Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

# КВАЛІФІКАЦІЙНА РОБОТА
# Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

**РОЗРОБЛЕННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ ТА АНАЛІЗУ СТАНУ ВЕЛИКОЇ РОГАТОЇ ХУДОБИ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ**
(тема)


Виконав:
здобувач 4 року навчання,
групи ІТІНФ-22-1
Потапов В. О.
(прізвище, ініціали)


Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Любченко В. А.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2026 р.

# ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Потапову Владиславу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення та реалізація системи моніторингу та аналізу стану великої рогатої худоби з використанням методів машинного навчання

затверджена наказом університету від 26 травня 2026 року № 435Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 19 червня 2026 р.
3. Вихідні дані до роботи науково-методична та науково-технічна література, дані інтернет-мережі, бібліотеки OpenCV, Ultralytics YOLO, набори даних для навчання моделей компёютерного зору, програмні засоби Python, Django, PostgreSQL, Redis та Docker.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз предметної області та існуючих систем моніторингу великої рогатої худоби.
2. Дослідження та вибір моделей комп'ютерного зору.
3. Огляд методів оцінки ваги великої рогатої худоби.
4. Розробка та реалізація програмної системи аналізу відео.
5. Тестування та оцінка ефективності розробленої системи.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми моніторингу великої рогатої худоби, аналіз існуючих методів оцінки ваги, постановка задачі, порівняння досліджуваних моделей та отриманих результатів, архітектура системи, структура бази даних, інтерфейс користувача, висновки та перспективи розвитку.

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

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

**КАЛЕНДАРНИЙ ПЛАН**

| № з/п | Назва етапів роботи | Строк / терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Отримання завдання на кваліфікаційну роботу | 13.04.2026 | |
| 2 | Аналіз завдання, підбір літератури | 14.04.26-17.04.26 | |
| 3 | Аналіз літератури з проблеми, що вирішується | 18.04.26-22.04.26 | |
| 4 | Аналіз існуючих методів розпізнавання | 23.04.26-25.04.26 | |
| 5 | Формування кроків вирішення проблеми | 26.04.26-28.04.26 | |
| 6 | Програмна реалізація | 26.04.26-15.05.26 | |
| 7 | Аналіз отриманих результатів | 16.05.26-19.05.26 | |
| 8 | Оформлення пояснювальної записки | 20.05.26-09.06.26 | |
| 9 | Перевірка на нормоконтроль | 10.06.26-19.06.26 | |
| 10 | Перевірка на плагіат | 11.06.26-30.06.26 | |
| 11 | Рецензування | 20.06.26-30.06.26 | |
| 12 | Підготовка презентації та доповіді | 20.06.26-30.06.26 | |
| 13 | Занесення роботи в електронний архів | 30.06.26-09.07.26 | |
| 14 | Попередній захист кваліфікаційної роботи | 30.06.26-09.07.26 | |

Дата видачі завдання 13 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Любченко В. А.
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 74 с., 10 табл., 23 рис., 2 дод., 34 джерело.

ВІДЕОАНАЛІЗ, КОМП'ЮТЕРНИЙ ЗІР, МАШИННЕ НАВЧАННЯ, МОНІТОРИНГ ВЕЛИКОЇ РОГАТОЇ ХУДОБИ, ОЦІНКА ВАГИ, DJANGO, LLM, PYTHON, TRACKING, YOLO.

Об'єктом роботи є система автоматизованого моніторингу великої рогатої худоби з використанням методів комп'ютерного зору та машинного навчання.

Метою роботи є розробка програмної системи, яка дозволяє виконувати виявлення тварин у відеопотоці, відстеження окремих особин, визначення їх породи та оцінку ваги.

Під час роботи використано моделі комп'ютерного зору, метод Клювера-Штрауха, модель XGBoost та мультимодальні великі мовні моделі.

Практична цінність роботи полягає у створенні системи, що дозволяє автоматизувати процес моніторингу тварин без використання контактних вимірювальних засобів.

Область застосування: фермерські господарства, системи автоматизованого моніторингу тварин, агропромисловий комплекс.

VIDEO ANALYSIS, COMPUTER VISION, MACHINE LEARNING, CATTLE MONITORING, WEIGHT ESTIMATION, DJANGO, LLM, PYTHON, TRACKING, YOLO.

The object of the work is an automated cattle monitoring system using computer vision and machine learning methods.

The goal of the work is to develop a software system that allows for animal detection in a video stream, tracking of individual individuals, determining their breed, and estimating weight.

The work used computer vision models, the Kluwer-Strauch method, XGBoost model, and multimodal large language models.

The practical value of the work lies in the creation of a system that allows you to automate the process of monitoring animals without the use of contact measuring devices.

Scope: farms, automated animal monitoring systems, agro-industrial complex.

# ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів ..... 7
Вступ ..... 8
1 Огляд та аналіз предметної області ..... 8
1.1 Загальний стан питання ..... 9
1.2 Розвиток галузі ..... 14
1.2.1 Прогрес у сфері комп'ютерного зору ..... 14
1.2.2 Тенденції розвитку моніторингу великої рогатої худоби ..... 17
1.3 Огляд існуючих застосунків ..... 19
1.4 Постановка задачі ..... 22
2 Аналіз методів комп'ютерного зору та оцінки ваги для розробки алгоритму моніторингу стану великої рогатої худоби ..... 23
2.1 Моделі сегментації комп'ютерного зору ..... 24
2.1.1 Архітектури сімейства U-Net (U-Net та U-Net++) ..... 24
2.1.2 Моделі сімейства YOLO для сегментації (YOLO-Seg) ..... 25
2.2 Моделі комп'ютерного зору для пошуку ключових точок ..... 29
2.3 Моделі для вирішення задачі класифікації породи ..... 31
2.3.1 EffitientNet ..... 31
2.3.2 YOLO-cls ..... 31
2.3.3 DeiT ..... 32
2.3.4 Swin Transformer ..... 32
2.3.5 ConvNeXt ..... 33
2.3.6 Порвіняння ..... 33
2.4 Методи визначення ваги великої рогатої худоби з використанням комп'ютерного зору ..... 36
2.4.1 Класчний метод Клювер-Штрауха ..... 37
2.4.2 Метод з використанням машинного навчання ..... 38
2.4.3 LLM ..... 39
2.4.4 Використання комп'ютерного зору та виділення ознак ..... 41

2.5 Висновки на основі результатів ..... 42
3 Розробка системи аналізу відео для оцінки ваги корів ..... 45
3.1 Розробка серверної частини ..... 45
3.1.1 Серверні технології та backend-архітектура ..... 47
3.1.2 Технології комп'ютерного зору та штучного інтелекту ..... 50
3.1.3 Технології контейнеризації та розгортання ..... 53
3.2 Клієнтський застосунок ..... 54
3.3 Алгоритм та демонстрація роботи ..... 58
Висновки ..... 63
Перелік джерел посилання ..... 65
Додаток А Приклади промптів та отриманих результатів ..... 69
Додаток Б Файли для налаштовування Docker-контейнеру ..... 71

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ВРХ – велика рогата худоба

кг – кілограм

мс – мілісекунда

CNN – Convolutional Neural Network (згорткова нейронна мережа)

DeiT – Data-efficient Image Transformer

Docker – платформа контейнеризації для розгортання та запуску програмного забезпечення в ізольованому середовищі

GPU – Graphics Processing Unit (графічний процесор)

IoU – Intersection over Union (метрика перетину областей)

LLM – Large Language Model (велика мовна модель)

MBConv – Mobile Inverted Bottleneck Convolution (спеціальний будівельний блок в архітектурі нейромереж, який застосовується для аналізу зображень)

Precision – метрика точності класифікації

ReID – Re-Identification (повторна ідентифікація особи)

Recall – метрика повноти класифікації

Top-1 – метрика, за якої правильний клас має найбільшу ймовірність

Top-5 – метрика, за якої правильний клас входить до п’яти найбільш імовірних

U-Net – архітектура нейронної мережі для задач сегментації зображень

ViT – Vision Transformer (Зоровий трансформер)

YOLO – You Only Look Once (серія моделей для вирішення основних задач комп’ютерного зору у реальному часі)

XGBoost – eXtreme Gradient Boosting (програмна бібліотека для реалізації алгоритмів машинного навчання у рамках інфраструктури Gradient Boosting)

# ВСТУП

Сучасний розвиток інформаційних технологій та методів штучного інтелекту суттєво впливає на автоматизацію процесів у сільському господарстві. Одним із найбільш перспективних напрямків є використання систем комп'ютерного зору та машинного навчання для моніторингу стану великої рогатої худоби. Такі системи дозволяють автоматизувати процеси спостереження за тваринами, зменшити вплив людського фактора та підвищити ефективність управління фермерськими господарствами.

Важливим показником стану великої рогатої худоби є вага тварини, оскільки вона використовується для оцінки здоров'я, контролю розвитку, визначення ефективності годування та прогнозування продуктивності. Традиційні методи визначення ваги потребують використання спеціалізованого обладнання або безпосереднього контакту з твариною, що може бути складним у реальних умовах фермерських господарств. У зв'язку з цим актуальним є створення автоматизованих систем оцінки ваги на основі аналізу зображень та відеоданих.

Актуальність роботи полягає у необхідності створення програмної системи автоматизованого визначення ваги великої рогатої худоби за відеоданими з використанням методів комп'ютерного зору та машинного навчання. Використання подібних систем дозволяє підвищити точність моніторингу стану тварин, знизити трудові витрати та забезпечити більш ефективне ведення фермерського господарства.

Практична цінність роботи полягає у можливості використання розробленої системи у фермерських господарствах для автоматизованого контролю стану великої рогатої худоби без необхідності використання дорогого спеціалізованого обладнання.

# 1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Загальний стан питання

У сучасних умовах розвитку агропромислового комплексу особливо важливо підвищувати ефективність управління господарством [1]. Одним із ключових напрямків є автоматизований моніторинг великої рогатої худоби, який дозволяє отримувати актуальну інформацію про тварин у реальному часі.

Моніторинг стану великої рогатої худоби є важливою складовою ефективного ведення тваринництва. Традиційні методи контролю базуються на безпосередньому спостереженні за тваринами або використанні спеціалізованих вимірювальних засобів. Незважаючи на розвиток сучасних технологій, такі підходи досі широко застосовуються у фермерських господарствах [1].

Найпростішим і найбільш поширеним методом є візуальний контроль, який здійснюється працівниками господарства. Оцінка стану тварин проводиться на основі зовнішніх ознак, таких як поведінка, рухова активність, апетит та загальний вигляд. Перевагою даного підходу є його простота та відсутність необхідності у використанні додаткового обладнання. Крім того, досвідчений працівник може комплексно оцінити стан тварини, враховуючи різні непрямі ознаки. Водночас цей метод має суттєві недоліки, зокрема суб’єктивність оцінювання, залежність від кваліфікації персоналу та неможливість забезпечення безперервного моніторингу. При великій кількості тварин ефективність такого підходу значно знижується через високі витрати часу.

Для точного визначення маси тварин застосовуються вагові платформи або спеціалізовані ваги (рис. 1.1). У цьому випадку тварина фізично розміщується на ваговому пристрої, після чого фіксується її маса. Основною перевагою цього методу є висока точність, оскільки результат вимірювання є прямим і не потребує додаткових обчислень.

Рисунок 1.1 – Ваги для зважування великих тварин

Основною перевагою цього методу є висока точність, оскільки результат вимірювання є прямим і не потребує додаткових обчислень. Проте застосування такого підходу пов’язане з рядом обмежень. Зокрема, необхідність фізичного контакту з твариною може викликати стрес, що негативно впливає на її стан. Крім того, процес зважування є трудомістким і потребує залучення персоналу, що ускладнює його регулярне використання, особливо у великих господарствах.

Іншим традиційним підходом є визначення ваги на основі геометричних параметрів тіла, таких як довжина тулуба та обхват грудної клітки (рис. 1.2). Отримані значення використовуються у емпіричних формулах або таблиць для розрахунку маси тварини. Існує декілька методів які використовують подібні значення. Тут можна виділити відносну простоту реалізації та відсутність необхідності у дорогому обладнанні.

Водночас точність результатів значною мірою залежить від правильності виконання вимірювань. Ручний характер збору даних призводить до виникнення похибок, а також ускладнює застосування методу у випадку великої кількості тварин.

Рисунок 1.2 – Приклад вимірювання стрічкою

Додатково слід враховувати, що емпіричні формули часто залежать від породи та індивідуальних особливостей тварини.

Сучасним напрямом розвитку традиційних підходів є використання носимих сенсорів, які кріпляться до тіла тварини та забезпечують автоматичний збір даних (рис. 1.3). До таких пристроїв належать акселерометри, GPS-трекери та інші датчики активності. Вони дають можливість безперервного моніторингу та отримання об’єктивних даних без участі людини. Це дозволяє більш точно відстежувати поведінку та стан тварин у динаміці. Крім того, накопичення даних за тривалий період часу дає змогу виявляти зміни у поведінці тварин та своєчасно реагувати на можливі проблеми зі здоров’ям або умовами утримання [1].

Разом з тим, такі рішення потребують значних фінансових витрат на придбання та обслуговування обладнання. Крім того, наявність фізичних пристроїв на тілі тварини може викликати дискомфорт, а також існує ризик їх пошкодження або втрати.

Рисунок 1.3 – Приклад кріплення сенсору

Разом з тим, такі рішення потребують значних фінансових витрат на придбання та обслуговування обладнання. Крім того, наявність фізичних пристроїв на тілі тварини може викликати дискомфорт, а також існує ризик їх пошкодження або втрати.

Традиційні методи моніторингу великої рогатої худоби, незважаючи на свою практичну цінність, мають ряд суттєвих обмежень. Більшість із них є трудомісткими, залежать від людського фактору та не забезпечують можливості безперервного автоматизованого контролю. Крім того, їх застосування у великих господарствах є ускладненим через обмежену масштабованість.

Комп’ютерний зір дозволяє здійснювати безконтактний аналіз відеоданих, що забезпечує можливість автоматичного контролю великої кількості тварин одночасно. На відміну від традиційних підходів, такий метод не потребує фізичного контакту з твариною та дозволяє організувати безперервний моніторинг у режимі реального часу.

Використання методів комп’ютерного зору дозволяє вирішити ряд проблем, притаманних традиційним підходам. Зокрема, усувається суб’єктивність оцінювання, оскільки аналіз виконується автоматично на

основі алгоритмів машинного навчання. Крім того, значно зменшується трудомісткість процесу, оскільки відпадає необхідність постійної участі персоналу у спостереженні та вимірюваннях.

Ще однією важливою перевагою є можливість масштабування системи. На відміну від ручного контролю, комп'ютерний зір дозволяє одночасно аналізувати велику кількість тварин без суттєвого зниження ефективності. Це особливо актуально для великих фермерських господарств.

Також комп'ютерний зір забезпечує можливість отримання додаткових характеристик тварин, які складно або неможливо визначити традиційними методами. Зокрема, аналіз відеоданих дозволяє оцінювати поведінкові характеристики, траєкторію руху, активність, а також непрямі ознаки стану здоров'я.

У сучасних дослідженнях можна виділити такі основні завдання:

− детекція тварин на зображеннях та відео;

− відстеження окремих особин у часі;

− оцінка фізичних та геометричних характеристик, зокрема пози тварин;

− аналіз таких параметрів, як маса або поведінкові особливості.

Ключовим елементом таких систем є детекція об'єктів, яка дозволяє локалізувати тварин у кадрі. Для цього використовуються сучасні моделі глибокого навчання, що забезпечують високу точність і швидкість обробки даних у реальному часі [2, 3]. На основі результатів детекції будуються більш складні етапи обробки, такі як відстеження об'єктів, аналіз пози та оцінка характеристик.

Разом з тим, використання комп'ютерного зору також пов'язане з певними труднощами. До них належать залежність якості роботи від умов освітлення, можливі перекриття тварин у кадрі, варіації ракурсу зйомки та обмежена кількість спеціалізованих наборів даних для навчання моделей. Крім того, складність сцени та динамічний характер відеопотоку можуть призводити до помилок детекції та відстеження.

Таким чином, застосування комп’ютерного зору дозволяє суттєво підвищити рівень автоматизації та ефективності моніторингу великої рогатої худоби, проте потребує розробки комплексних систем, що враховують специфіку задачі та забезпечують достатню точність у реальних умовах експлуатації.

## 1.2 Розвиток галузі

Розвиток комп’ютерного зору пов’язаний із переходом від класичних методів обробки зображень до моделей глибокого навчання. Сучасні підходи базуються на використанні нейронних мереж, які здатні автоматично виділяти ознаки та ефективно аналізувати візуальні дані.

У задачах моніторингу великої рогатої худоби це дозволило створити системи, що автоматично виявляють тварин, відстежують їх та визначають додаткові характеристики.

### 1.2.1 Прогрес у сфері комп’ютерного зору

Розвиток комп’ютерного зору є одним із ключових факторів, що забезпечують створення сучасних інтелектуальних систем аналізу відеоданих. Протягом останніх десятиліть відбувся значний перехід від класичних алгоритмів обробки зображень до методів глибокого навчання, що дозволило суттєво підвищити точність і надійність розпізнавання об’єктів.

Ранні підходи базувалися на використанні ручно заданих ознак, таких як градієнти, текстурні характеристики та контури об’єктів. До таких методів належать алгоритми SIFT, HOG та Haar-каскади. Вони дозволяли вирішувати окремі задачі розпізнавання, проте не були достатньо універсальними та значно залежали від умов освітлення, ракурсу зйомки та шумів.

Прорив у розвитку відбувся з появою згорткових нейронних мереж, які здатні автоматично виділяти релевантні ознаки з зображень [1]. Використання глибоких нейронних мереж дозволило значно покращити результати у задачах класифікації, детекції та сегментації об’єктів.

Одним із перших успішних застосувань глибокого навчання стала модель AlexNet, яка продемонструвала значний приріст точності на задачі класифікації зображень [4]. Подальший розвиток отримали архітектури ResNet, що дозволяють будувати значно глибші мережі за рахунок використання залишкових з’єднань.

У задачах детекції об’єктів сформувалося два основних підходи: двоетапні та одноетапні детектори.

Двоетапні детектори, такі як Faster R-CNN, спочатку генерують області-кандидати, а потім класифікують їх. Такий підхід забезпечує високу точність, але має обмежену швидкодію, що ускладнює його використання в реальному часі [5].

Одноетапні детектори, зокрема сімейство YOLO, виконують детекцію об’єктів за один прохід нейронної мережі. Це дозволяє досягти високої швидкості обробки відео, що є критично важливим для систем відеоаналізу. Сучасні версії YOLO демонструють баланс між точністю та продуктивністю, що робить їх оптимальним вибором для практичних застосувань [6, 3, 7].

Подальшим етапом розвитку стали трансформерні архітектури, які використовують механізм уваги для моделювання взаємозв’язків між об’єктами у зображенні [8]. Такі моделі дозволяють більш ефективно працювати зі складними сценами, де об’єкти можуть перекриватися або мати різні масштаби.

Окрім детекції, важливим напрямом є багатокласова класифікація об’єктів, що дозволяє визначати належність об’єкта до певної категорії, наприклад породи тварини. Для цього використовуються CNN-архітектури або їх модифікації, що навчаються на великих наборах зображень.

Значний розвиток отримали також методи відстеження об’єктів, які дозволяють аналізувати динаміку руху об’єктів у відеопотоці. Сучасні алгоритми multi-object tracking поєднують результати детекції з алгоритмами асоціації об’єктів між кадрами [9]. Вони використовують інформацію про положення, швидкість руху та візуальні ознаки об’єктів.

Особливу роль відіграє використання методів ReID, які дозволяють повторно ідентифікувати об’єкти після їх тимчасового зникнення з кадру [10, 11]. Це є важливим для задач моніторингу великої рогатої худоби, де тварини можуть перекривати одна одну.

Сучасні системи також активно використовують апаратне прискорення, зокрема графічні процесори та спеціалізовані обчислювальні пристрої. Це дозволяє виконувати обробку відео у режимі реального часу навіть при високій роздільній здатності.

Таким чином, сучасний прогрес у сфері комп’ютерного зору характеризується переходом до універсальних моделей глибокого навчання, здатних вирішувати широкий спектр задач, а також інтеграцією різних підходів у єдині системи аналізу.

У рамках сучасних підходів до побудови систем відеоаналізу все частіше використовується модульна архітектура, яка передбачає послідовне виконання декількох етапів обробки даних.

Типовий конвеєр обробки відео включає:

− детекцію об’єктів – визначення положення тварин у кадрі за допомогою моделей типу YOLO;

− відстеження (tracking) – призначення унікального ідентифікатора кожній тварині та відстеження її між кадрами;

− класифікація – визначення окремих характеристик кожного об’єкту, наприклад породи;

− аналіз форми об’єкта – отримання маски або ключових точок тіла;

− обчислення геометричних характеристик – визначення довжини тіла, ширини та інших параметрів;

− обробка результатів – інтеграція та інтерпретація результатів, отриманих на попередніх етапах, з метою визначення прикладних характеристик тварин, наприклад ваги або поведінки.

Саме такий підхід дозволяє як забезпечити баланс між точністю та швидкодією системи, так і отримані результати агрегувати та зберегти у структурованому вигляді, що дозволяє виконувати подальший аналіз, моніторинг та інтеграцію з іншими інформаційними системами. Це дозволяє перейти від низькорівневих результатів моделей (bounding box, keypoints, masks) до високорівневих показників, що мають практичну цінність у тваринництві.

### 1.2.2 Тенденції розвитку моніторингу великої рогатої худоби

Сучасні системи моніторингу великої рогатої худоби розвиваються у напрямі підвищення рівня автоматизації, точності аналізу та інтеграції з іншими інформаційними системами фермерських господарств.

Однією з ключових тенденцій є перехід від ізольованих рішень до комплексних систем, що поєднують декілька задач комп’ютерного зору. Якщо раніше системи були орієнтовані лише на детекцію тварин або аналіз окремих параметрів, то сучасні підходи передбачають одночасне виконання детекції, відстеження, класифікації та оцінки тварин.

Важливою тенденцією є використання відео як основного джерела даних. На відміну від статичних зображень, відеопотік дозволяє отримувати інформацію про поведінку та переміщення тварин у просторі. Це відкриває можливість для більш глибокого аналізу, зокрема виявлення змін у стані тварин, як показано на рисунку 1.4. Завдяки цьому можна одразу визначати нетипову поведінку, зниження активності або інші ознаки, що можуть свідчити про погіршення стану здоров’я худоби.

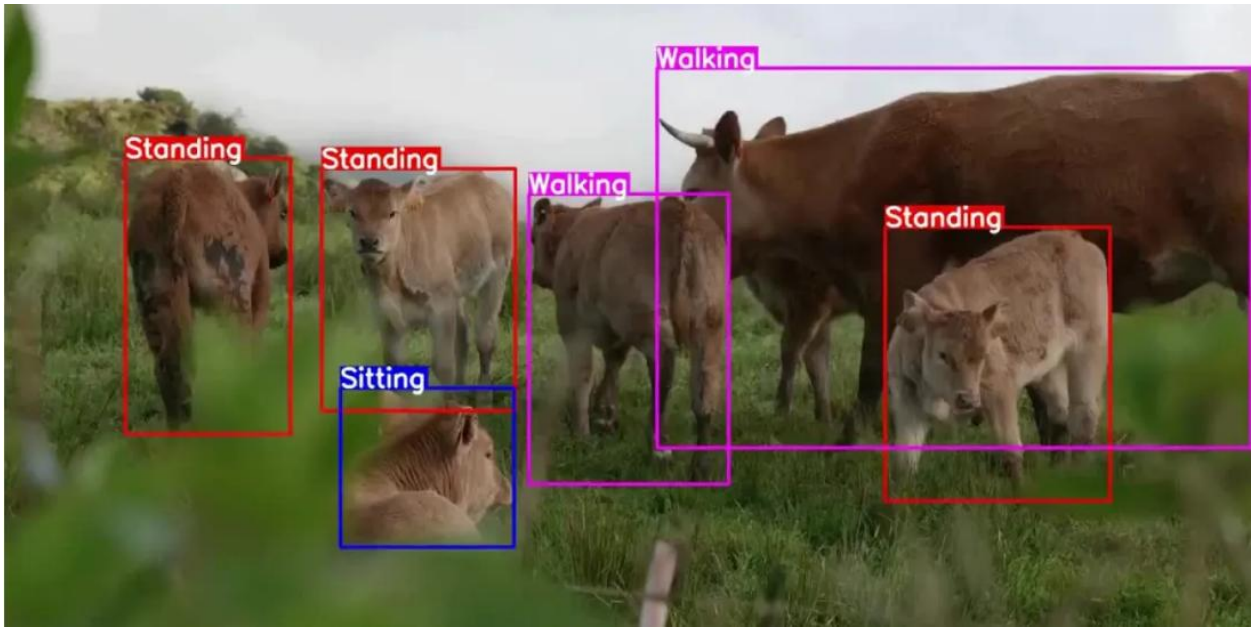



Рисунок 1.4 – Моніторинг поведінки та положення корів за допомогою Ultralytics YOLOv8 [7]

Значну роль відіграє розвиток алгоритмів multi-object tracking, які дозволяють відстежувати велику кількість тварин одночасно [10, 9, 11]. Такі алгоритми забезпечують збереження ідентичності кожної тварини у відеопотоці, що є необхідною умовою для накопичення історичних даних та подальшого аналізу.

Ще однією тенденцією є використання моделей глибокого навчання, спеціально адаптованих під задачі тваринництва [12, 2, 13]. Це включає навчання моделей на спеціалізованих наборах даних, що містять зображення великої рогатої худоби в реальних умовах. Такий підхід дозволяє підвищити точність роботи систем у порівнянні з універсальними моделями.

Зростає інтерес до побудови систем, здатних працювати у режимі реального часу [12]. Це досягається за рахунок використання ефективних моделей детекції та оптимізації процесу інференсу. Реальний час є критично важливим для практичного використання систем у фермерських господарствах для свєчасного виявлення проблем.

Також можна побачити зниження вартості впровадження систем. Використання відкритого програмного забезпечення та доступних апаратних

рішень дозволяє не тільки краще проводити певні дослідження, а й впроваджувати системи моніторингу навіть у невеликих господарствах.

Зростає увага і до питань масштабованості та адаптивності систем. Сучасні рішення повинні легко підлаштовуватися під різні умови утримання тварин, змін освітлення та конфігурації середовища.

Ще одним напрямом є підвищення точності оцінки фізичних параметрів тварин, зокрема вгодованості. Це пов’язано з розвитком методів аналізу зображень та машинного навчання, що дозволяють отримувати більш точні оцінки на основі візуальних даних.

Таким чином, сучасні тенденції розвитку систем моніторингу великої рогатої худоби спрямовані на створення комплексних, автоматизованих та масштабованих рішень, що забезпечують ефективний аналіз даних у реальному часі.

## 1.3 Огляд існуючих застосунків

У сучасних умовах розвитку точного тваринництва спостерігається зростаючий інтерес до використання інтелектуальних систем моніторингу великої рогатої худоби. На ринку та у наукових дослідженнях представлено ряд програмних рішень, що використовують методи комп’ютерного зору та машинного навчання для аналізу стану тварин.

Одним із відомих прикладів є система CattleEye, яка використовує відеокамери та алгоритми штучного інтелекту для автоматичного моніторингу поведінки корів (рис. 1.5). Система аналізує відеопотік, виявляє тварин та оцінює їхню рухову активність, зокрема для виявлення кульгавості, а також робить оцінку вгодованості (Body Condition Score, BCS) [14]. Отримані результати можуть використовуватись для оперативного контролю стану тварин та підтримки прийняття рішень у фермерських господарствах.

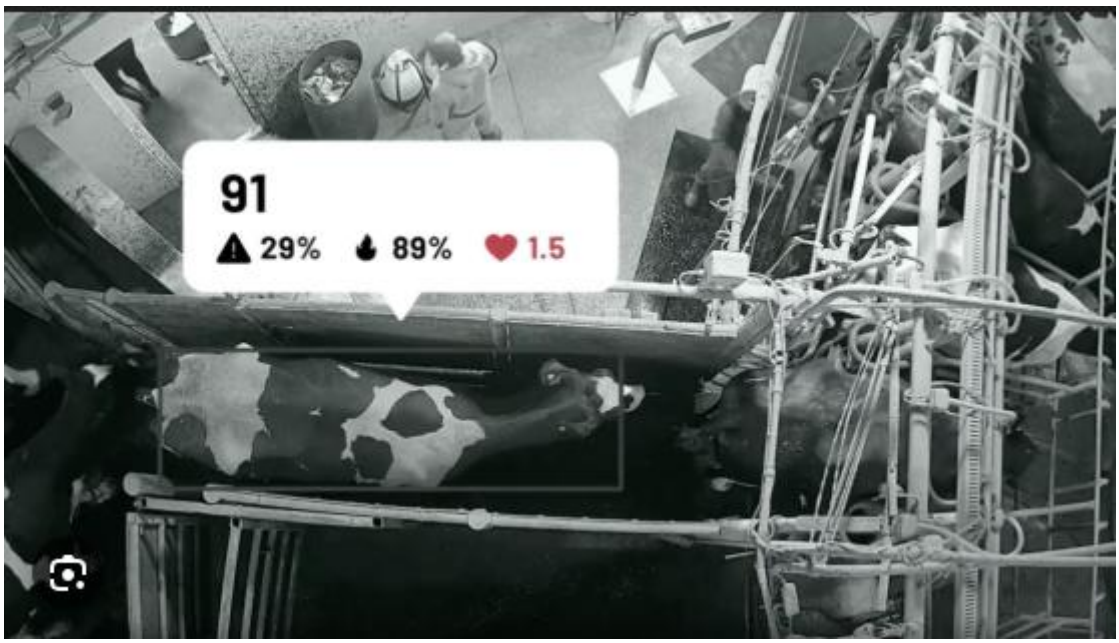



Рисунок 1.5 – Демонстрація роботи CattleEye

Основною перевагою такого підходу є можливість безперервного збору даних без участі людини, що значно підвищує ефективність моніторингу.

Іншим прикладом є рішення, розроблені організацією Carnot CEA LIST, які орієнтовані на моніторинг стану тварин у режимі реального часу. Такі системи використовують поєднання комп'ютерного зору та сенсорних технологій для збору та аналізу даних. Вони забезпечують високу точність оцінки стану тварин, однак їх впровадження часто потребує складної інфраструктури та значних фінансових витрат.

У наукових дослідженнях також представлено численні прототипи систем, що вирішують окремі задачі моніторингу. Зокрема, системи детекції та підрахунку тварин у відеопотоці, моделі класифікації поведінки, алгоритми виявлення аномалій у поведінці та рішення для оцінки стану здоров'я на основі непрямих ознак [2].

Багато з цих систем демонструють високу точність у лабораторних умовах, однак їх застосування у реальних фермерських господарствах ускладнюється рядом факторів. До основних обмежень існуючих рішень можна віднести орієнтацію на вирішення лише однієї задачі, відсутність інтеграції кількох компонентів у єдину систему, високу обчислювальну

складність окремих методів, залежність від умов зйомки та складність масштабування та впровадження у виробничих умовах.

Окрему категорію становлять системи, що використовують носимі сенсори у поєднанні з програмним аналізом даних. Такі рішення дозволяють отримувати точні фізіологічні показники, проте мають недоліки у вигляді інвазивності та необхідності обслуговування обладнання.

Слід також зазначити, що деякі сучасні підходи орієнтовані на використання складних методів сегментації зображень для більш точного виділення тварин [8]. Проте у практичних застосуваннях такі методи не завжди виправдані через високу обчислювальну вартість та складність інтеграції у системи реального часу. Додатково виникають труднощі з обробкою відеоданих у складних умовах, таких як нерівномірне освітлення або перекриття об’єктів, що може знижувати ефективність.

Таким чином, аналіз існуючих застосунків показує, що, незважаючи на значний прогрес у сфері комп’ютерного зору, більшість рішень мають обмежену функціональність або складність впровадження. Це зумовлює необхідність розробки більш універсальних та ефективних систем, які поєднують декілька задач аналізу відеоданих у межах єдиної автоматизованої послідовності дій. Важливим аспектом є також забезпечення стабільної роботи таких систем у реальних умовах експлуатації, що потребує врахування різноманітних зовнішніх факторів.

З урахуванням сучасних тенденцій доцільним є використання підходу, що базується на поєднанні методів детекції, оцінки поз, відстеження об’єктів та класифікації. Такий підхід дозволяє забезпечити достатню точність аналізу при збереженні високої швидкодії та придатності до практичного застосування. Крім того, модульна структура подібних систем спрощує їх адаптацію до різних умов експлуатації та розширення функціональних можливостей у майбутньому.

## 1.4 Постановка задачі

На основі проведеного аналізу предметної області, сучасних методів комп’ютерного зору та існуючих рішень у сфері моніторингу великої рогатої худоби можна зробити висновок про необхідність розробки комплексної автоматизованої системи, що забезпечує ефективний аналіз відеоданих у реальних умовах фермерського господарства [1].

Об’єктом роботи є система автоматизованого моніторингу великої рогатої худоби з використанням методів комп’ютерного зору та машинного навчання.

Метою роботи є розробка програмної системи, яка дозволяє виконувати виявлення тварин у відеопотоці, відстеження окремих особин, визначення їх породи та оцінку ваги.

Для досягнення поставленої мети необхідно вирішити такі задачі:

− обрати та обґрунтувати метод детекції тварин у відеопотоці;

− реалізувати алгоритм відстеження (tracking) окремих особин;

− розробити модель класифікації для визначення породи тварин [4];

− обрати та реалізувати метод оцінки ваги тварин на основі зображень;

− об’єднати окремі модулі у єдину програмну систему;

− провести експериментальну оцінку ефективності системи.

У якості вхідних даних системи розглядаються відеопотоки або окремі зображення з камер спостереження. Результатом роботи системи є структурована та записана в базу даних інформація, про виявлених тварин, яка включає їх ідентифікацію у часі (tracking ID), визначену породу та оцінену вагу. Отримані дані можуть бути використані для подальшого аналізу або інтеграції з іншими інформаційними системами фермерського господарства.

Таким чином, розроблювана система повинна забезпечувати достатню точність та швидкість для практичного застосування у фермерських господарствах.

# 2 АНАЛІЗ МЕТОДІВ КОМП'ЮТЕРНОГО ЗОРУ ТА ОЦІНКИ ВАГИ ДЛЯ РОЗРОБКИ АЛГОРИТМУ МОНІТОРИНГУ СТАНУ ВЕЛИКОЇ РОГАТОЇ ХУДОБИ

В роботі проаналізовано деклька сучасних методів комп'ютерного зору та машинного навчання для автоматизованого моніторингу стану великої рогатої худоби (ВРХ), а також вибір найбільш ефективних підходів для реалізації системи, що дозволяє:

− виконувати детекцію та відстеження тварин у відеопотоці;

− визначати породу кожної окремої особини;

− оцінювати масу тіла на основі отриманних характеристик.

Для досягнення поставленої мети було проведено низку досліджень, у сервісі Google Colab використовуючи графічниц процесор T4, сучасних архітектур нейронних мереж які застосовуються у задачах комп'ютерного зору.

Для навчання та порівняння моделей використовується набір даних Cattle Weight Detection Model + Dataset (12k~), розташований на платформі Kaggle [14, 16]. Він містить понад 12 000 зображень великої рогатої худоби. Цей набір даних включає:

− зображення тварин з боку;

− інструкції сегментації (маски);

− координати ключових точок;

− інформацію про реальну масу тварин.

Наявність розмітки робить даний набір придатним навчання як моделей сегментації, і моделей визначення ключових точок, але так як початковий формат масок придатний до навчання лише моделей класу Semantic Segmentation, було додатково зроблено копію у так званому yolo форматі, що дозволить зробити порівняння принципово різних моделей. Додатково набір зображень було розділено на тренувальну (2080) та валідаційну частину (520).

## 2.1 Моделі сегментації комп’ютерного зору

### 2.1.1 Архітектури сімейства U-Net (U-Net та U-Net++)

Моделі типу U-Net є класичним підходом до задач семантичної сегментації [17]. Вони добре працюють у випадках, коли необхідно точно виділити об’єкт на зображенні, навіть при обмеженій кількості даних. Архітектура U-Net базується на енкодер-декодерній структурі з пропусками (skip-connections), що дозволяє поєднувати контекстну інформацію та деталі високої роздільної здатності (рис. 2.1).

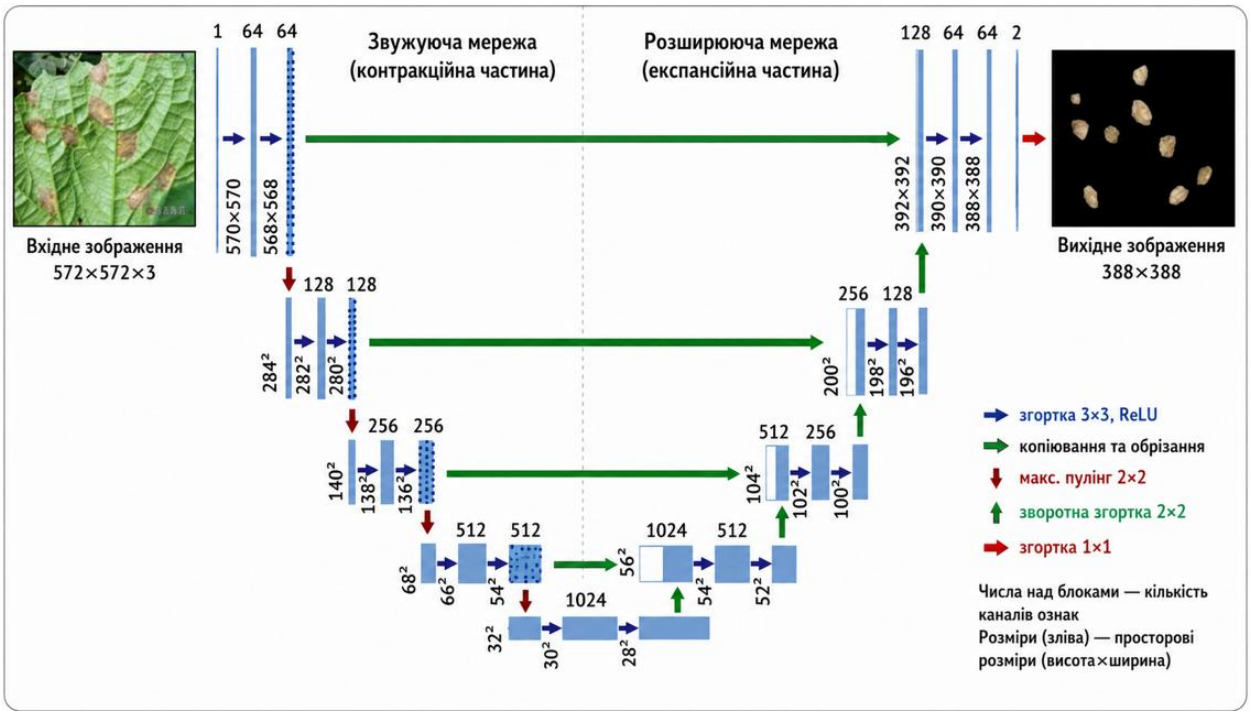



Рисунок 2.1 – Схема архітектури U-Net

Розширена версія U-Net++ використовує вкладені skip-з’єднання та більш складну структуру декодера, що дозволяє:

– покращити точність сегментації;

– зменшити втрати просторової інформації;

– краще узгоджувати ознаки між рівнями мережі.

У рамках роботи ці моделі будуть використовуватися як пряме порівняння семантичної сегментації із instance моделями. Результати роботи моделей представлено у наступних двох таблицях.

За результатами, наведеними у таблиці 2.1, модель U-Net++ забезпечує вищу точність сегментації за метриками Dice та mIoU. Водночас, вона має значно більший час інференсу порівняно з базовою U-Net. Однак такі показники швидкодії не дуже вражають коли йде мова про роботу з відео, або у реальному часі.

Таблиця 2.1 – Метрики досліждень сегментації моделей UNet

| **Модель** | **Dice** | **IoU** | **Inference, мс** |
|---|---|---|---|
| UNet | 0,971066782 | 0,9439009699 | 77,61654709 |
| UNet++ | 0,9790044671 | 0,9588940153 | 137,2770324 |

### 2.1.2 Моделі сімейства YOLO для сегментації (YOLO-Seg)

YOLO-підхід є одностадійним детектором об’єктів, який був розширений для задач instance segmentation [18]. На відміну від U-Net, YOLO-Seg одночасно:

− знаходить об’єкти на зображенні;

− виконує їх сегментацію.

У даній роботі YOLO-Seg буде застосовуватися для instance segmentation, що дозволяє виділяти кожну тварину окремо навіть у групових сценах і розглядатися як головний претиндент у продакшені.

З отриманих результатів (табл. 2.2) можна зробити висновок, що зі збільшенням швидкодії моделей спостерігається певне зниження точності моделей. Найшвидшими є легкі моделі, зокрема YOLOv8n-seg і YOLO11n-seg, які при цьому забезпечують доволі високі показники точності.

Серед усіх моделей найвищу точність сегментації показала YOLO26l-seg (90.94%), але з відносно високим часом інференсу. Водночас YOLOv8m-seg і YOLO26m-seg виглядають як найбільш збалансовані варіанти, поєднуючи високу точність і помірну швидкість. Тож, для поставленої задачі доцільно обрати моделі з оптимальним балансом між точністю та швидкістю.

Таблиця 2.2 – Метрики моделей Yolo-seg

| **Модель** | **Inference, мс** | **mAP50_95_mask, %** | **mAP50_95_box, %** | **Params, M** |
|---|---|---|---|---|
| YOLOv8n-seg | 5,18608505 | 90,28 | 98,48 | 3,4 |
| YOLOv8s-seg | 10,90994260 | 90,42 | 98,65 | 11,8 |
| YOLOv8m-seg | 22,29485167 | 90,44 | 98,85 | 27,3 |
| YOLOv8l-seg | 38,99494036 | 90,42 | 98,86 | 46,0 |
| YOLOv8x-seg | 55,80239422 | 89,79 | 98,78 | 71,8 |
| YOLO11n-seg | 5,94273129 | 89,72 | 98,39 | 2,9 |
| YOLO11s-seg | 10,65966197 | 90,03 | 98,43 | 10,1 |
| YOLO11m-seg | 27,15216654 | 90,11 | 98,59 | 22,4 |
| YOLO11l-seg | 32,16106830 | 89,90 | 98,43 | 27,6 |
| YOLO11x-seg | 57,57715183 | 89,59 | 98,41 | 62,1 |
| YOLO26n-seg | 6,42429806 | 90,19 | 98,36 | 2,7 |
| YOLO26s-seg | 11,37450836 | 90,69 | 98,15 | 10,4 |
| YOLO26m-seg | 28,33826926 | 90,50 | 98,62 | 23,6 |
| YOLO26l-seg | 34,70885295 | 90,94 | 97,41 | 28,0 |
| YOLO26x-seg | 65,68686028 | 90,80 | 98,54 | 62,8 |

Якщо порівнювати YOLO та UNet в сегментації, то спостерігається різниця не лише у числових метриках, а й у самій філософії архітектур. Моделі UNet орієнтовані на максимально точне відновлення форми об’єкта на піксельному рівні, що підтверджується високими значеннями точності (табл. 2.1).

У свою чергу, моделі сімейства YOLO-seg забезпечують значно кращу швидкодію, що особливо важливо для задач, пов’язаних із обробкою

відеопотоку або роботою в реальному часі. Хоча їх точність сегментації дещо поступається UNet-подібним моделям, різниця не є критичною для більшості практичних застосувань. При цьому YOLO додатково надає інформацію про обмежувальні рамки об’єктів, що розширює можливості використання.

Окремо виділимо версію YOLOE, яка архітектурно, на відміну від класичних моделей YOLO (рис. 2.2), має низку суттєвих відмінностей. Традиційні моделі сімейства YOLO (наприклад, YOLOv8, YOLO11, YOLO26) працюють в межах закритого словника класів [18]: вони можуть розпізнавати лише ті об’єкти, які були присутні в навчальному наборі даних. Класи у таких моделях жорстко зафіксовані на етапі навчання, а вихідний шар детектора має фіксовану кількість категорій. Це означає, що для додавання нових класів необхідно виконувати повторне донавчання моделі на відповідному наборі даних.

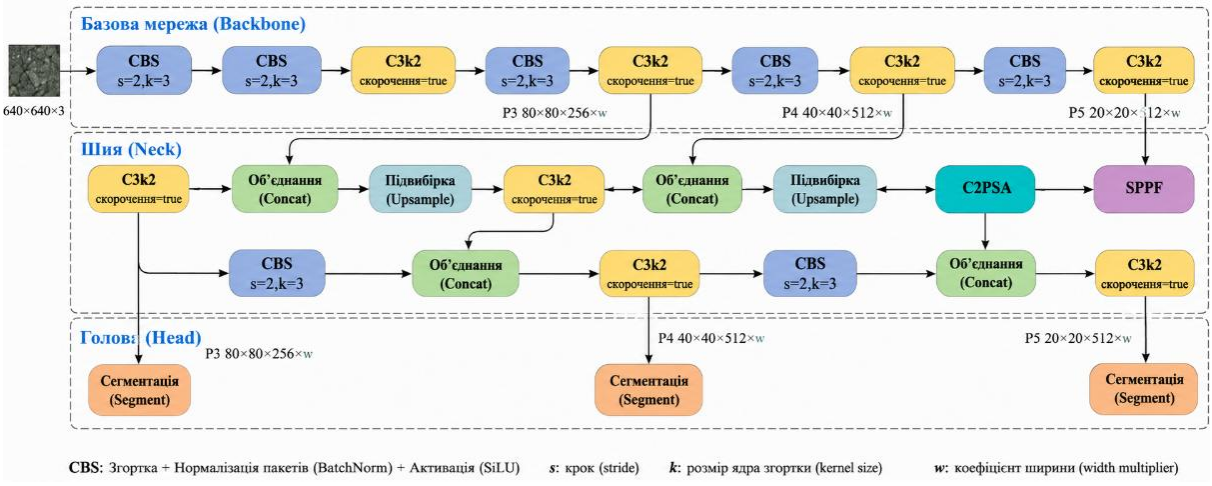



Рисунок 2.2 – Архітектура моделей YOLO11-seg

Натомість YOLOE реалізує підхід відкритого словника (open-vocabulary detection), що принципово змінює спосіб взаємодії з моделлю (рис. 2.3). Замість жорстко визначеного списку класів, YOLOE здатний використовувати механізми зіставлення візуальних ознак із семантичними представленнями, отриманими з текстових або візуальних підказок (prompts) [19]. Завдяки цьому модель здатна виявляти об’єкти, які не були явно представлені у тренувальних даних LVIS або Objects365.

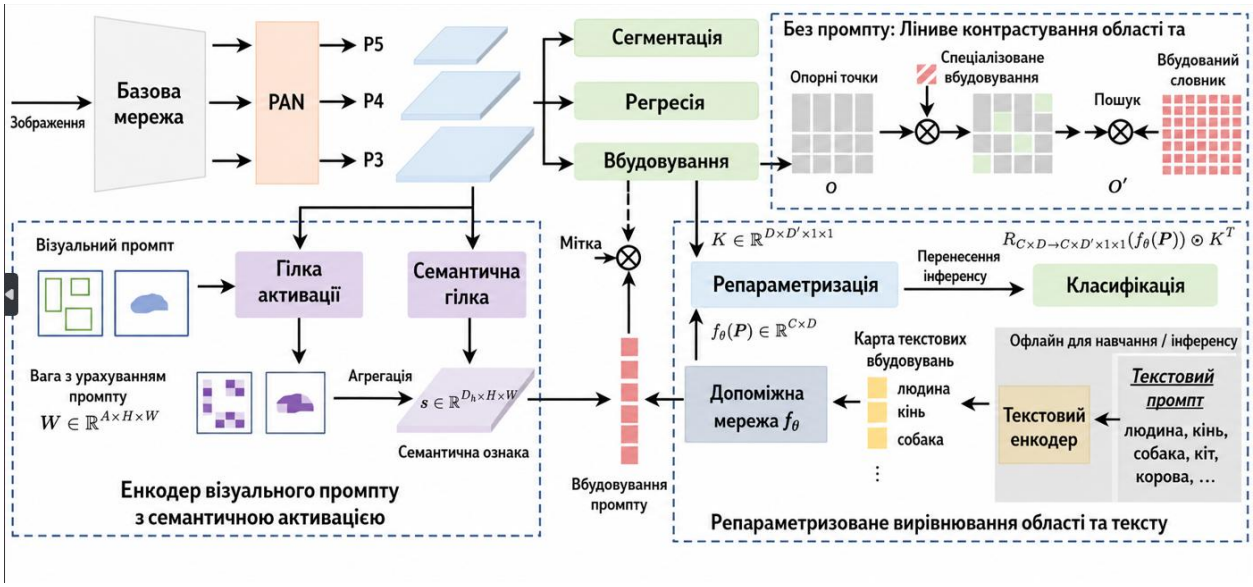



Рисунок 2.3 – Архітектура моделей YOLOE

Таким чином, YOLOE є більш гнучкою та універсальною моделлю порівняно з класичними YOLO-підходами, особливо в задачах, де перелік об’єктів є динамічним або заздалегідь невизначеним. Це робить її перспективною для застосувань у реальних умовах, де важко забезпечити повне покриття всіх можливих класів на етапі навчання. Дослідження буде проводитися як на донавчених версіях з приставкою PF (Prompt Free), так і базові з використанням текстових підказок.

Аналіз результатів, наведених у таблиці 2.3, підтверджує ефективність запропонованого підходу YOLOE, проте водночас демонструє суттєву залежність якості від способу використання підказок. Зокрема, донавчені моделі з приставкою показують значно кращі результати порівняно з базовими версіями: значення mAP50–95 для масок досягає 0.91–0.92, для обмежувальних рамок – до 0.98–0.99, а міра F1 наближається до 1.0, що свідчить про високу точність і стабільність виявлення.

У той же час базові моделі YOLOE, які використовують текстові підказки, демонструють помітно нижчі показники (зокрема, F1 у межах 0.69–0.73), що може бути пов’язано з залежністю від якості та релевантності сформульованих prompt’ів. Це вказує на те, що, попри концептуальну

гнучкість open-vocabulary підходу, його практична ефективність значною мірою визначається способом інтеграції семантичної інформації.

Порівнюючи отримані результати з класичними моделями сегментації, можна відзначити, що YOLOE-PF демонструє співставну або вищу якість відносно моделей сімейства YOLO-seg, де значення mAP50–95 для масок знаходяться на рівні близько 0.90–0.91.

Таблиця 2.3 – Метрики моделей YOLOE

| **Модель** | **mAP50_95_mask, %** | **mAP50_95_box, %** | **Inference, мс** | **Mipa F1** |
|---|---|---|---|---|
| YOLOE11m-seg-PF | 0,9179 | 0,9895 | 28,04 | 0,9975 |
| YOLOE11l-seg-PF | 0,9208 | 0,9876 | 34,17 | 0,9975 |
| YOLOE26m-seg-PF | 0,8643 | 0,9729 | 29,80 | 0,9843 |
| YOLOE26l-seg-PF | 0,8843 | 0,9722 | 32,69 | 0,9870 |
| YOLOE11m-seg | 0,7977 | 0,9189 | 25,51 | 0,7340 |
| YOLOE11l-seg | 0,7628 | 0,8752 | 32,11 | 0,7200 |
| YOLOE26m-seg | 0,8160 | 0,9420 | 30,72 | 0,6910 |
| YOLOE26l-seg | 0,7969 | 0,9166 | 40,84 | 0,6910 |

## 2.2 Моделі комп’ютерного зору для пошуку ключових точок

В даному розділі розглядаються виключно моделі сімейства YOLO-Pose, оскільки вони забезпечують оптимальне поєднання точності та швидкодії [18], а також добре інтегруються з іншими моделями, використаними у дослідженні. Це дозволяє побудувати єдиний уніфікований процес обробки.

На відміну від стандартних моделей YOLO, що виконують лише детекцію об’єктів, YOLO-Pose додатково передбачає визначення набору ключових точок, які описують скелет об’єкта. У контексті даної роботи такими точками є анатомічні орієнтири тіла корови: холка, плечовий суглоб, тазові кістки тощо, що використовуються для подальшого розрахунку ознак.

YOLO-Pose архітектурно базується на класичній YOLO концепції (рис 2.2), проте вихідний шар моделі модифіковано. Окрім координат обмежувального прямокутника та імовірності класу, модель повертає координати для кожної ключової точки та показник достовірності.

Отримані результати, що розташовані в таблиці 2.4 свідчать про те, що всі розглянуті моделі YOLO-Pose забезпечують дуже високі показники точності, а відмінності між ними за метриками mAP, Precision та Recall є незначними. Найкращий результат за метрикою mAP50-95 продемонструвала модель Yolo26n-pose (0.944), однак модель Yolo11n-pose показує співставну якість при меншому часі інференсу – лише 4.9 мс на GPU T4, що є найкращим показником швидкодії серед усіх протестованих моделей.

Таблиця 2.4 – Результати досліджень моделей Yolo-Pose

| **Модель** | **mAP50-95 (pose)** | **Precision** | **Recall** | **Inference GPU T4, мс** |
|---|---|---|---|---|
| Yolo11n-pose | 0,938 | 0,9998 | 1,0 | 4,9 |
| Yolo11s-pose | 0,935 | 0,9998 | 1,0 | 7,7 |
| Yolo11m-pose | 0,935 | 0,9998 | 1,0 | 20,4 |
| Yolo11l-pose | 0,928 | 0,9998 | 1,0 | 26,9 |
| Yolo26n-pose | 0,944 | 0,9995 | 1,0 | 5,7 |
| Yolo26s-pose | 0,940 | 0,9962 | 0,9992 | 8,8 |
| Yolo26m-pose | 0,937 | 0,9885 | 0,9956 | 21,7 |
| Yolo26l-pose | 0,938 | 0,9958 | 1,0 | 27,6 |

Із збільшенням розміру моделей суттєвого приросту точності не спостерігається, проте час обробки зображення значно зростає – до 20–27 мс для моделей середнього та великого розміру. Це свідчить про те, що для задач реального часу найбільш доцільним є використання компактних моделей, зокрема Yolo11n-pose або Yolo26n-pose, які забезпечують оптимальне співвідношення між точністю та швидкодією.

## 2.3 Моделі для вирішення задачі класифікації породи

Для дослідження та пошуку рішення даної задачі було використано інший набір даних на платформі Roboflow: cow Computer Vision Dataset [20, 21]. Він містить 38000 зображень і 66 класів різноманітних порід. Перелік моделей можна побачити на рисунку 2.4.

### 2.3.1 EffitientNet

Модель EfficientNet є однією з найбільш ефективних архітектур згорткових нейронних мереж, запропонованих для задач класифікації зображень [22]. Її ключова особливість полягає у використанні методу compound scaling, який дозволяє збалансовано масштабувати глибину, ширину та роздільну здатність вхідних даних. На відміну від традиційного підходу, де ці параметри змінюються незалежно, EfficientNet забезпечує оптимальне співвідношення між точністю та обчислювальними витратами.

Архітектура базується на блоках MBConv, що були запозичені з MobileNet, і використовує механізми squeeze-and-excitation для покращення якості ознак. Це дозволяє моделі ефективно виділяти важливі просторові та канальні характеристики зображення. Модель має декілька варіантів (від B0 до B7), які відрізняються масштабом і складністю.

### 2.3.2 YOLO-cls

Модель YOLO-cls є адаптацією архітектури YOLO для задач класифікації зображень [23]. У варіанті для класифікації фінальні шари моделі модифікуються таким чином, щоб виконувати прогноз класу для всього

зображення. Це дозволяє використовувати вже існуючі напрацювання та оптимізації, розроблені для YOLO, в даній області комп’ютерного зору.

### 2.3.3 DeiT

Модель DeiT (Data-efficient Image Transformer) є розвитком архітектури Vision Transformer (ViT), орієнтованим на ефективне навчання з обмеженою кількістю даних [24]. DeiT став однією з перших трансформерних моделей, яка змогла досягти високої точності класифікації без використання надзвичайно великих наборів даних. Основною особливістю DeiT є використання механізму knowledge distillation, при якому компактна модель навчається за допомогою більш складної моделі-вчителя.

На відміну від згорткових нейронних мереж, DeiT використовує механізм самоуваги (self-attention), який дозволяє враховувати взаємозв’язки між усіма частинами зображення. Вхідне зображення розбивається на фіксовані патчі, які перетворюються у послідовність векторів і подаються на вхід трансформера. Це дозволяє моделі формувати глобальні представлення ознак.

### 2.3.4 Swin Transformer

Swin Transformer є ієрархічною трансформерною архітектурою, яка була спеціально розроблена для задач комп’ютерного зору [25]. Її ключова інновація полягає у використанні механізму shifted windows, що дозволяє обмежити обчислення самоуваги локальними вікнами, одночасно забезпечуючи обмін інформацією між ними.

Завдяки ієрархічній структурі, Swin Transformer формує багаторівневі представлення ознак, подібно до згорткових мереж. Це дозволяє моделі

ефективно працювати як з дрібними деталями, так і з глобальним контекстом зображення.

### 2.3.5 ConvNeXt

Модель ConvNeXt є сучасною згортковою архітектурою, яка поєднує класичні підходи CNN з ідеями, запозиченими у трансформерів [26]. Вона була розроблена з метою переосмислення згорткових мереж з урахуванням сучасних практик навчання та архітектурних рішень.

ConvNeXt використовує великі ядра згортки, покращену нормалізацію та оптимізовані блоки обробки ознак, що дозволяє досягати продуктивності, співставної з трансформерними моделями. При цьому вона зберігає переваги CNN, такі як ефективність обчислень та стабільність навчання.

### 2.3.6 Порівняння

Результати показують, що саме сімейство YOLO11 демонструє найкращі результати за точністю (рис. 2.5, 2.6). Його значення top-1 знаходяться в межах 79–80%, а top-5 стабільно досягає 98.5%, що свідчить про високу впевненість моделі навіть у складних випадках класифікації. Особливо виділяються версії YOLO11m, YOLO11l та YOLO11x**,** які практично не відрізняються за якістю, але перевершують інші архітектури.

Другий рівень за точністю займають EfficientNet та частково YOLOv8. EfficientNet-B1, B3 та B5 показують помірно хороші результати на рівні 67–68% top-1, що робить їх стабільними, але вже помітно слабшими за YOLO11. YOLOv8-cls знаходиться ще нижче, приблизно 60–62%, що свідчить про його обмежену ефективність у даній задачі класифікації. Transformer-моделі,

зокрема DeiT та Swin, демонструють найнижчу якість (приблизно 29–55% залежно від варіанта), що вказує на їх слабку адаптацію до цього типу даних.


Рисунок 2.4 – Перелік всіх використаних моделей для порівняння в задачі класифікації

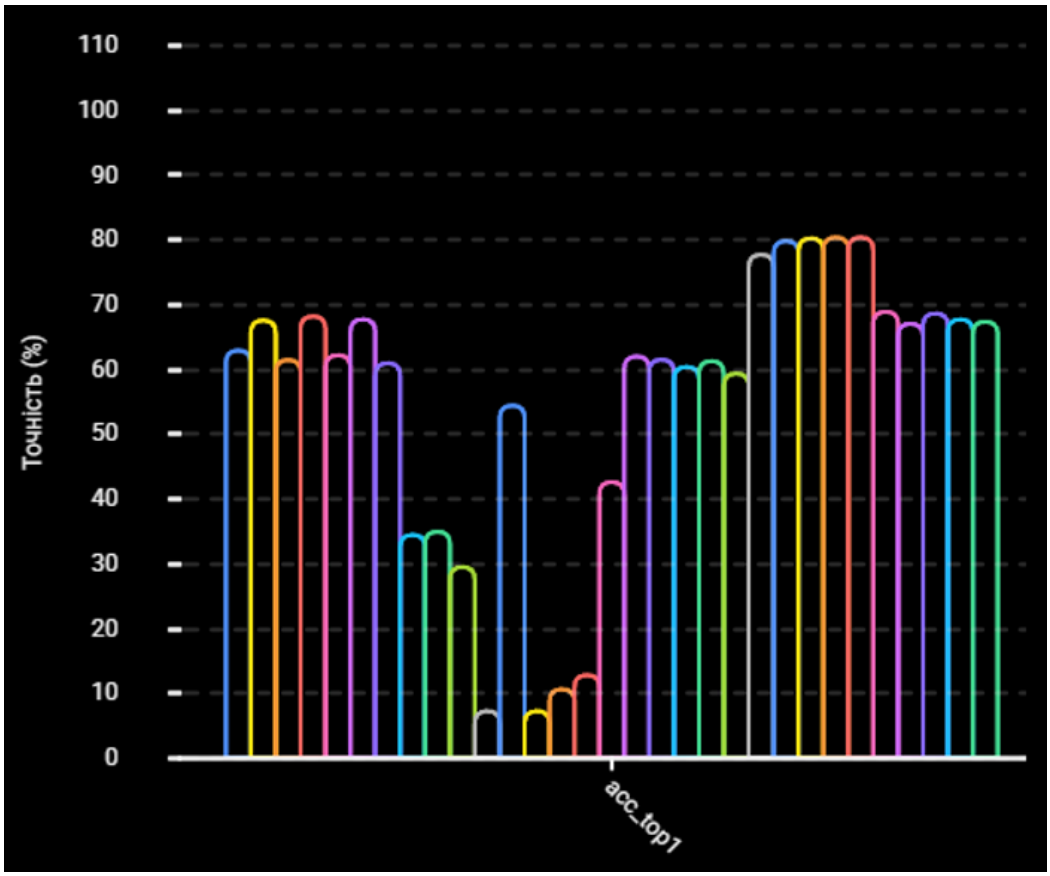



Рисунок 2.5 – Порівняння моделей у точності метрикою top1

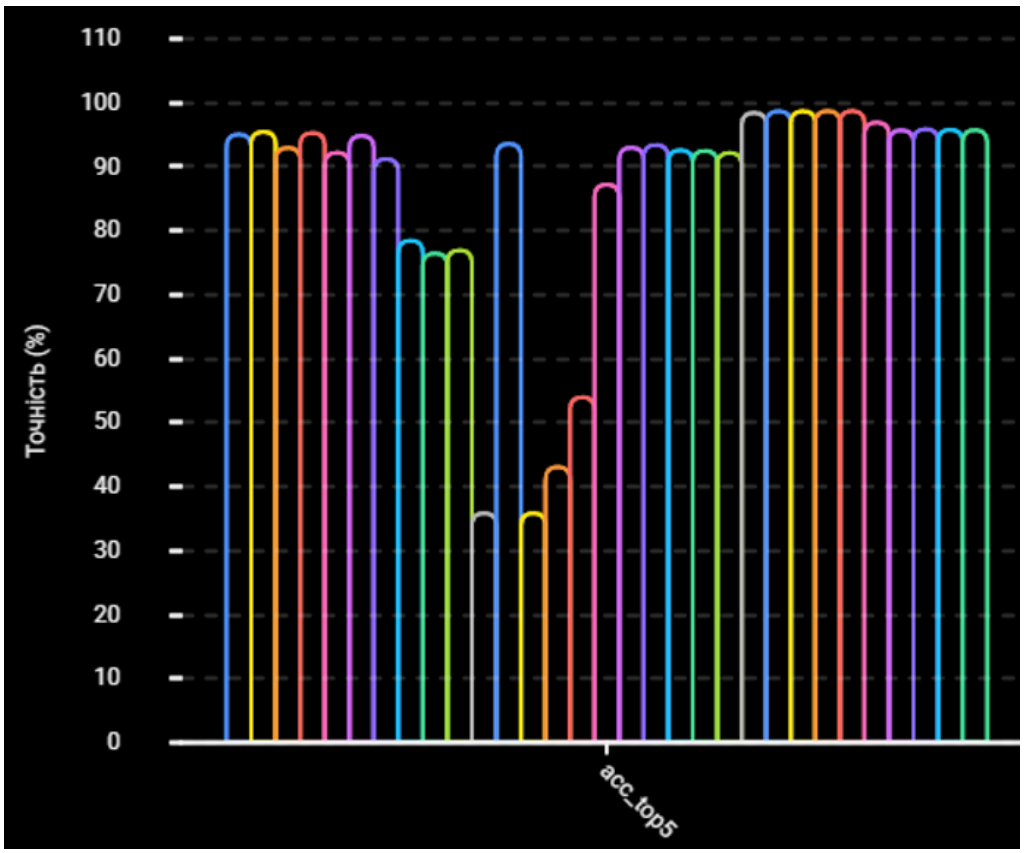



Рисунок 2.6 – Графік порівняння моделей у точності метрикою top5

З точки зору швидкості, на рисунку 2.7 можна побачити, що найкращі результати очікувано показують легкі моделі YOLOv8**.** YOLOv8n та YOLOv8s є найшвидшими – приблизно 2.5–3 мс на зображення, що робить їх оптимальними для реального часу. Сімейство YOLO11 також демонструє хорошу швидкість: 4–7 мс, з невеликим зростанням затримки при переході до більших варіантів, але без критичної втрати продуктивності.

EfficientNet та ConvNeXt займають середню позицію за швидкістю: приблизно 8–18 мс, при цьому більші версії (наприклад EfficientNet-B6 або ConvNeXt-Base) стають значно повільнішими.

Найповільнішими виявилися Swin-моделі та великі EfficientNet, де резутати досягають 20–30 мс і більше, що робить їх менш придатними для задач реального часу.

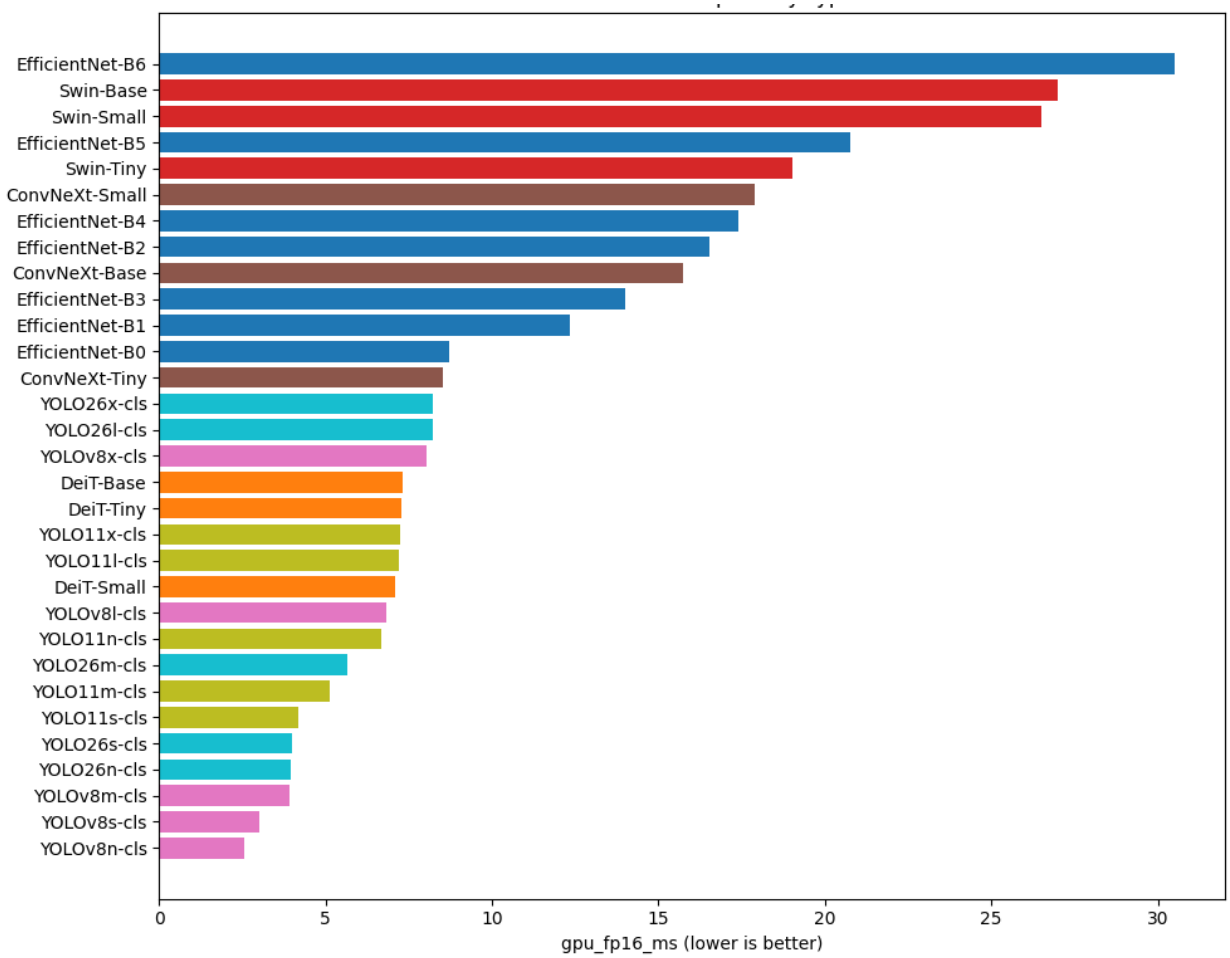



Рисунок 2.7 – Графік швидкості моделей класифікації

## 2.4 Методи визначення ваги великої рогатої худоби з використанням комп’ютерного зору

В роботі розглядаються та порівнюються три підходи до визначення маси великої рогатої худоби без використання вагового обладнання:

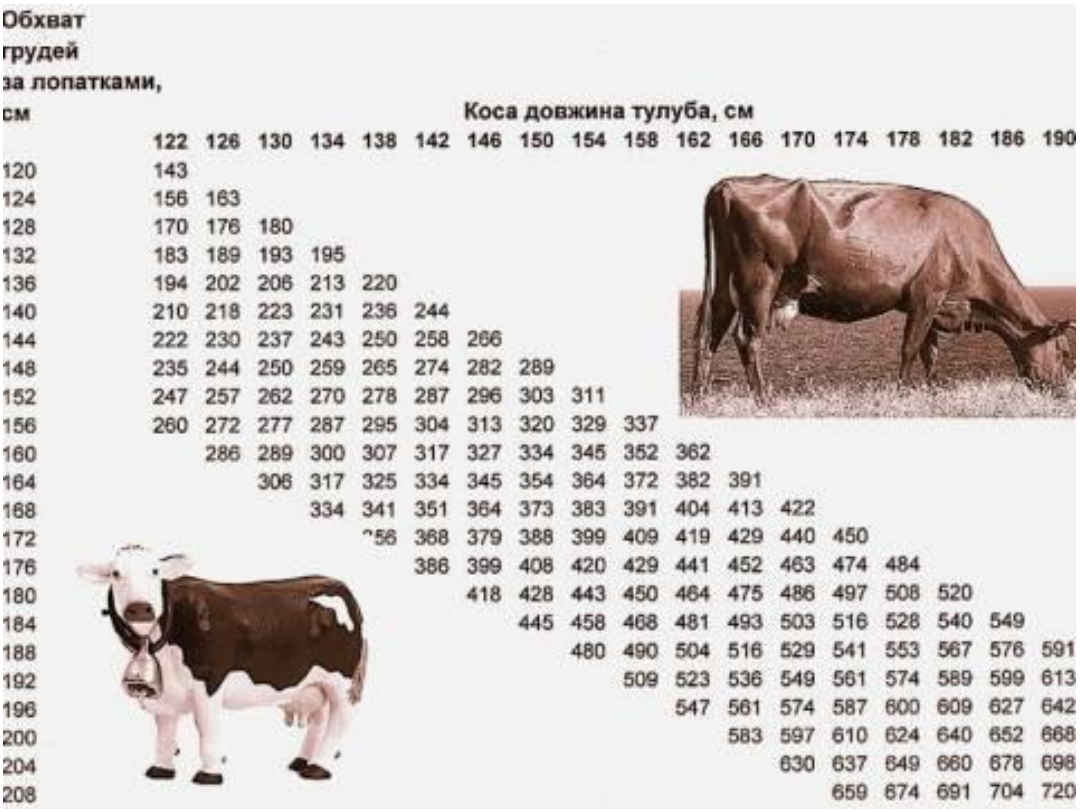
– класичний метод Клювера-Штрауха;

– метод на основі машинного навчання з використанням моделі XGBoost;

– метод на основі великих мовних моделей (LLM), що аналізують зображення та додаткові параметри тварини.

### 2.4.1 Класчний метод Клювер-Штрауха

Одним із найбільш відомих методів є метод Клювера-Штрауха, який базується на вимірюванні лінійних параметрів тіла тварини [27, 28]. У межах даного підходу маса обчислюється за таблицею (рис. 2.8), що враховує обхват грудей і косу довжину тулуба. Метод набув широкого поширення завдяки своїй простоті та достатньо високій точності при оцінюванні живої маси великої рогатої худоби.



| Обхват грудей за лопатками, см | Коса довжина тулуба, см | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 122 | 126 | 130 | 134 | 138 | 142 | 146 | 150 | 154 | 158 | 162 | 166 | 170 | 174 | 178 | 182 | 186 | 190 |
| 120 | 143 | | | | | | | | | | | | | | | | | |
| 124 | 156 | 163 | | | | | | | | | | | | | | | | |
| 128 | 170 | 176 | 180 | | | | | | | | | | | | | | | |
| 132 | 183 | 189 | 193 | 195 | | | | | | | | | | | | | | |
| 136 | 194 | 202 | 206 | 213 | 220 | | | | | | | | | | | | | |
| 140 | 210 | 218 | 223 | 231 | 236 | 244 | | | | | | | | | | | | |
| 144 | 222 | 230 | 237 | 243 | 250 | 258 | 266 | | | | | | | | | | | |
| 148 | 235 | 244 | 250 | 259 | 265 | 274 | 282 | 289 | | | | | | | | | | |
| 152 | 247 | 257 | 262 | 270 | 278 | 287 | 296 | 303 | 311 | | | | | | | | | |
| 156 | 260 | 272 | 277 | 287 | 295 | 304 | 313 | 320 | 329 | 337 | | | | | | | | |
| 160 | | 286 | 289 | 300 | 307 | 317 | 327 | 334 | 345 | 352 | 362 | | | | | | | |
| 164 | | | 306 | 317 | 325 | 334 | 345 | 354 | 364 | 372 | 382 | 391 | | | | | | |
| 168 | | | | 334 | 341 | 351 | 364 | 373 | 383 | 391 | 404 | 413 | 422 | | | | | |
| 172 | | | | | 356 | 368 | 379 | 388 | 399 | 409 | 419 | 429 | 440 | 450 | | | | |
| 176 | | | | | | 386 | 399 | 408 | 420 | 429 | 441 | 452 | 463 | 474 | 484 | | | |
| 180 | | | | | | | 418 | 428 | 443 | 450 | 464 | 475 | 486 | 497 | 508 | 520 | | |
| 184 | | | | | | | | 445 | 458 | 468 | 481 | 493 | 503 | 516 | 528 | 540 | 549 | |
| 188 | | | | | | | | | 480 | 490 | 504 | 516 | 529 | 541 | 553 | 567 | 576 | 591 |
| 192 | | | | | | | | | | 509 | 523 | 536 | 549 | 561 | 574 | 589 | 599 | 613 |
| 196 | | | | | | | | | | | 547 | 561 | 574 | 587 | 600 | 609 | 627 | 642 |
| 200 | | | | | | | | | | | | 583 | 597 | 610 | 624 | 640 | 652 | 668 |
| 204 | | | | | | | | | | | | | 630 | 637 | 649 | 660 | 678 | 698 |
| 208 | | | | | | | | | | | | | | 659 | 674 | 691 | 704 | 720 |

Рисунок 2.8 – Приклад таблиці для методу Клювера-Штрауха

У класичному варіанті цей метод потребує ручного вимірювання параметрів тварини, однак в межах даної роботи ці вимірювання автоматично отримуються із зображень за допомогою методів комп’ютерного зору. Це дозволяє зменшити вплив людського фактора, прискорити процес оцінювання та забезпечити можливість безконтактного визначення маси тварини.

Результати в таблиці 2.5 показують, що метод може забезпечувати прийнятну точність для окремих тварин, однак стабільність оцінювання залишається недостатньою. Для різних прикладів похибка суттєво відрізняється та в окремих випадках перевищує 60 %, що свідчить про високу чутливість методу до точності визначення лінійних параметрів тіла тварини. Крім того, використання фіксованих табличних залежностей не дозволяє враховувати індивідуальні особливості тварин.

Таблиця 2.5 – Результати досліджень методу Клювера-Штрауха

| Справжня вага, кг | Результат, кг | Похибка, % |
|---|---|---|
| 229 | 196 | 14,41 |
| 114 | 189 | 65,79 |
| 137 | 189 | 37,96 |
| 465 | 356 | 23,44 |

### 2.4.2 Метод з використанням машинного навчання

Другий підхід базується на використанні алгоритмів машинного навчання, зокрема моделі XGBoost [29]. На відміну від аналітичного методу, XGBoost дозволяє враховувати складні нелінійні залежності між вхідними ознаками та масою тварини.

Як вхідні дані використовуються ознаки, отримані із зображень:

− відстані між ключовими точками тіла;

− лінійні розміри (довжина, висота);

− площа та форма силуету тварини;

− різноманітні співвідношення та похідні характеристики.

Особливістю цього підходу є те, що модель навчається на реальних даних і здатна адаптуватися до різних порід і умов зйомки.

Результати (табл. 2.6) демонструють, що модель XGBoost здатна забезпечувати високу точність для окремих прикладів. Але водночас спостерігається значне відхилення, зокрема у випадку з твариною масою 465 кг, де похибка перевищила 65%. Однією з причин цього є те, що модель працює виключно з числовими ознаками, отриманими із зображення, і не враховує візуальні особливості тварини, зокрема рівень вгодованості. У реальних умовах тварини можуть мати подібні лінійні розміри та пропорції тіла, але суттєво відрізнятися за масою через різний об’єм м’язової та жирової тканини, що ускладнює точне прогнозування ваги лише на основі числових параметрів.

Таблиця 2.6 – Результати досліджень використання XGBoost

| Справжня вага, кг | Результат, кг | Похибка, % |
|---|---|---|
| 229 | 136 | 40,61 |
| 114 | 113 | 0,88 |
| 137 | 139 | 1,46 |
| 465 | 162 | 65,16 |

### 2.4.3 LLM

Третій підхід базується на використанні великих мовних моделей (LLM), які мають мультимодальні можливості обробки зображень і текстової інформації. Завдяки цьому такі моделі можуть застосовуватися у широкому спектрі задач та предметних областей [30, 31]. Використання таких моделей дозволяє спробувати підвищити точність оцінювання маси тварини завдяки аналізу не лише числових параметрів, а й безпосередньо самого зображення. На відміну від моделей машинного навчання, що працюють переважно зі структурованими числовими ознаками, мультимодальні LLM здатні

враховувати візуальні характеристики тварини, зокрема рівень вгодованості, особливості тілобудови та загальну кондицію тіла.

У межах цього методу на вхід моделі подаються:

− зображення тварини;

− додаткова інформація у вигляді лінійних розмірів.

Результати, наведені в таблиці 2.7, демонструють, що мультимодальні LLM здатні забезпечувати достатньо високу точність оцінювання маси ВРХ навіть без використання спеціалізованих моделей регресії. Найкращий результат серед протестованих моделей показала Claude Sonnet 4.6, середня похибка якої склала 12.32 %. Також відносно стабільні результати продемонстрували Gemini 3.1 Pro та GPT-5.5 Instant.

Таблиця 2.7 – Результати досліджень з LLM

| **Справжня вага** | **Gemini 3 Flash** | **Gemini 3.1 Pro** | **Claude Opus 4.7** | **Claude Sonnet 4.6** | **GPT-5.5 Instant** |
|---|---|---|---|---|---|
| 137 кг | 145 кг | 175 кг | 110 кг | 148 кг | 155 кг |
| 114 кг | 138 кг | 140 кг | 75 кг | 112 кг | 130 кг |
| 229 кг | 158 кг | 245 кг | 140 кг | 210 кг | 185 кг |
| 465 кг | 265 кг | 395 кг | 280 кг | 320 кг | 340 кг |
| **Середня похибка** | 25,23% | 18,15% | 33,14% | 12,32% | 18,32% |

Отримані результати підтверджують, що використання мультимодальних моделей дозволяє враховувати не лише числові параметри, а й візуальні особливості тварини, зокрема рівень вгодованості, пропорції тіла та загальну кондицію. Це частково компенсує обмеження підходів, що базуються виключно на числових ознаках. У результаті найкращу точність серед протестованих підходів продемонструвала модель Claude Sonnet 4.6 із середньою похибкою 12.32 %.

### 2.4.4 Використання комп’ютерного зору та виділення ознак

У всіх розглянутих підходах використовується єдиний етап виділення ознак на основі комп’ютерного зору.

Сегментація зображення дозволяє виділити силует тварини, а визначення ключових точок – локалізувати анатомічно значущі орієнтири. На основі цих даних обчислюються відстані між точками у піксельному просторі.

Однак отримані значення відстаней у пікселях не дозволяють безпосередньо використовувати їх у класичних зоотехнічних формулах. У зв’язку з цим вводиться етап калібрування, що дозволяє перевести піксельні відстані в реальні одиниці вимірювання (наприклад, сантиметри) [32].

Калібрування виконується з використанням еталонного об’єкта відомого розміру (наприклад, маркера або стікера, розміщеного на тілі тварини). Знаючи реальний розмір об’єкта та його розмір у пікселях на зображенні, визначається коефіцієнт масштабування:

$$k = \frac{L_{real}}{L_{pixel}}, \tag{2.1}$$

де $L_{real}$ – реальний розмір еталонного об’єкта;

$L_{pixel}$ – його розмір у пікселях на зображенні.

Після визначення коефіцієнта масштабування всі відстані між ключовими точками переводяться у реальні одиниці:

$$D_{real} = k \cdot D_{pixel}. \tag{2.2}$$

Таким чином стає можливим відновлення наближених фізичних параметрів тварини, таких як:

− обхват грудей;

− довжина тулуба;

− висота в холці.

Отримані значення використовуються всіма способами.

Для застосування методу Клювера-Штрауха. Розраховані параметри безпосередньо підставляються в таблицю, що дозволяє отримати оцінку маси без ручних вимірювань.

Для формування ознак моделі XGBoost. Ті ж самі параметри, а також додаткові похідні характеристики (співвідношення довжин, площа маски, пропорції тіла) використовуються як вхідні дані для моделі машинного навчання.

Для застосування методу на основі LLM. Виділені ознаки (лінійні розміри, пропорції, площа силуету), разом із зображенням, подаються на вхід мультимодальної моделі, яка використовує їх як структуровану та контекстну інформацію для формування оцінки маси.

Таким чином, усі розглянуті методи використовують єдиний набір базових ознак, отриманих за допомогою комп’ютерного зору, хоча й по-різному інтерпретують їх: від прямого підстановлення у формули до використання в якості вхідних даних для моделей машинного навчання та LLM. Це забезпечує коректність їх порівняльного аналізу в межах роботи та дозволяє об’єктивно оцінити точність і доцільність застосування кожного підходу в системах автоматизованого моніторингу великої рогатої худоби.

## 2.5 Висновки на основі результатів

Проведені тести дозволили визначити найбільш доцільні архітектури для кожної із поставлених задач.

Для задачі сегментації зображень найбільш придатними для практичного використання виявилися моделі сімейства YOLO-seg, зокрема YOLOv8m-seg та YOLO26m-seg, які забезпечують оптимальну точність сегментації при гарній швидкості.

Для задачі визначення ключових точок найбільш ефективними виявилися компактні моделі Yolo11n-pose та Yolo26n-pose. При цьому збільшення розміру моделей не дало суттєвого приросту якості, але значно збільшило обчислювальні витрати.

У задачі класифікації породи найкращі результати продемонструвало сімейство YOLO11, яке забезпечило найвищі показники точності серед усіх протестованих архітектур. Моделі EfficientNet показали стабільні, але нижчі результати, тоді як трансформерні архітектури DeiT та Swin Transformer виявилися менш ефективними для даного типу даних.

Таким чином, результати дослідження свідчать про те, що саме моделі сімейства YOLO є найбільш універсальним та ефективним рішенням для побудови комплексної системи моніторингу ВРХ, оскільки вони забезпечують високу швидкість роботи, достатню точність та можливість інтеграції різних задач комп'ютерного зору в єдиному програмному середовищі.

Що до методів оцінки ваги, то результати показали, що при використанні лише одного бокового ракурсу тварини найбільш ефективним підходом для визначення маси ВРХ є застосування мультимодальних великих мовних моделей (LLM). На відміну від класичного методу Клювера-Штрауха та моделі XGBoost, які переважно базуються на аналізі числових параметрів і лінійних розмірів, LLM здатні додатково враховувати безпосередньо візуальні особливості тварини, зокрема рівень вгодованості, пропорції тіла та загальну кондицію. Це дозволяє частково компенсувати обмеження, пов'язані з використанням лише бокового зображення, де неможливо повноцінно оцінити об'єм тіла та просторові характеристики тварини.

В результаті запропоновано наступний алгоритм обробки відео (рис. 2.9, 2.10).

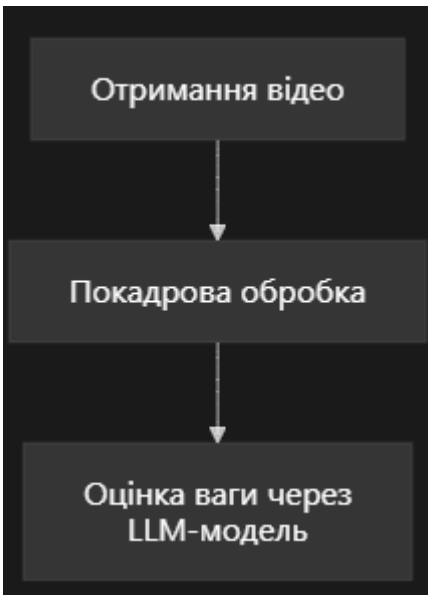



Рисунок 2.9 – Основні етапи роботи алгоритму

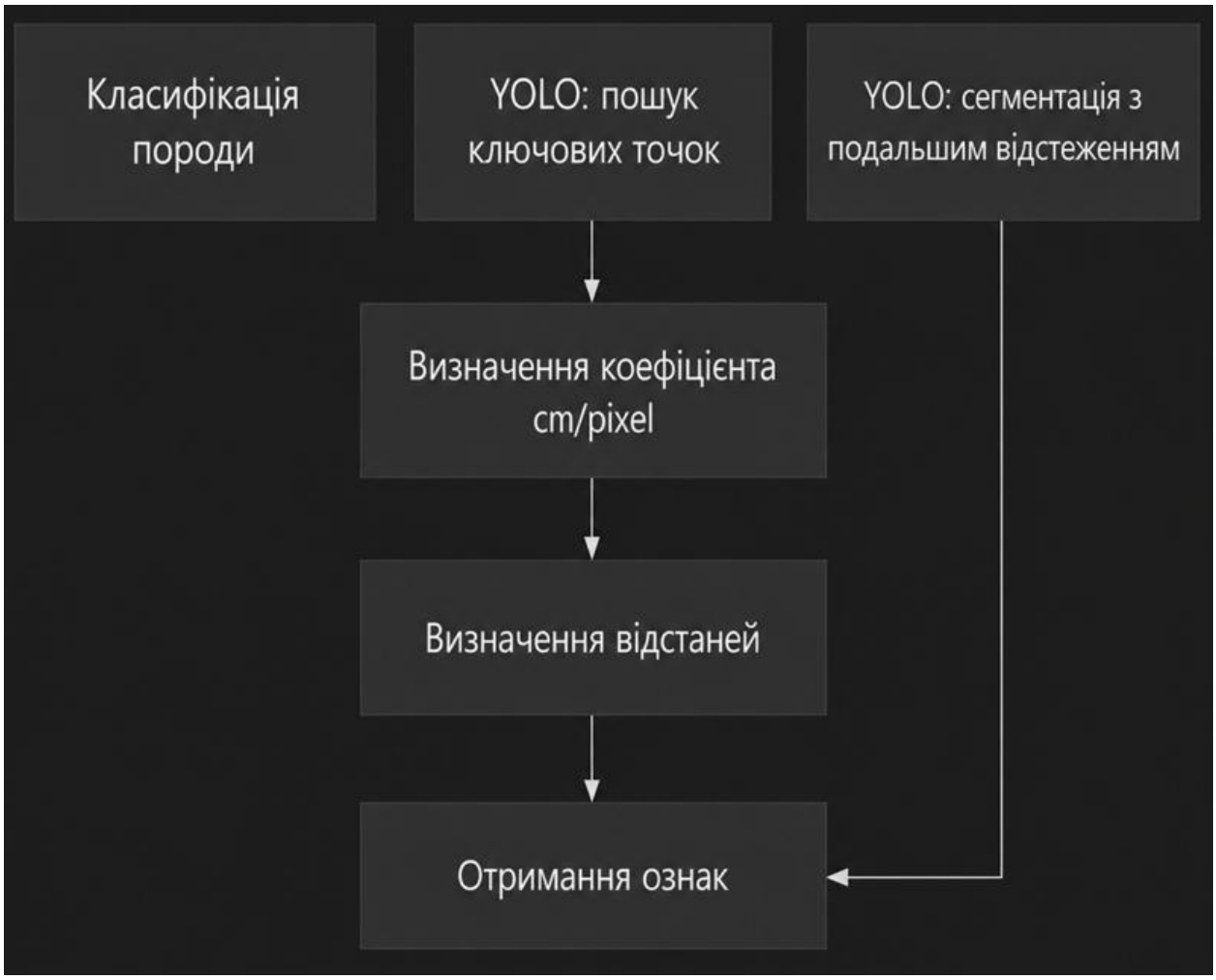



Рисунок 2.10 – Підпункти етапу «Покадрова обробка»

# 3 РОЗРОБКА СИСТЕМИ АНАЛІЗУ ВІДЕО ДЛЯ ОЦІНКИ ВАГИ КОРІВ

## 3.1 Розробка серверної частини

У даному розділі описується реалізація серверної частини програмного комплексу для автоматизованого аналізу відео корів та оцінки їх ваги. Серверна частина є основним обчислювальним модулем системи, оскільки саме на сервері виконуються всі операції, пов’язані з комп’ютерним зором, обробкою відеопотоку, запуском моделей штучного інтелекту, формуванням результатів та взаємодією з базою даних.

Основною задачею серверу є:

− прийом відеофайлів від клієнтського застосунку;

− збереження файлів та створення задач аналізу;

− організація асинхронної обробки відео;

− запуск моделей комп’ютерного зору;

− виділення характеристик корови;

− оцінка ваги за допомогою LLM;

− формування анотованого відео;

− збереження результатів;

− передача результатів клієнту.

Для реалізації серверної частини було обрано мову Python. Даний вибір обумовлений широкою підтримкою сучасних технологій комп’ютерного зору, машинного навчання та штучного інтелекту. Незважаючи на те, що Python належить до інтерпретованих мов програмування, основні обчислення у системі виконуються оптимізованими бібліотеками, реалізованими мовами C, C++. Завдяки цьому ресурсоємні операції комп’ютерного зору та обробки відео виконуються з високою продуктивністю, а Python використовується переважно як засіб інтеграції окремих компонентів системи та організації логіки роботи серверного застосунку.

Для серверної частини було використано мікросервісний підхід із логічним розділенням відповідальності між окремими сервісами. HTTP-сервер відповідає лише за прийом запитів та повернення результатів, тоді як ресурсоємний аналіз виконується б окремими процесами.

Архітектура серверної частини була обрана таким чином, щоб забезпечити:

− масштабованість;

− асинхронну обробку;

− незалежність компонентів;

− підтримку GPU та CPU режимів.

Таким чином серверна частина дозволяє ефективно виконувати тривалий аналіз відео без блокування користувацьких запитів. Структура проекту зображена на рисунку 3.1.


Рисунок 3.1 – Структура вебсерверу

### 3.1.1 Серверні технології та backend-архітектура

Для реалізації серверної частини системи було обрано фреймворк Django. Вибір даного фреймворку обумовлений наявністю великої кількості готових компонентів, необхідних для побудови повноцінної серверної системи. Додатковою перевагою є вбудована ORM-система, яка дозволяє працювати з базою даних через об’єктну модель без необхідності написання SQL-запитів вручну. Це спрощує розробку, підвищує підтримуваність програмного коду та забезпечує більш структуровану організацію даних у системі.

Важливою перевагою Django для даного проєкту стала проста інтеграція із системою фонової обробки та базою даних. Це дозволило реалізувати серверну архітектуру із чітким розподілом відповідальності між компонентами та забезпечити надійне виконання довготривалих задач аналізу відео.

Для фонової обробки було задіяно популярну розподільну систему черг завдань Celery. Django після отримання відео не виконує аналіз самостійно, а лише створює задачу та передає її до Celery. Після цього окремий Celery worker отримує задачу та виконує весь процес обробки у фоновому режимі.

Такий підхід має декілька важливих переваг. По-перше, HTTP-сервер залишається вільним та може продовжувати обробляти нові запити. По-друге, система отримує можливість масштабування, оскільки можна запускати декілька worker-процесів паралельно. По-третє, помилки обробки ізольовані від основного web-сервера.

У межах системи Celery відповідає за:

− запуск фонових задач;

− виконання аналізу відео;

− оновлення статусу задач;

− ізоляцію довготривалих обчислень;

− підтримку асинхронної архітектури.

Також у роботі задіяний брокер повідомлень Redis, який буде використовуватися між Django та Celery. Основною його задачею є організація передачі задач від HTTP-сервера до Celery, які виконують фонову обробку відео.

Після отримання відеофайлу Django створює задачу аналізу та поміщає її до черги Redis. Далі Celery постійно відстежує появу нових задач у черзі та, отримавши одну з них, розпочинає виконання обробки відео. Таким чином Redis виступає проміжним буфером між web-сервером та системою асинхронної обробки.

Базою даних для збереження інформації про задачі обробки відео виступає PostgreSQL. Результати аналізу та службові дані серверної частини. Використання окремої бази даних є необхідним для централізованого збереження результатів роботи системи, контролю стану задач та організації взаємодії між компонентами серверної архітектури.

Після отримання відеофайлу сервер створює новий запис у базі даних, який містить унікальний ідентифікатор задачі, поточний статус обробки, назву відеофайлу, етап виконання та службову інформацію. У процесі роботи Celery worker постійно оновлює дані у PostgreSQL, зокрема прогрес аналізу, поточний стан та результати роботи моделей комп’ютерного зору.

Після завершення обробки у базі даних також зберігаються результати аналізу кожної корови. До них входять визначена порода, геометричні характеристики, оцінка ваги та рівень впевненості моделі. Завдяки цьому система забезпечує централізоване та структуроване збереження всіх результатів обробки. Збереження відбувається у дві таблиці.

Модель VideoTask використовується для збереження службової інформації про кожну задачу обробки відео (табл. 3.1). Вона містить ідентифікатор задачі, поточний статус виконання, значення прогресу, етап обробки та інформацію про можливі помилки. Саме ця таблиця використовується клієнтським застосунком для відстеження стану виконання аналізу відео.

Таблиця 3.1 – Поля таблиці VideoTask

| **Поле** | **Тип даних** | **Опис** |
|---|---|---|
| task_id | UUID / String | Унікальний ідентифікатор задачі |
| videofile_name | String | Назва завантаженого відеофайлу |
| received_at | DateTime | Час отримання відео сервером |
| status | String | Поточний стан задачі (processing, done, error) |
| progress | Integer | Відсоток виконання обробки (0–100 %) |
| stage | String | Поточний етап виконання |
| error | Text | Опис помилки при виникненні збою |

Таблиця CowResult містить результати аналізу окремих корів, виявлених у відео. Первинним ключем використовується окреме поле id замість складеного ключа з полів task та cow_id. Таке рішення обумовлене особливостями ORM Django, яка більш ефективно працює з єдиним первинним ключем. Для запобігання дублюванню результатів додатково використовується обмеження унікальності для комбінації полів task і cow_id.

Для кожної тварини зберігаються розраховані біометричні характеристики, визначена порода, оцінка ваги та рівень впевненості прогнозу (табл. 3.2). Зв’язок із таблицею VideoTask дозволяє отримати всі результати, що належать до конкретної задачі обробки відео (рис. 3.2).

Таблиця 3.2 – Поля таблиці CowResult

| **Поле** | **Тип даних** | **Опис** |
|---|---|---|
| id | Integer | Первинний ключ запису |
| task | UUID / String | Посилання на задачу обробки |
| cow_id | Integer | Ідентифікатор корови у відео |
| measurements | JSON | Набір розрахованих біометричних характеристик |
| breed | String | Визначена порода корови |
| weight | Float | Прогнозована вага |
| weight_confidence | Float | Рівень впевненості оцінки ваги |

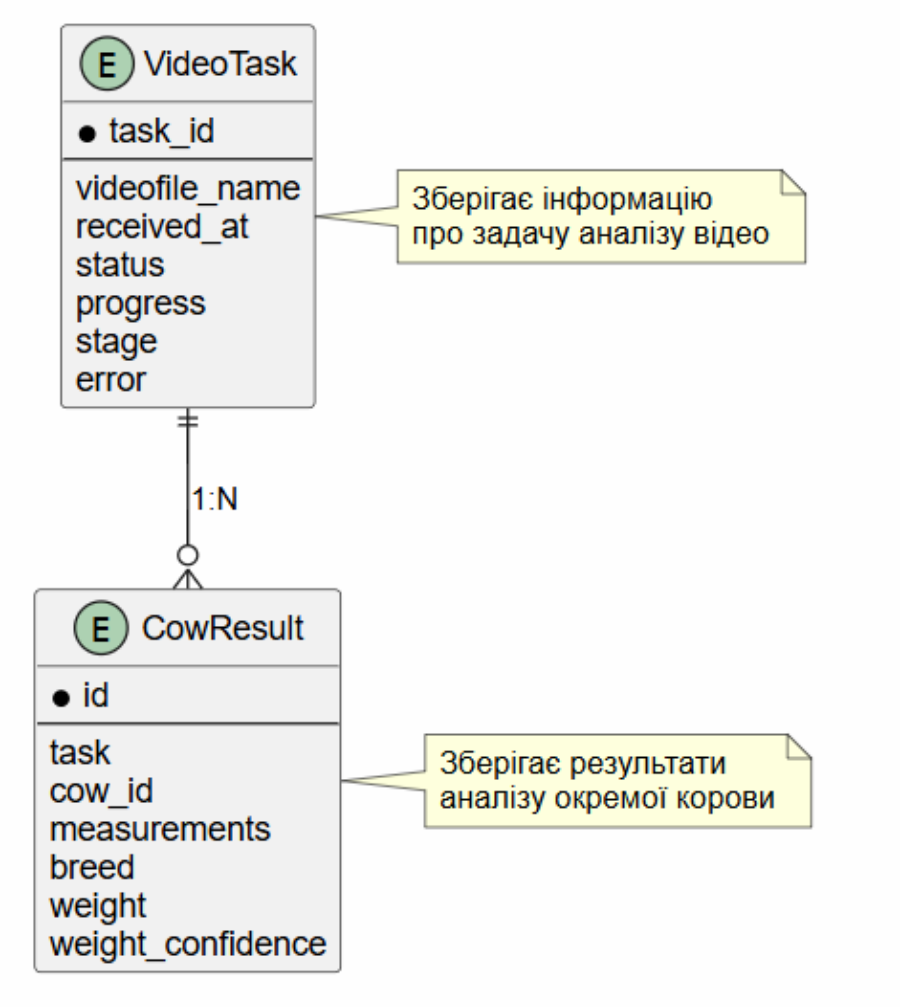



Рисунок 3.2 – ER-діаграма бази даних

### 3.1.2 Технології комп’ютерного зору та штучного інтелекту

Для реалізації системи аналізу відео та оцінки ваги корів у роботі використовується комплекс бібліотек і сервісів, що забезпечують повний цикл комп’ютерного зору та подальшого інтелектуального аналізу даних. Основними компонентами цього процесу є OpenCV, Ultralytics та Anthropic API.

OpenCV є базовою бібліотекою для роботи з відео та графічною обробкою кадрів. Після отримання відеофайлу Celery worker відкриває його за допомогою OpenCV та виконує покадрове зчитування. Кожен кадр

передається до моделей комп'ютерного зору для подальшого аналізу. Бібліотека використовується для:

– відкриття відеофайлів;

– покадрового зчитування;

– зміни розміру кадрів;

– роботи з координатами;

– створення графічних анотацій;

– запису результатів у фінальний відеофайл.

Після отримання результатів моделей OpenCV використовується для побудови анотацій на кадрах. На зображення наносяться bounding box, ключові точки, ID корови та порода. Також бібліотека застосовується для генерації фінального анотованого відео annotated.mp4, яке потім передається клієнтському застосунку.

Для реалізації задач комп'ютерного зору використовується фреймворк Ultralytics. Його перевагою є можливість працювати з усіма моделями лінійки YOLO у різних форматах. Для забезпечення роботи системи використовуються три моделі сімейства YOLO та алгоритм відстеження BoT-SORT (табл. 3.3). Для прискорення обробки застосовуються моделі у форматі .pt при наявності GPU, а при роботі на CPU автоматично використовуються оптимізовані моделі у форматі .onnx.

Таблиця 3.3 – Перелік використовуваних YOLO моделей

| **Модель** | **Тип задачі** | **GPU-версія** | **CPU-версія** |
|---|---|---|---|
| YOLO26l-seg | Виділення контуру корови на кадрі | Yolo26l-seg.pt | Yolo26l-seg.onnx |
| YOLO26n-poseB | Визначення анатомічних ключових точок тіла корови | Yolo26n-poseB.pt | Yolo26n-poseB.onnx |
| YOLO11m-cls | Визначення породи тварини | Yolo11m-cls.pt | Yolo11m-cls.onnx |
| BoT-SORT | Відстеження корів між кадрами відео | botsort_reid20.yaml | botsort_reid20.yaml |

Після завершення етапу комп’ютерного зору система переходить до оцінки ваги корови за допомогою моделі Claude Sonnet 4.6. Для взаємодії з моделлю використовується Anthropic API – програмний інтерфейс, розроблений компанією Anthropic для доступу до власних LLM-моделей без використання графічного інтерфейсу. Використання API дозволяє автоматизувати процес надсилання запитів та отримання відповідей без застосування GUI-сервісів, які орієнтовані переважно на ручну взаємодію користувача.

Додатковою перевагою використання API є можливість більш гнучкого налаштування параметрів взаємодії з моделлю. Зокрема, система дозволяє:

– обирати конкретну модель;

– задавати максимальну кількість токенів у відповіді;

– налаштовувати temperature для контролю випадковості відповідей;

– задавати system prompt;

– обмежувати витрати шляхом контролю розміру запитів та відповідей;

– автоматизувати серійне опрацювання великої кількості даних.

Під час роботи системи формуються числові характеристики корови, а також зберігається оптимальне зображення тварини. Після цього система:

– завантажує розраховані характеристики;

– формує prompt;

– передає зображення та prompt до Anthropic API;

– отримує структурований JSON-результат;

– витягує значення ваги та confidence prediction.

Для числових обчислень та роботи з багатовимірними масивами даних у системі використовується NumPy.

Бібліотека активно застосовується під час математичних операцій, обробки координат ключових точок, обчислення відстаней між ними та формування набору біометричних характеристик корови.

### 3.1.3 Технології контейнеризації та розгортання

Під час розробки серверних систем часто виникають проблеми, пов’язані з відмінностями середовищ виконання. Різні версії Python, бібліотек, драйверів або системних залежностей можуть призводити до нестабільної роботи застосунку на різних комп’ютерах чи серверах. Для вирішення цієї проблеми у системі використовується контейнеризація за допомогою Docker.

Контейнеризація дозволяє запускати застосунок у ізольованому середовищі разом з усіма необхідними бібліотеками, залежностями та конфігурацією. Це забезпечує однакову поведінку системи незалежно від операційної системи або сервера, на якому виконується запуск.

Docker забезпечує:

– ізоляцію компонентів системи;

– стандартизацію середовища виконання;

– спрощення розгортання;

– переносимість між різними операційними системами;

– стабільність залежностей та конфігурації.

У Docker-середовищі кожен основний компонент системи запускається в окремому контейнері (рис. 3.3). У системі використовуються такі контейнери:

– web – Django backend;

– worker – Celery worker для фонової обробки відео;

– postgres – сервер PostgreSQL;

– redis – брокер повідомлень Redis.

Такий підхід дозволяє ізолювати сервіси один від одного та забезпечує більш стабільну роботу всієї системи. Крім того, контейнеризація значно спрощує перенесення проєкту між різними середовищами розробки та серверами.

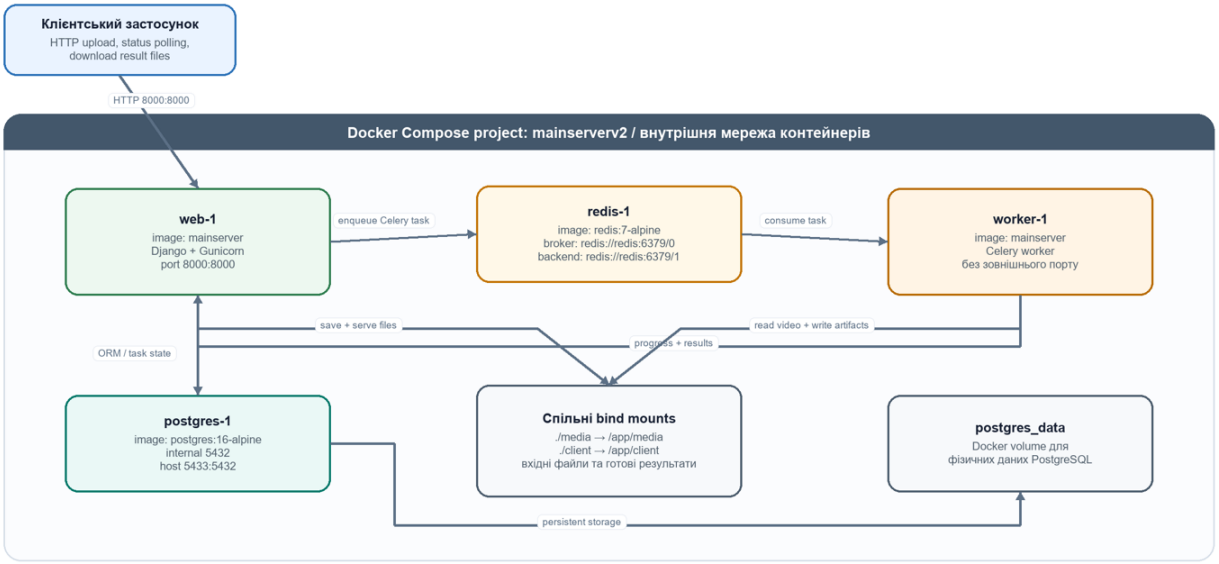



Рисунок 3.3 – Схема взаємодії Docker-контейненрів

Контейнери web і worker збираються з одного образу mainserver, але виконують різні entrypoint-скрипти. web приймає HTTP-запити клієнта та створює задачі, Redis передає ці задачі worker, worker виконує обробку відео. Обидва контейнери звертаються до PostgreSQL і використовують спільні bind mounts /media та ./client для вхідних файлів і готових результатів.

Для керування контейнерами використовується Docker Compose. Він автоматично створює мережу між контейнерами, забезпечує передачу змінних середовища та спрощує взаємодію між окремими сервісами (рис. 3.4).

Використання Docker також дозволяє уникнути проблем із несумісністю бібліотек, версій Python та системних залежностей. Усі необхідні компоненти встановлюються безпосередньо всередині контейнерів, що забезпечує однакову поведінку системи незалежно від середовища запуску Конфігурація Docker-контейнера серверної частини наведено у додатку А.

## 3.2 Клієнтський застосунок

Для взаємодії користувача із серверною частиною системи було розроблено окремий десктопний клієнтський застосунок. Його основним

призначенням є завантаження відеофайлів на сервер, відстеження процесу обробки та завантаження отриманих результатів. На відміну від серверної частини, клієнт не виконує обчислювально складних операцій комп'ютерного зору та не використовує моделі штучного інтелекту. Усі ресурсоємні задачі виконуються на сервері, тоді як клієнт виступає інтерфейсом взаємодії між користувачем та backend-сервісом.

Архітектура клієнтського застосунку побудована за модульним принципом та складається з декількох основних компонентів:

– модуль графічного інтерфейсу користувача;

– модуль мережевої взаємодії із сервером;

– модуль фонової обробки запитів;

– модуль локального збереження результатів.

Для реалізації графічного інтерфейсу розглядалися декілька популярних бібліотек Python, зокрема Tkinter та PyQt5. Бібліотека Tkinter входить до стандартного набору Python та добре підходить для створення простих настільних застосунків. Проте функціональні вимоги розробленої системи передбачають виконання тривалих мережевих операцій, завантаження великих відеофайлів, відображення прогресу обробки та використання багатопотоковості для забезпечення безперервної роботи інтерфейсу.

PyQt5 надає більш розвинуті засоби для створення сучасних графічних інтерфейсів, а також містить вбудовану підтримку багатопотоковості через механізм QThread. Це дозволяє виконувати тривалі операції у фоновому режимі без блокування головного вікна програми. Крім того, бібліотека містить широкий набір готових компонентів інтерфейсу, зокрема прогрес-бари, діалогові вікна, елементи введення даних та механізми обміну повідомленнями між потоками. З огляду на необхідність забезпечення стабільної роботи застосунку під час передачі та обробки відеофайлів, для реалізації клієнтської частини було обрано саме PyQt5.

Точкою входу до застосунку є файл main.py, який створює екземпляр QApplication, ініціалізує головне вікно програми та запускає цикл обробки

подій. Головне вікно реалізоване за допомогою бібліотеки PyQt5 та містить усі елементи керування, необхідні для роботи користувача із системою (рис. 3.4).

Інтерфейс клієнтського застосунку містить:

– поле введення IP-адреси сервера;

– кнопку вибору відеофайлу;

– індикатор прогресу обробки;

– інформаційне поле поточного статусу;

– блок відображення результатів обробки.

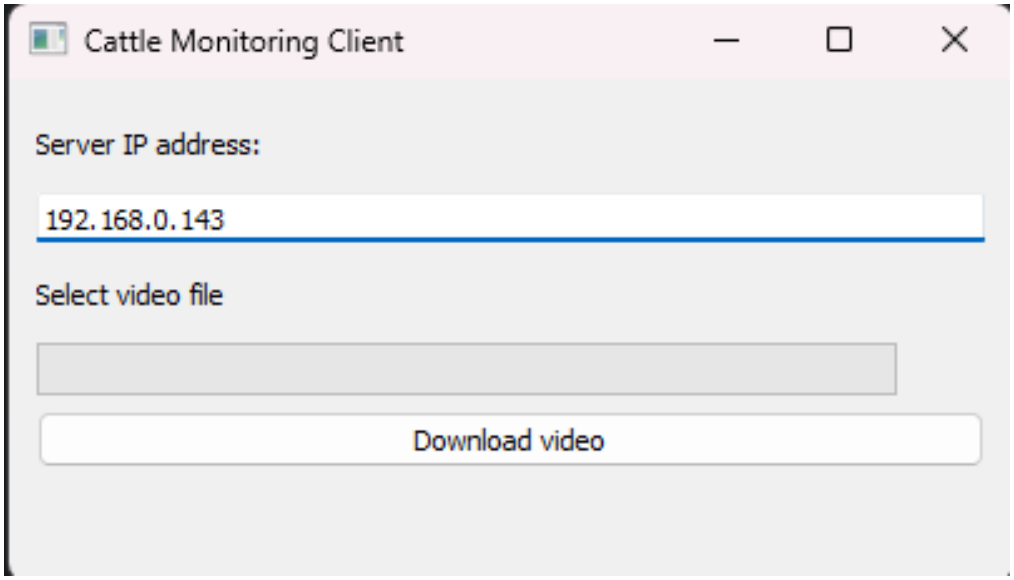



Рисунок 3.4 – Головне вікно клієнтського застосунку

Перед початком передачі даних клієнтський застосунок виконує перевірку введених користувачем параметрів. Зокрема, здійснюється валідація IP-адреси сервера та перевірка коректності вибраного файлу. Для запобігання помилковому завантаженню підтримуються лише відеофайли визначених форматів (.mp4, .avi, .mov, .mkv, .wmv, .flv, .webm). Додатково виконується перевірка MIME-типу файлу, що дозволяє переконатися у відповідності його вмісту заявленому формату.

Для забезпечення відображення поточного стану виконання задачі у клієнтському застосунку реалізовано механізм періодичного опитування сервера. Це дозволяє отримувати інформацію про стан обробки без підтримки

постійного мережевого з’єднання між клієнтом та сервером. Отримані від сервера дані використовуються для оновлення елементів графічного інтерфейсу та інформування користувача про хід виконання задачі.

Важливою особливістю клієнтського застосунку є використання механізму багатопотоковості. Операції передавання відеофайлів та взаємодії із сервером можуть виконуватися протягом тривалого часу, що потенційно призводить до блокування графічного інтерфейсу. Для уникнення цієї проблеми використовується клас QThread, який дозволяє виконувати мережеві операції у фоновому потоці. Обмін інформацією між робочим потоком та інтерфейсом користувача реалізовано за допомогою механізму сигналів та слотів PyQt5.

На відміну від серверної частини, клієнтський застосунок не використовує систему керування базами даних. Уся службова інформація зберігається лише протягом виконання програми у вигляді внутрішнього стану інтерфейсу. Постійно зберігаються лише результати обробки, завантажені із сервера. Такий підхід дозволяє зменшити складність клієнтської частини та мінімізувати вимоги до обчислювальних ресурсів користувацького комп’ютера.

Окрему увагу приділено механізмам обробки помилок. Клієнт виконує перевірку коректності введених даних, контролює успішність HTTP-запитів, обробляє ситуації недоступності сервера, відсутності задачі або помилок під час обробки відео. У випадку виникнення помилки користувач отримує відповідне повідомлення через графічний інтерфейс, а виконання фонових операцій завершується коректним чином без аварійного завершення програми.

Тому було реалізовано клієнтський застосунок, у якіому окремі компоненти відповідають за інтерфейс користувача, мережеву взаємодію із сервером, виконання фонових операцій та обробку результатів. Такий підхід дозволяє зменшити зв’язність між модулями, спростити подальший супровід програмного забезпечення та полегшити його розширення новими функціями.

## 3.3 Алгоритм та демонстрація роботи

Після завершення розробки серверної та клієнтської частин було реалізовано повний цикл автоматизованого аналізу відео корів та оцінки їх ваги. Робота системи побудована за принципом клієнт-серверної взаємодії, де клієнтський застосунок відповідає за взаємодію з користувачем та передачу даних, а сервер виконує всі обчислювально складні операції комп'ютерного зору та штучного інтелекту. Загальний алгоритм роботи системи наведено на рисунку 3.5.

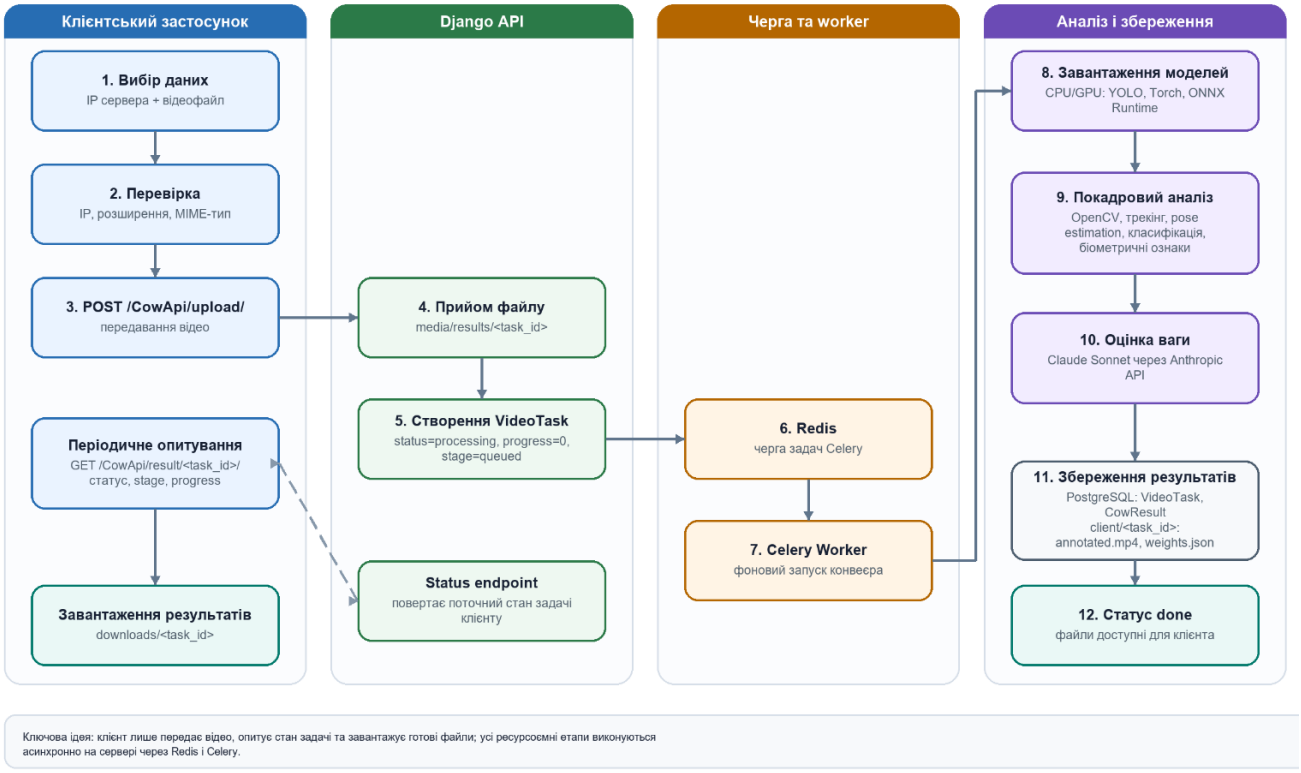



Рисунок 3.5 – Загальний алгоритм роботи системи

Робота системи починається із запуску клієнтського застосунку. Після відкриття головного вікна користувач вводить IP-адресу сервера та обирає відеофайл із локального комп'ютера. Перед початком передавання даних виконується перевірка коректності введеної адреси сервера, розширення

файлу та його MIME-типу. Це дозволяє запобігти надсиланню некоректних даних та зменшити кількість помилок під час роботи системи.

Після успішного проходження перевірок клієнтський застосунок формує HTTP POST-запит до endpoint /CowApi/upload/, через який відеофайл передається на сервер. Після отримання файлу Django-сервер зберігає його у файловій системі, генерує унікальний ідентифікатор задачі та створює відповідний запис у таблиці VideoTask. Одночасно в базі даних фіксується початковий стан обробки, що містить статус processing, значення прогресу 0% та етап виконання queued.

Після створення задачі сервер передає її до системи Celery через брокер повідомлень Redis. На цьому етапі HTTP-запит завершується, а клієнт отримує унікальний ідентифікатор задачі. Завдяки використанню асинхронної архітектури користувачу не потрібно очікувати завершення обробки у відкритому HTTP-з’єднанні, що значно підвищує стабільність роботи системи.

Далі Celery Worker отримує задачу із черги Redis та запускає основний конвеєр аналізу відео. Перед початком обробки виконується автоматичне завантаження моделей комп’ютерного зору.

Після ініціалізації моделей запускається покадрова обробка відеофайлу. За допомогою OpenCV відео відкривається та зчитується послідовність кадрів. Кожен кадр проходить через декілька етапів аналізу. Спочатку модель сегментації та трекінгу виконує виявлення корів і присвоює кожній тварині унікальний ідентифікатор для подальшого відстеження між кадрами. Далі модель оцінки пози визначає ключові точки тіла корови, а класифікаційна модель встановлює породу тварини.

На основі координат ключових точок система розраховує набір біометричних характеристик, необхідних для подальшої оцінки ваги. До них належать довжина тіла, висота окремих частин корпусу, співвідношення між різними анатомічними зонами та інші похідні параметри. Одночасно система за правилом відбирає найбільш інформативний кадр кожної корови, а саме так

щоб вона була в центрі кадру, який в подальшому буде використовується для аналізу мовною моделлю.

У процесі обробки на кадри також наносяться графічні анотації. Для кожної корови відображаються межі об’єкта, ключові точки, ідентифікатор тварини та службова інформація. Оброблені кадри послідовно записуються до нового відеофайлу annotated.mp4.

Після завершення комп’ютерного зору система переходить до етапу оцінки ваги. Для кожної знайденої корови формується спеціалізований запит до моделі Claude Sonnet 4.6 через Anthropic API. Запит містить як зображення корови, так і набір розрахованих біометричних характеристик. На основі отриманих даних мовна модель визначає прогнозовану вагу тварини та рівень впевненості прогнозу. Приклади промптів та отриманих результатів наведені у додатку Б.

Отримані результати зберігаються у таблиці CowResult бази даних PostgreSQL. Крім цього, сервер формує файл weights.json, який містить структуровані результати аналізу всіх виявлених корів. Після успішного завершення обробки статус задачі у таблиці VideoTask змінюється на done, а прогрес встановлюється у значення 100 %.

Паралельно із виконанням аналізу клієнтський застосунок періодично надсилає GET-запити до endpoint /CowApi/result/<task_id>/ для отримання поточного стану задачі. Такий механізм опитування дозволяє відображати користувачу прогрес виконання без необхідності підтримки постійного мережевого з’єднання. Під час роботи сервер повертає інформацію про поточний етап виконання, відсоток завершення та статус задачі.

Після отримання статусу done клієнт автоматично завантажує сформовані сервером результати. До них належать анотоване відео annotated.mp4 та файл weights.json із результатами оцінки ваги. Завантажені файли зберігаються локально у каталозі downloads/<task_id>, після чого інформація про розташування файлів відображається користувачу у графічному інтерфейсі.

Для демонстрації роботи системи було проведено тестове завантаження відеозапису з коровами через клієнтський застосунок. Початковий вигляд головного вікна програми наведено на рисунку 3.4.

Також нас є можливість самостійно ввести IP адресу серверу, якщо він буде випадково зтертий, або наш веб-сервер почне працювати на іншому сервері.

Після вибору відеофайлу та запуску аналізу користувач може спостерігати поточний прогрес виконання задачі та її стан, а після завершення система автоматично завантажує результати та відображає шляхи до сформованих файлів (рис. 3.6, 3.7).

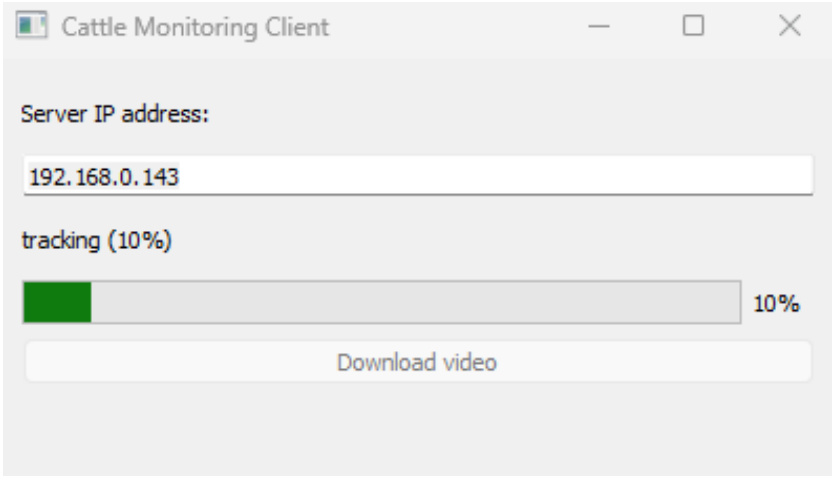



Рисунок 3.6 – Відображення прогресу аналізу відео

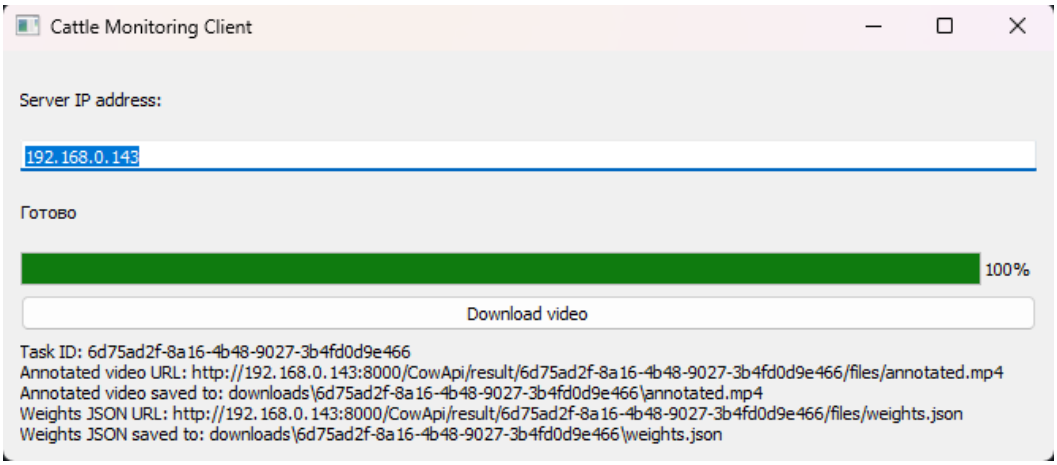
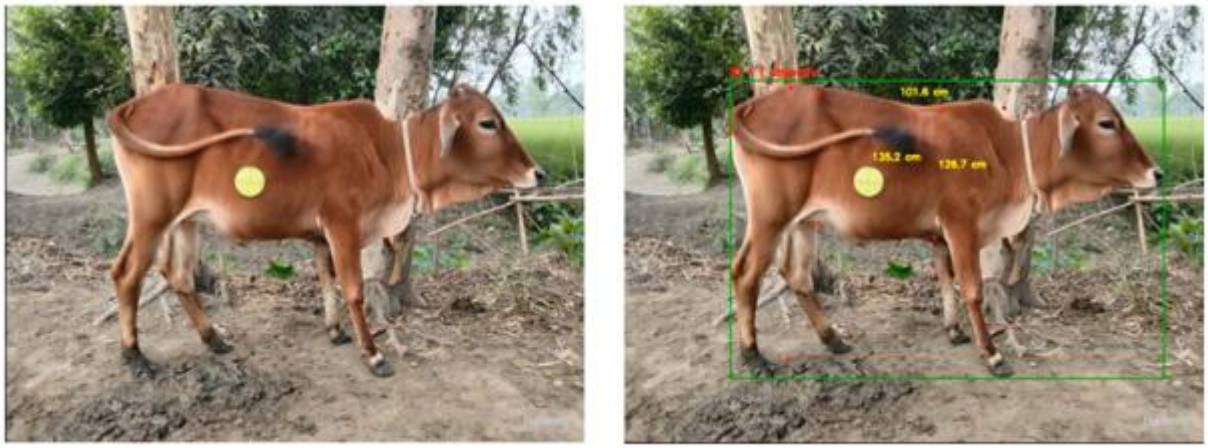



Рисунок 3.7 – Результати роботи клієнтського застосунку

На рисунку 3.8 (а, б) представлено приклад одного і того ж кадру відправленого і отриманого відео. Як бачимо тут вже відображаються знайдені корови, ключові точки та порода.



(а) (б)

Рисунок 3.8 – Порівняння кадрів на відео:
(а) кадр чистого відео, який відправив користувач;
(б) кадр з отриманного анотованого відео

Таким чином, реалізована система забезпечує повний автоматизований цикл обробки відеоматеріалів із коровами — від завантаження вихідного файлу до отримання готових результатів оцінки ваги. Використання асинхронної архітектури на базі Django, Celery та Redis дозволяє виконувати тривалу обробку без блокування користувацьких запитів, а інтеграція моделей комп’ютерного зору та великих мовних моделей забезпечує високий рівень автоматизації процесу аналізу.

# ВИСНОВКИ

У рамках кваліфікаційної роботи була повноцінно розроблена система для автоматизованого аналізу відеозаписів корів та оцінки їх живої маси. Реалізована архітектура поєднує моделі комп'ютерного зору з великою мовною моделлю, що забезпечує оцінку ваги на основі отриманих біометричних характеристик і зображення тварини.

Запропоноване рішення є особливо доцільним для невеликих та середніх фермерських господарств, які не мають можливості придбати дороге спеціалізоване обладнання для автоматичного зважування худоби. Крім того, система може використовуватися на сімейних фермах, приватних господарствах та невеликих молочних підприємствах, де регулярне ручне зважування тварин є трудомістким або економічно невигідним.

Разом із тим існують перспективи подальшого вдосконалення розробленої системи. Одним із напрямів розвитку є оптимізація взаємодії з великою мовною моделлю шляхом перенесення частини інструкцій із користувацького запиту до системного промпту. Потенційно це може зменшити кількість передаваних токенів під час кожного звернення до Claude Sonnet, що потенційно знизить вартість використання API та скоротить час обробки запитів.

Іншим перспективним напрямом є використання двох камер одночасно: камери бокового огляду та камери верхнього огляду. У поточній реалізації оцінка ваги виконується переважно на основі інформації, отриманої з бокового ракурсу, що обмежує можливість точного визначення ширини та повноти тіла тварини. Додавання верхнього ракурсу дозволить отримати додаткові просторові характеристики корпусу корови, покращити оцінку її вгодованості та підвищити точність прогнозування ваги. У перспективі це може стати основою для побудови більш точних тривимірних моделей тварин та подальшого зменшення похибки визначення живої маси.

Таким чином, розроблена система демонструє практичну придатність для автоматизації контролю вагових показників великої рогатої худоби та може бути використана як доступне програмне рішення для фермерських господарств різного масштабу з можливістю подальшого розширення функціональних можливостей і підвищення точності оцінювання.

Результати роботи апробовано у вигляді статей під час X Міжнародної науково-практичної конференції «Science of post-industrial society: globalization and transformation processes» [33], тез доповідей під час VII Міжнародної науково-практичної конференції «Theoretical and practical aspects of modern scientific research» [34] та Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті».

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Antognoli, V., Presutti, L., Bovo, M., Torreggiani, D., & Tassinari, P. (2025). Computer vision in dairy farm management: A literature review of current applications and future perspectives. *Animals*, *15*(17), 2508.

2. Hao, W., Ren, C., Han, M., Zhang, L., Li, F., & Liu, Z. (2023). Cattle body detection based on yolov5-ema for precision livestock farming. *Animals*, *13*(22), 3535.

3. Redmon, J., & Farhadi, A. (2018, 8 квітня). *YOLOv3: An incremental improvement*. arXiv.org.

4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

5. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149.

6. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. (2015, 8 грудня). *SSD: Single shot multibox detector*. arXiv.org.

7. Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., & Ding, G. (2025, 10 березня). *YOLOE: Real-time seeing anything*. arXiv.org.

8. Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2023, 17 квітня). *DETRs beat yolos on real-time object detection*. arXiv.org.

9. Guan, Z., Wang, Z., Zhang, G., Li, L., Zhang, M., Shi, Z., & Jiang, N. (2025). Multi-object tracking review: Retrospective and emerging trend. *Artificial Intelligence Review*, *58*(8).

10. Aharon, N., Orfaig, R., & Bobrovsky, B.-Z. (2022, 29 червня). *BoT-SORT: Robust associations multi-pedestrian tracking*. arXiv.org.

11. Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2021, 13 жовтня). *ByteTrack: Multi-object tracking by associating every detection box*. arXiv.org.

12. Araújo, V., Rili, I., Gisiger, T., Gambs, S., Vasseur, E., Cellier, M., & Diallo, A. (2025, 3 січня). *AI-Powered cow detection in complex farm environments*. arXiv.org.

13. Long, T., Yu, R., You, X., Shen, W., Wei, X., & Gu, Z. (2025). FSCA-YOLO: An enhanced yolo-based model for multi-target dairy cow behavior recognition. *Animals*, *15*(17), 2631.

14. What is CattleEye?. *CattleEye*. URL: https://cattleeye.com/en-gb/what-we-do/what-is-cattleeye (дата звернення 14.05.2026).

15. *Kaggle: The world's AI proving ground*. (б. д.). Kaggle: The World's AI Proving Ground. URL: https://www.kaggle.com/ (дата звернення15.04.2026).

16. Cattle weight detection model + dataset (12k~). *Kaggle: The World's AI Proving Ground*. URL: https://www.kaggle.com/datasets/sadhliroomyprime/cattle-weight-detection-model-dataset-12k/data (дата звернення15.04.2026).

17. Medical Image Segmentation Review: The Success of U-Net / R. Azad та ін. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2024. С. 1–20.

18. Ultralytics. (2023a, 12 листопада). *Home*. Home - Ultralytics YOLO Docs. URL: https://docs.ultralytics.com/ (дата звернення 14.05.2026).

19. YOLOE: Real-Time Seeing Anything | Ultralytics Docs. *Ultralytics Docs*. URL: https://docs.ultralytics.com/models/yoloe (дата звернення 14.05.2026).

20. *Roboflow: Computer vision tools for developers and enterprises*. (б. д.). Roboflow: Computer vision tools for developers and enterprises. https://roboflow.com/ (дата звернення 14.05.2026).

21. Cow Computer Vision Dataset // Roboflow Universe. URL: https://universe.roboflow.com/yoloooo-zg39u/cow-q40wa (дата звернення 17.05.2026).

22. Tan M., Le Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv.org*.

23. Ultralytics. (2023b, 12 листопада). *Image classification*. Home - Ultralytics YOLO Docs. URL: https://docs.ultralytics.com/tasks/classify/

24. Training data-efficient image transformers & distillation through attention / H. Touvron та ін. *PMLR*.

25. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows / Z. Liu та ін. *arXiv.org*.

26. A ConvNet for the 2020s / Z. Liu та ін. *arXiv.org*.

27. Таблиця виміру великої рогатої худоби ⊳ Погляд UA. *Погляд UA*. URL: https://pogliad.ua/tablyczya-vymiru-velykoyi-rogatoyi-hudoby/ (дата звернення 14.05.2026).

28. Protect&Feed - комбикорм для животноводства. (2022, 27 січня). *Как определить вес крс. определение веса телят и коров методом взятия промеров* [Відео]. YouTube. https://www.youtube.com/watch?v=iFs6xtiySnc

29. Bentéjac C., Csörgő A., Martínez-Muñoz G. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*. 2020.

30. Yakovleva, O., Matúšová, S., Liubchenko, V., & Maksimov, H. (2025). Innovative solutions based on AI in education, on the example of generating multimodal lecture notes. Scientific Journal of Bratislava University of Economics and Management «Public Administration and Regional Development, Economics, Management and Marketing», vol. 21, no. 1, pp.140–163.

31. Кульмінський, Я. К., & Любченко, В. А. (2025). Застосування мультимодальних моделей ШІ для автоматизованої оцінки інтерфейсів користувача. *Development of science: theories, methodology, practice and technologies: матеріали IX Міжнародної науково-практичної конференції* (с. 76–80). International Science Group.

32. Klette R. Cameras, Coordinates, and Calibration. *Undergraduate Topics in Computer Science*. London, 2014. С. 215–243.

33. Потапов В., Любченко В. Сучасні методи класифікації та кластеризації в сфері комп'ютерного зору. *Grail of Science*. 2026. № 63. С. 725–730.

34. Potapov V. Modern methods of object detection in computer vision. *Theoretical and practical aspects of modern scientific research* / chair V. Liubchenco. 2026. C. 151–154.