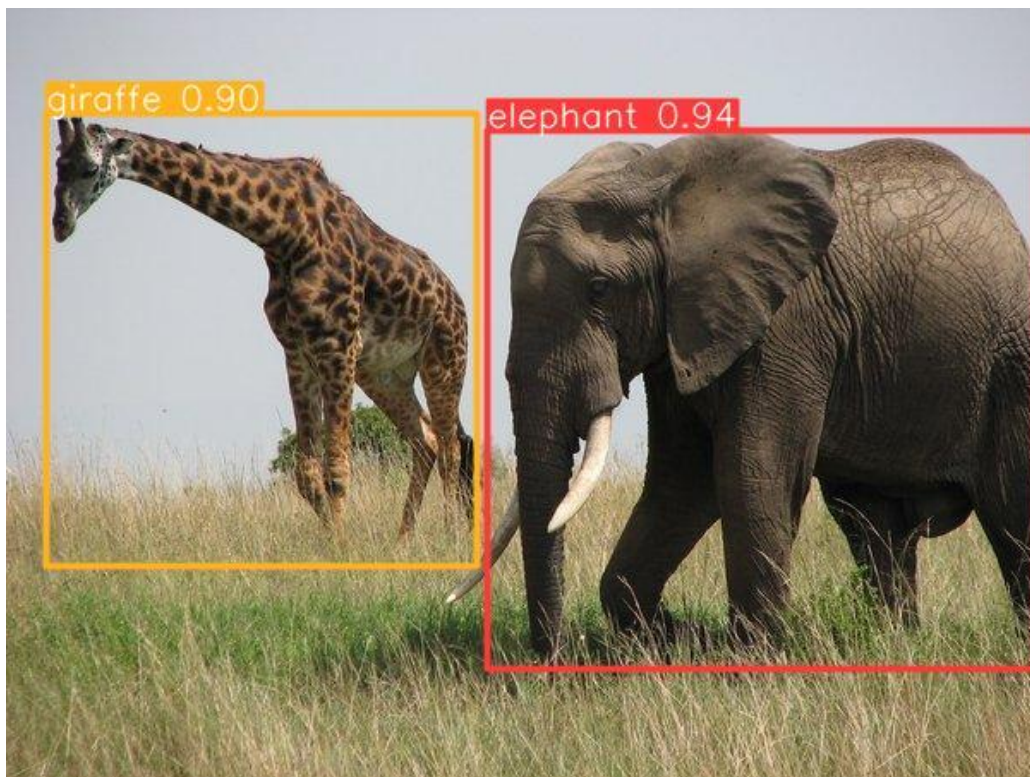


Рисунок 2.15 – Результат обробки

ВИСНОВКИ

Тема розпізнавання об'єктів є надзвичайно важливою у сучасному світі, зважаючи на стрімкий розвиток технологій штучного інтелекту та машинного навчання. Розпізнавання об'єктів дозволяє автоматизувати процеси ідентифікації та класифікації різних об'єктів на зображеннях та відео, що має велике значення для багатьох галузей, включаючи безпеку, транспорт, медицину та промисловість. Завдяки цьому, системи відеоспостереження можуть автоматично виявляти підозрілі об'єкти або поведінку, автомобілі з автопілотом можуть безпечно рухатися по дорогах, а в медичних додатках можна швидко і точно діагностувати хвороби за допомогою аналізу медичних зображень.

Особливу увагу заслуговує розпізнавання тварин, яке має безліч корисних застосувань у природоохоронних ініціативах та сільському господарстві. У сфері охорони навколишнього середовища автоматичне розпізнавання тварин допомагає в моніторингу популяцій дикої природи, що сприяє збереженню біорізноманіття та забезпеченню екологічного балансу. За допомогою цих технологій можна здійснювати безперервний контроль за чисельністю та поведінкою рідкісних та зникаючих видів, що дозволяє своєчасно вживати заходів для їх захисту.

У першому розділі даної роботи було проаналізовано предметну область, зокрема розглянуто проблематику комп'ютерного зору, використання в ньому машинного навчання, розпізнавання тварин та моделей машинного навчання, які в цьому допомагають, також було проведено аналіз інструментів і технологій, які можуть бути використані при розробці власних рішень та взаємодії з засобами штучного інтелекту.

У другому розділі було спроектовано, розроблено та випробувано можливості системи розпізнавання тварин на базі моделі розпізнавання об'єктів YOLOv5. Система дозволяє користувачу завантажувати власні зображення та

розпізнає на них образи тварин і повертає зображення, на якому знайдені об'єкти відмічені рамками.

Таким чином, тема розпізнавання об'єктів, зокрема тварин, є надзвичайно актуальною завдяки своїй значущості для різних сфер діяльності людини. Вона відкриває нові можливості для автоматизації та покращення ефективності процесів, що сприяє загальному прогресу та розвитку суспільства. Використання передових алгоритмів глибокого навчання дозволяє досягти високої точності та швидкості розпізнавання, що робить ці технології особливо перспективними та важливими для майбутнього.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алекс Крізевскі, Ілля Сутскевер, та Джеффри Е. Хінтон. Класифікація Imagenet з глибокими згортковими нейронними мережами // *Advances in neural information processing systems*. С. 1097–1105.
2. Оуні Ханнун, Карл Кейс, Джаред Каспер, Браян Катанзаро, Грег Діамос, Еріх Ельсен, Раян Пренджер, Санджив Сатхеш, Шубхо Сенгупта, Адам Коутс, та ін. Deepspeech: Масштабування кінця-до-кінця розпізнавання мови // *arXiv preprint arXiv:1412.5567*.
3. Томас Міколов, Ілля Сутскевер, Кай Чен, Грег С. Коррадо, та Джефф Дін. Розподілені представлення слів і фраз та їхня композиційність // *Advances in Neural Information Processing Systems*. С. 3111–3119.
4. Йошуа Бенгіо, Аарон К. Курвіль, та Паскаль Вінсент. Ненаглядне навчання ознак та глибоке навчання: огляд і нові перспективи // *CoRR*, abs/1206.5538. Т. 1.
5. Адам Коутс, Броді Гувал, Тао Ван, Девід Ву, Браян Катанзаро, та Ендрю Нг. Глибоке навчання з використанням стандартних НРС систем // *Proceedings of The 30th International Conference on Machine Learning*. С. 1337–1345.
6. Куок В. Ле. Побудова високорівневих ознак з використанням великомасштабного ненаглядного навчання // *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*. С. 8595–8598. IEEE.
7. Ян ЛеКун, Леон Ботту, Йошуа Бенгіо, та Патрік Хафнер. Навчання на основі градієнтів, застосоване до розпізнавання документів // *Proceedings of the IEEE*. Т. 86, Вип. 11. С. 2278–2324.
8. Джефф Хокінс та Діліп Джордж. Ієрархічна темпоральна пам'ять: концепції, теорія та термінологія // *Technical report, Numenta*.

9. Янів Тайгман, Мін Ян, Марк'Ауреліо Ранцато, та Ліор Вольф. Deepface: Скорочення розриву до людського рівня продуктивності в розпізнаванні облич // Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. С. 1701–1708. IEEE.
10. Чак Розенберг. Покращення пошуку фото: Крок через семантичний розрив // Google Research Blog. URL: <http://googleresearch.blogspot.it/2013/06/improving-photo-search-step-across.html>.
11. Давіде Мальтоні. Розпізнавання образів за допомогою ієрархічної темпоральної пам'яті // Technical report, DEIS - University of Bologna. URL: http://cogprints.org/9187/1/HTM_TR_v1.0.pdf.
12. Wikimedia Commons. Анатомія мультиполярного нейрона. URL: https://upload.wikimedia.org/wikipedia/commons/1/10/Blausen_0657_MultipolarNeuron.png.
13. Лука Ніколіні. Згорткова нейронна мережа.
14. Тьяго М. Джеронімо, Карлос Е. Д. Крус, Едуардо С. Б'янкі, Фернандо де Соуза Кампос, та Пауло Р. Агір. MLP та ANFIS, застосовані до передбачення діаметрів отворів у процесі свердління // INTECH Open Access Publisher.
15. Давіде Мальтоні та Ерік М. Рен. Інкрементальне навчання шляхом передачі повідомлень у ієрархічній темпоральній пам'яті // Artificial Neural Networks in Pattern Recognition. С. 24–35. Springer.
16. Ян ЛеКун, Фу Джіе Хуан, та Леон Ботту. Методи навчання для загального розпізнавання об'єктів із інваріантністю до положення та освітлення // Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Т. 2. С. II–97. IEEE.
17. Стюарт Рассел, Пітер Норвіг, та Штучний інтелект. Сучасний підхід // Artificial Intelligence. Prentice-Hall, Englewood Cliffs. Т. 25.

18. Ришард С. Міхальські, Джеймі Г. Карбонелл, та Том М. Мітчелл. Машинне навчання: Підхід штучного інтелекту // Springer Science & Business Media.
19. Юрген Шмідхубер. Глибоке навчання в нейронних мережах: огляд // Neural Networks. Т. 61. С. 85–117.
20. Філ Саймон. Занадто великий, щоб ігнорувати: Бізнес-аргументи для великих даних // John Wiley & Sons.
21. Том М. Мітчелл. Машинне навчання // Burr Ridge, IL: McGraw Hill. Т. 45.
22. Стеван Харнад. Гра з анотаціями: Про Тьюринга (1950) про обчислювальні машини та інтелект // The Turing Test Sourcebook: Philosophical and Methodological Issues in the Quest for the Thinking Computer.
23. Дж. Редмон та ін. Ви дивитесь лише один раз: уніфіковане виявлення об'єктів у реальному часі // матеріали конференції IEEE з комп'ютерного бачення та розпізнавання образів (CVPR). С. 779-788.
24. В. Лю та ін. SSD: Single shot multibox detector // Європейська конференція з комп'ютерного зору. С. 21-37. Спрингер.
25. С. Рен та ін. Швидший R-CNN: на шляху до виявлення об'єктів у реальному часі за допомогою мереж регіональних пропозицій // Досягнення в нейронних системах обробки інформації. С. 91-99.
26. М. Сендлер та ін. Інвертовані залишки та лінійні вузькі місця: мобільні мережі для класифікації, виявлення та сегментації // матеріали конференції IEEE з комп'ютерного зору та розпізнавання образів. С. 4510-4520.
27. А. Гейгер, П. Ленц і Р. Уртасун. Чи готові ми до автономного водіння? Пакет тестів KITTI vision // Конференція з комп'ютерного зору та розпізнавання образів (CVPR).

28. Л. Дуат та ін. Кодер-декодер із згорткою, що відокремлюється атрусом, для сегментації семантичного зображення // матеріали Європейської конференції з комп'ютерного зору (ECCV). С. 801-818.
29. Х.-К. Нгуєн, Т.-Х. Нгуєн, Р. Шерер, В.-Х. Ле. Уніфікована наскрізна структура YOLOv5-HR-TCM для автоматичної 2D/3D оцінки пози людини для додатків у реальному часі // Датчики. Т. 22, Вип. 22. С. 5419. doi: 10.3390/s22145419.
30. С. Цао та ін. Виявлення об'єктів у режимі реального часу в доповненій реальності // Transactions IEEE on Visualization and Computer Graphics. Т. 24, Вип. 1. С. 17-27.
31. Х.-С. Шин та ін. Глибокі згорткові нейронні мережі для автоматизованого виявлення: архітектури CNN, характеристики набору даних і навчання передачі // IEEE Transactions on Medical Imaging. Т. 35, Вип. 5. С. 1285-1298.
32. Ю. Дуан, Х. Чен, Р. Хаутхут, Дж. Шульман, П. Аббел. Порівняльний аналіз глибокого навчання з підкріпленням для постійного контролю // Міжнародна конференція з машинного навчання. С. 1329-1338.
33. Ю. Чжан, Р. Гроссе. Track, then Decide: Category-Agnostic Vision-based Multi-Object Tracking // preprint arXiv arXiv:1806.07235.

ДОДАТОК А. ЛІСТИНГ КОДУ ПРОГРАМИ

```

import argparse
import io
from PIL import Image
import torch
from flask import Flask, render_template, request, redirect

app = Flask(__name__)

# Define the classes you want to detect, e.g., person, car, dog
TARGET_CLASSES = [ "bird", "cat", "dog", "horse", "cow", "elephant",
"bear", "zebra", "giraffe"]

@app.route("/", methods=["GET", "POST"])
def predict():
    if request.method == "POST":
        if "file" not in request.files:
            return redirect(request.url)
        file = request.files["file"]
        if not file:
            return

        img_bytes = file.read()
        img = Image.open(io.BytesIO(img_bytes))
        results = model(img)

        # Filter results to include only target classes
        filtered_preds = [pred for pred in results.pred[0] if
model.names[int(pred[5])] in TARGET_CLASSES]

        # Update results with filtered predictions
        results.pred[0] = torch.stack(filtered_preds) if
filtered_preds else torch.zeros((0, 6))

        results.render() # updates results.imgs with boxes and labels
        results.save(save_dir="static/", exist_ok=True)

        return redirect("static/image0.jpg")

    return render_template("index.html")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Flask app exposing
yolov5 models")
    parser.add_argument("--port", default=5000, type=int, help="port
number")
    args = parser.parse_args()

    model = torch.hub.load('ultralytics/yolov5', 'yolov5s',
pretrained=True) # force_reload = recache latest code

```

```

model.eval()
app.run(host="0.0.0.0", port=args.port) # debug=True causes
Restarting with stat

```

Лістинг А.1 – Код програми

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
    <link rel="stylesheet"
href="//stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.c
ss" integrity="sha384-
GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMP0DgiMDm4iYMj70gZWKYbI706tWS"
crossorigin="anonymous">
    <style>
      .bd-placeholder-img {
        font-size: 1.125rem;
        text-align: middle;
      }
      @media (min-width: 768px) {
        .bd-placeholder-img-lg {
          font-size: 3.5rem;
        }
      }
      body {
        background-color: #f8f9fa;
      }
      .form-signin {
        max-width: 330px;
        padding: 15px;
        margin: auto;
      }
      .form-signin .form-control {
        position: relative;
        box-sizing: border-box;
        height: auto;
        padding: 10px;
        font-size: 16px;
      }
      .form-signin .form-control:focus {
        z-index: 2;
      }
      .form-signin input[type="file"] {
        margin-bottom: 10px;
      }
      .form-signin button {
        background-color: #007bff;
      }

```



```

</style>
<link rel="stylesheet" href="/static/style.css">
<title>Object Detection | Дипломна робота</title>
</head>
<body class="text-center">
  <div class="container">
    <div class="header">
      <h1 class="display-4">Розпізнавання тварин на зображеннях</h1>
      <p class="lead">Завантажте зображення щоб знайти на ньому
тварин</p>
    </div>
    <div class="row justify-content-center">
      <div class="col-md-12">
        <form class="form-signin" method="post"
enctype="multipart/form-data">
          <div class="custom-file mb-3">
            <input type="file" name="file" class="custom-file-input"
id="inputfile" required>
            <label class="custom-file-label" for="inputfile">Обрати
файл</label>
          </div>
          <button class="btn btn-lg btn-primary btn-block"
type="submit">Обробити</button>
          <p class="mt-5 mb-3 text-muted">Дипломна робота</p>
        </form>
      </div>
    </div>
    </div>
    <script src="//code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTElPi6jizo"
crossorigin="anonymous"></script>
    <script
src="//cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.
js" integrity="sha384-
wHAiFfRlMfy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYulhA6ZOblgut"
crossorigin="anonymous"></script>
    <script
src="//stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js"
integrity="sha384-
B0UglyR+jN6CkvvICOB2joaf5I413gm9GU6Hclog6Ls7i6U/mkkaduKaBhlAXv9k"
crossorigin="anonymous"></script>
    <script>
      $('#inputfile').on('change', function() {
        let fileSize = this.files[0].size / 1024 / 1024; // in MB
        if (fileSize > 1) {
          $('#inputfile').val(null);
          alert('File is too big. Images more than 1MB are not
allowed.');
```

```
        if($.inArray(ext, ['jpg', 'jpeg']) === -1) {
            $("#inputfile").val(null);
            alert('Only JPEG/JPG files are allowed.');
```

```
        } else {
            // Show selected file name
            $(this).next('.custom-file-label').html($(this).val().split('\\').pop());
        }
    });
</script>
</body>
</html>
```

Лістинг А.2 – Розмітка головної сторінки

АНОТАЦІЯ

Лащук М. Т – **Реалізація розпізнавання об'єктів на зображеннях** –
Рукопис.

Кваліфікаційна робота за спеціальністю 122 Комп'ютерні науки. -
Волинський національний університет імені Лесі Українки, Луцьк. - 2024р.

Робота присвячена розробці програмного забезпечення для розпізнавання об'єктів на зображеннях за допомогою методів комп'ютерного зору та глибокого навчання. Розглянуто актуальність таких систем для автоматизації різних процесів у промисловості, безпеці, медицини та інших сферах. Мета роботи: реалізація системи розпізнавання об'єктів на зображеннях, яка використовує алгоритми глибокого навчання для забезпечення високої точності та ефективності. Завдання включають огляд сучасних методів розпізнавання об'єктів на зображеннях, тренування моделей глибокого навчання, реалізацію програмного забезпечення в середовищі PyCharm, оцінку точності та продуктивності розробленої системи.

Ключові слова: розпізнавання об'єктів, глибоке навчання, CNN, TensorFlow, PyCharm, YOLO, комп'ютерний зір, автоматизація.