

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ, ОБЛІКУ ТА ФІНАНСІВ

Кафедра комп'ютерних технологій і моделювання систем

Кваліфікаційна робота
на правах рукопису

Новік Владислав Сергійович
(прізвище, ім'я, по батькові здобувача освіти)

УДК 004.853:636.2.03

КВАЛІФІКАЦІЙНА РОБОТА

Розробка методу автоматизованого визначення вгодованості великої рогатої
худоби з використанням глибокого навчання

(тема роботи)

122 Комп'ютерні науки

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня магістр

кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело

Новік В. С.

(підпис, ініціали та прізвище здобувача вищої освіти)

Керівник роботи

Веретюк С. М.

(прізвище, ім'я, по батькові)

К.Т.Н

(науковий ступінь, вчене звання)

Житомир – 2025

Висновок кафедри комп'ютерних технологій і моделювання систем:

за результатами попереднього захисту: _____

Протокол засідання кафедри комп'ютерних технологій і моделювання систем
№ _____ від « _____ » _____ 20__ р.

Завідувач кафедри комп'ютерних технологій і моделювання систем

к.п.н., доцент

(науковий ступінь, вчене звання)

« _____ » _____ 20__ р.

(підпис)

М. О. Ковальчук

(прізвище, ім'я, по батькові)

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)
(прізвище, ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

лаборант кафедри

(науковий ступінь, вчене звання)

(підпис)

В. В. Корольчук

(прізвище, ім'я, по батькові)

АННОТАЦІЯ

Новік В. С. Розробка методу автоматизованого визначення вгодованості великої рогатої худоби з використанням глибокого навчання – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота на здобуття освітнього ступеня магістр за спеціальністю 122 – Комп’ютерні науки. – Поліський національний університет, Житомир, 2025.

Обсяг кваліфікаційної роботи: 41 сторінок (23 – рисунків, 10 – таблиць, 2 – додатки, 46 – джерел).

Ключові слова: оцінка вгодованості тварин, комп’ютерний зір, сегментація зображень, глибоке навчання, нейронні мережі, обробка зображень, мобільні технології.

Кваліфікаційну роботу присвячено розробці методу та системи для автоматизованого визначення вгодованості великої рогатої худоби, що спрямовано на вирішення проблеми відсутності швидких та об’єктивних інструментів оцінки стану тварин без залучення сторонніх експертів.

Програмна реалізація здійснена із застосуванням сучасних алгоритмів комп’ютерного зору: для підготовки даних використано метод ітеративної псевдо-розмітки на базі YOLO11x та SAM2.1, а безпосередня оцінка виконується нейромережею Mask R-CNN з архітектурою Swin Transformer. Система реалізована як клієнт-серверний комплекс, що включає мобільний застосунок для Android, серверну частину на Node.js для обробки запитів та базу даних для збереження історії вимірювань.

Практичне значення отриманих результатів полягає у можливості впровадження неінвазивного моніторингу вгодованості, що дозволяє фермерським господарствам оптимізувати раціони годівлі, покращувати добробут тварин та підвищувати загальну рентабельність виробництва.

SUMMARY

Novik V. S. Development of a method for automated determination of cattle body condition using deep learning – Qualification work as a manuscript.

Master’s qualification thesis for the specialization 122 – Computer Science. – Polissia National University, Zhytomyr, 2025.

Thesis length: 41 pages (232 figures, 10 tables, 2 appendices, 46 references).

Keywords: animal body condition scoring, computer vision, image segmentation, deep learning, neural networks, image processing, mobile technologies.

The thesis is devoted to the development of a method and system for the automated determination of cattle body condition. The study aims to address the lack of rapid and objective tools for assessing animal status without the need for external experts.

The software implementation utilizes modern computer vision algorithms: an iterative pseudo-labeling method based on YOLO11x and SAM2.1 was used for data preparation, while the direct assessment is performed by a Mask R-CNN neural network with a Swin Transformer architecture. The system is realized as a client-server complex comprising an Android mobile application, a Node.js backend for request processing, and a database for storing measurement history.

The practical value of the obtained results lies in the potential for implementing non-invasive body condition monitoring. This allows farms to optimize feeding rations, improve animal welfare, and increase overall production profitability.

АННОТАЦІЯ.....	3
SUMMARY.....	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1. Теоретичні основи автоматизованого визначення вгодваності ВРХ....	8
1.1 Огляд методів оцінювання вгодваності великої рогатої худоби.....	8
1.2 Аналіз існуючих систем комп'ютерного зору в тваринництві.....	10
1.3 Архітектури глибокого навчання для детекції та сегментації об'єктів (YOLO, Mask R-CNN, трансформери).....	12
1.4 Методи pseudo labeling та їх застосування.....	15
Висновки до першого розділу.....	17
РОЗДІЛ 2. Постановка задачі та методичні основи дослідження.....	18
2.1 Постановка задачі автоматизованого визначення вгодваності ВРХ.....	18
2.2 Вимоги до системи та критерії ефективності (точність детекції, швидкість обробки, зручність використання).....	19
2.3 Методи збору та підготовки даних	21
2.4. Метрики оцінювання якості моделей (mAP, IoU, F1-score).....	23
Висновки до другого розділу.....	27
РОЗДІЛ 3. Розробка методу та реалізація програмного комплексу.....	28
3.1 Архітектура системи (клієнт-серверна модель).....	28
3.2. Розробка моделі глибокого навчання.....	29
3.3 Серверна частина та база даних.....	31
3.4. Мобільний застосунок.....	32
Висновки до третього розділу.....	33
Розділ 4. Експериментальні дослідження та тестування.....	35
4.1 Постановка експериментів та тестові датасети.....	35
4.2. Аналіз результатів роботи моделей.....	36
4.3. Стрес-тестування серверної частини.....	37
Висновки до четвертого розділу.....	39
Висновки.....	40
ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	42
ДОДАТКИ.....	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BCS (Body Condition Score) — показник вгодованості великої рогатої худоби.

CNN (Convolutional Neural Network) — згорткова нейронна мережа.

YOLO (You Only Look Once) — сімейство моделей для детекції об'єктів у реальному часі.

SAM (Segment Anything Model) — модель для генерації сегментаційних масок.

Mask R-CNN — модель інстанс-сегментації.

Swin Transformer — трансформерна архітектура для обробки зображень.

mAP (mean Average Precision) — середня точність за всіма класами.

IoU (Intersection over Union) — метрика якості сегментації.

COCO — формат анотацій для сегментації та детекції.

Pascal VOC — формат анотацій для детекції об'єктів.

API (Application Programming Interface) — інтерфейс взаємодії між компонентами системи.

GPU (Graphics Processing Unit) — графічний процесор для прискорення обчислень.

JSON (JavaScript Object Notation) — формат обміну структурованими даними.

ВСТУП

Сучасний аграрний сектор України, що функціонує в умовах повномасштабної війни та паралельної відбудови, потребує інноваційних рішень для підвищення ефективності тваринництва. Оцінка вгодованості великої рогатої худоби є критично важливою складовою управління поголів'ям, однак традиційні методи залишаються суб'єктивними, трудомісткими та викликають стрес у тварин. У цих умовах впровадження методів глибокого навчання та комп'ютерного зору є інноваційним підходом для створення неінвазивних, об'єктивних і масштабованих систем моніторингу стану ВРХ.

Мета роботи: Розробити метод автоматизованої оцінки вгодованості великої рогатої худоби за зображеннями, заснований на сегментаційній архітектурі Mask R-CNN з ієрархічною базовою моделлю Swin Transformer, з метою підвищення об'єктивності контролю стану тварин, оптимізації ресурсів годівлі та сприяння формуванню сталого аграрного підприємництва в умовах повномасштабної війни та паралельної відбудови аграрного сектору України.

Завдання дослідження: Виконати аналіз предметної області та існуючих методів оцінки вгодованості, розробити архітектуру нейромережі (Mask R-CNN з Swin Transformer) та скласти датасет з полігональними масками за допомогою iterative pseudo labeling (YOLOv11x та SAM 2.1 L). Також здійснити практичну реалізацію програмного комплексу, включаючи мобільний застосунок, Inference Server та Backend Server, та провести експериментальне порівняння архітектур для оцінки їхньої точності, робастності та продуктивності.

Об'єкт дослідження: Процеси моніторингу та оцінки фізичного стану великої рогатої худоби за допомогою комп'ютерного зору.

Предмет дослідження: Математичні, алгоритмічні та інформаційні моделі для автоматизованого визначення вгодованості ВРХ на основі архітектур глибокого навчання (Mask R-CNN, Swin Transformer) та їхнє застосування в практиці сталого тваринництва.

Огляд сучасного стану проблеми: Існуючі автоматизовані системи оцінки BCS мають обмеження щодо точності та стійкості до умов зйомки, а глибокі згорткові мережі часто втрачають просторову роздільну здатність, ускладнюючи розрізнення тонких морфологічних деталей. Запропонований підхід з Mask R-CNN та Swin Transformer націлений на забезпечення більш точної сегментації та класифікації завдяки механізму багаторівневої уваги.

Методи дослідження: Включають методи глибокого навчання (Mask R-CNN, Swin Transformer), алгоритми комп'ютерного зору (детекція, сегментація, класифікація), методи обробки та анотації даних (iterative pseudo labeling з YOLO та SAM), математичного моделювання та експериментальні методи. Програмні засоби: Python, PyTorch, MMDetection, OpenCV, NumPy та GPU з підтримкою CUDA.

Наукова новизна: Уперше запропоновано та досліджено застосування Swin Transformer як базової моделі архітектури Mask R-CNN для задачі автоматизованої оцінки вгодованості великої рогатої худоби. Такий підхід забезпечує ієрархічне представлення ознак та механізм багаторівневої уваги (hierarchical self-attention), що підвищує точність локалізації контурів тіла та розрізнення тонких морфологічних змін порівняно з традиційними CNN-бекбонами (ResNet-50, ResNeXt-101). Проведено кількісний порівняльний аналіз ефективності різних бекбон-архітектур Mask R-CNN у контексті класифікації категорій BCS та якості сегментації.

Практичне значення: Розроблена система є цінним інструментом для формування ефективного, екологічно свідомого та кібербезпечного аграрного бізнесу. Вона дозволяє оптимізувати витрати кормів (потенційна економія до 20-30%[1]), підвищити якість та безпеку продукції, забезпечити сталість малих і середніх ферм шляхом неінвазивного та об'єктивного моніторингу вгодованості великої рогатої худоби. Рішення сприяє своєчасному виявленню проблем зі здоров'ям тварин та забезпечує основу для індивідуалізованої годівлі, що є критично важливим в умовах поствоєнного відновлення України та глобальних кліматичних змін.

РОЗДІЛ 1. Теоретичні основи автоматизованого визначення вгодованості ВРХ

1.1 Огляд методів оцінювання вгодованості великої рогатої худоби

Вгодованість великої рогатої худоби (ВРХ) є критично важливим параметром, що відображає якість годівлі, стан здоров'я, репродуктивну ефективність та загальну продуктивність тваринництва. Неоптимальна вгодованість може призвести до значних економічних втрат, таких як зниження надоїв, проблем з обміном речовин, імунодефіциту та падіння відсотка запліднення. Контроль цього показника є фундаментальним для ефективного та сталого ведення тваринницького господарства. Протягом тривалого часу для оцінки вгодованості використовувалися різні методи, кожен з яких має свої переваги та суттєві обмеження.

Традиційні методи оцінювання:

1. **Візуальна оцінка за шкалою BCS[2]:** Це найпоширеніший метод, що передбачає візуальний та пальпаційний огляд тварини. Фермери та зоотехніки оцінюють ступінь покриття кісток жиром у ключових анатомічних областях (наприклад, хребет, маклаки, сідничні горби) за стандартизованою шкалою (найчастіше від 1 до 5 або від 1 до 9).

Обмеження: Висока суб'єктивність, оскільки різні оператори можуть давати різні оцінки, залежність від досвіду та кваліфікації персоналу. Цей метод не дозволяє точної цифрової фіксації результатів, що ускладнює відстеження динаміки та аналіз даних.

2. **Метод Клювер-Штрауха[3]:** Використовує заміри обхватів тіла (наприклад, обхват грудей) для розрахунку живої маси тварини.

Обмеження: Має похибку до 15% при визначенні маси, вимагає фізичного контакту з твариною, що може викликати стрес, та є трудомістким процесом проведення замірів.

3. **Контрольне зважування[4]:** Дає точні показники живої маси тварини.

Обмеження: Вимагає високоякісного обладнання та спеціальної інфраструктури (ваги, прогони). Процедура зважування є стресовою для тварини, що може негативно впливати на її добробут та продуктивність.

Сучасні та автоматизовані методи оцінювання (та їхні обмеження):

З розвитком технологій комп'ютерного зору та машинного навчання з'явилися спроби автоматизації процесу оцінки вгодованості.

1. **3D-сканування:** Цей метод використовує 3D-камери (наприклад, Kinect) для створення тривимірної моделі тіла тварини, з якої потім екстрагуються морфологічні ознаки для оцінки вгодованості великої рогатої худоби.

- *Обмеження:* Вимагає дорогого обладнання та спеціалізованого програмного забезпечення. Залишається складною у використанні в польових умовах та чутливою до умов зйомки (освітлення, ракурсу, забруднень).

2. **Методи на основі 2D-зображень та глибокого навчання (CNN-бекбони)** [5]: Застосування згорткових нейронних мереж (CNN) у рамках архітектур детекції та класифікації (наприклад, R-CNN, YOLO) дозволило підвищити точність локалізації та класифікації порівняно з класичними алгоритмами комп'ютерного зору.

- *Обмеження:* Хоча CNN-бекбони значно покращили точність, згорткові ієрархії часто втрачають просторову роздільну здатність на верхніх рівнях. Це призводить до грубих карт ознак і недостатнього фокусування на дрібних морфологічних деталях, які є критично важливими для коректного розмежування близьких категорій BCS.

Ці традиційні та ранні автоматизовані методи часто призводять до несвоєчасного виявлення проблем з годівлею, нераціонального використання кормів, зниження продуктивності тварин та збільшення витрат на утримання. Помилки в оцінці BCS можуть мати критичні наслідки, такі як втрати надоїв до 2,3 кг на день[6] при недооцінці вгодованості або надмірні витрати на корми при переоцінці.

Таким чином, незважаючи на постійний прогрес у розробці систем оцінки вгодованості ВРХ, існуючі підходи все ще мають значні обмеження щодо точності, об'єктивності, масштабованості та зручності використання в реальних фермерських умовах. Це обумовлює необхідність розробки нових, більш досконалих та ефективних методів, здатних забезпечити детальний морфологічний аналіз та точну класифікацію ВСХ, що є передумовою для впровадження систем точного тваринництва.

1.2 Аналіз існуючих систем комп'ютерного зору в тваринництві

Розвиток комп'ютерного зору (КЗ) та штучного інтелекту трансформує тваринництво, дозволяючи автоматизувати моніторинг та оптимізувати виробничі процеси. Системи КЗ аналізують візуальні дані з камер, забезпечуючи неінвазивний контроль та підвищуючи добробут тварин.

Основні напрямки застосування КЗ у тваринництві[7]:

1. **Моніторинг здоров'я та раннє виявлення захворювань**[8]: Системи КЗ постійно аналізують зміни у ході, поставі, поведінці або зовнішньому вигляді тварин для виявлення ознак кульгавості, маститу, респіраторних захворювань та інших патологій. Використовуються стандартні, інфрачервоні та 3D-камери, а основою аналізу слугують моделі глибокого навчання (CNN, Mask R-CNN, YOLO).
2. **Оцінка вгодованості тіла (ВСХ) та ваги**[9]: Системи КЗ надають об'єктивну та неінвазивну оцінку ВСХ та ваги, що дозволяє оптимізувати раціони годівлі та покращувати репродуктивні показники. Сучасні підходи використовують 2D- та 3D-зображення з моделями, такими як YOLOv8 та Mask R-CNN, для вилучення морфологічних ознак.
3. **Аналіз поведінки та виявлення еструсу**[10]: Моніторинг поведінки тварин дозволяє виявляти стрес, агресію, зміни у режимах годування/пиття та ознаки еструсу для оптимізації репродуктивного циклу. Системи аналізують рухові

патерни та взаємодії, досягаючи високої точності (наприклад, YOLOv8 для виявлення еструсу).

4. **Ідентифікація та відстеження тварин**[11]: Системи КЗ розпізнають тварин за унікальними візуальними ознаками (морда, візерунки шерсті), що є фундаментальним для персоналізованого управління стадом. Використовуються алгоритми розпізнавання облич та відстеження об'єктів на базі глибокого навчання.
5. **Моніторинг процесу отелення**[12]: Автоматизовані системи КЗ відстежують вагітних корів, виявляючи ознаки наближення отелення та сповіщаючи фермерів, що покращує результати отелення.

Сучасні системи комп'ютерного зору у тваринництві базуються на передових методах глибокого навчання, таких як згорткові нейронні мережі та трансформери, а також на потужних архітектурах для детекції і сегментації об'єктів, включно з YOLO та Mask R-CNN. Для отримання більш точних даних вони інтегруються з різними сенсорами, зокрема тепловізорами та 3D-камерами, що дозволяє оцінювати стан тварин у трьох вимірах і в різних умовах освітлення.

Впровадження таких технологій відкриває нові можливості для автоматизованого моніторингу в реальному часі. Фермери отримують більш надійні дані для прийняття рішень, що сприяє підвищенню добробуту тварин, збільшенню продуктивності та ефективності використання ресурсів. Крім того, системи легко масштабуються та можуть бути адаптовані до різних розмірів господарств, від невеликих ферм до великих агропідприємств.

Водночас існують певні виклики. Системи комп'ютерного зору залишаються чутливими до змін зовнішнього середовища, таких як нерівномірне освітлення або часткове перекриття об'єктів. Значні витрати на обладнання та потреба у великих, ретельно анованих датасетах ускладнюють їх впровадження. Крім того, інтеграція таких рішень у існуючу інфраструктуру та стандартизація процесів залишаються важливими питаннями, над якими працюють дослідники та практики.

1.3 Архітектури глибокого навчання для детекції та сегментації об'єктів (YOLO, Mask R-CNN, трансформери)

Глибоке навчання здійснило революцію у комп'ютерному зорі, надавши потужні інструменти для автоматизації детекції та сегментації об'єктів. Ці технології мають величезне значення для тваринництва, дозволяючи точно ідентифікувати тварин, моніторити їхній стан та оцінювати морфологічні показники, такі як вгодованість.

YOLO[13] – це сімейство архітектур глибокого навчання, відоме своєю високою швидкістю та ефективністю у задачах детекції об'єктів. Будучи одностадійним детектором, YOLO обробляє все зображення одночасно. Основні принципи YOLO (див. рис. А.1, дод. А) включають: розбиття зображення на сітку, паралельне передбачення обмежувальних рамок, оцінок впевненості та ймовірностей класів для кожної комірки, а також використання немаксимального пригнічення (NMS) для фільтрації дублікатів. Перевагами YOLO є висока швидкість обробки в реальному часі та комплексне навчання, що дозволяє розуміти контекст. Обмеження ранніх версій включали проблеми з точністю локалізації та складністю виявлення малих, близько розташованих об'єктів. Сучасні версії, як YOLOv11x-seg, інтегрують можливості сегментації екземплярів.

Еволюція сімейства YOLO демонструє постійне покращення балансу між точністю та швидкістю обробки. Порівняльний аналіз продуктивності різних версій YOLO на датасеті COCO (див. рис. А.2, дод. А) показує, що YOLOv11 досягає найкращих результатів серед усіх версій. YOLOv11x забезпечує найвищу точність (mAP50-95 ~55%) при латентності близько 12 мс, що значно перевершує попередні версії: YOLOv10 досягає максимум ~53% mAP при подібній швидкості, YOLOv9 показує ~54.5% при значно вищій латентності ~16 мс, а YOLOv8 демонструє результати, близькі до YOLOv10. Альтернативні архітектури, такі як YOLOX (~51% mAP при 12 мс) та EfficientDet (низька швидкість при помірній точності), значно поступаються YOLOv11.

Різноманітність конфігурацій YOLOv11 дозволяє гнучко підбирати оптимальне рішення залежно від конкретних потреб у тваринництві:

- **YOLOv11n** (nano) — найшвидша конфігурація з латентністю < 2 мс та mAP ~39%, ідеальна для безперервного моніторингу поведінки та виявлення еструсу в реальному часі на пристроях з обмеженими обчислювальними ресурсами;
- **YOLOv11s/m** (small/medium) — оптимальний баланс із mAP ~47-51% при латентності 3-5 мс, підходить для оцінки вгодованості тіла (BCS), ідентифікації та відстеження тварин;
- **YOLOv11l/x** (large/extra-large) — максимальна точність (mAP ~53-55%) при латентності 8-12 мс, рекомендовані для високоточного моніторингу здоров'я, раннього виявлення захворювань (кульгавість, мастит, респіраторні захворювання) та моніторингу процесу отелення, де критична точність детекції тонких ознак.

Mask R-CNN – це потужна двостадійна архітектура для детекції об'єктів та сегментації екземплярів, що забезпечує високу точність локалізації та деталізоване виділення контурів. Вона є "мозком" системи для визначення вгодованості ВРХ, оскільки не просто знаходить корову, а й точно окреслює її форму. Основні компоненти Mask R-CNN (див. рис. А.3, дод. А) включають:

1. **Backbone**: Початкова згорткова мережа (або трансформер), що витягує ієрархічні ознаки.
2. **Feature Pyramid Network (FPN)**: Комбінує ознаки різних масштабів для ефективної детекції об'єктів різного розміру.
3. **Region Proposal Network (RPN)**: Генерує прямокутні області (RoI), де потенційно знаходяться об'єкти.
4. **RoI Align**: Вирівнює ділянки ознак для точного збереження просторових координат.
5. **Heads (Голови передбачень)**: Паралельні модулі для класифікації класу об'єкта, уточнення координат обмежувальної рамки та генерації бінарної

маски для кожного пікселя об'єкта (Mask Head). Переваги Mask R-CNN полягають у високій точності детекції та сегментації, а також здатності генерувати деталізовані полігональні маски. Основним обмеженням є його обчислювальна складність, що робить його повільнішим за одностадійні детектори.

Трансформери в комп'ютерному зорі спочатку були розроблені для обробки природної мови, використовуючи механізм само-уваги (self-attention) для моделювання залежностей у даних. Адаптація трансформерів до комп'ютерного зору (Vision Transformers, ViT) відкрила нові перспективи для аналізу зображень. Vanilla ViT розглядає зображення як послідовність "патчів", що дозволяє захоплювати глобальні залежності. Однак, він має високу обчислювальну складність для зображень високої роздільної здатності.

Swin Transformer (див. рис. А.4, дод. А) є важливою інновацією, що долає обмеження vanilla ViT, зокрема високу обчислювальну складність та відсутність ієрархічного представлення ознак. Swin Transformer будує ієрархічні карти ознак за допомогою блоків Patch Merging та використовує ключовий механізм Shifted Window Attention. Цей механізм обчислює само-увагу локально в межах непересічних вікон, а потім зсуває ці вікна на наступному шарі, забезпечуючи ефективний обмін інформацією між різними частинами зображення без зростання обчислювальних витрат.

Математично обчислення само-уваги у межах вікна в архітектурі Swin Transformer описується наступною формулою:

$$Attention(Q, K, V) = SoftMax\left(\frac{QK^T}{\sqrt{d}} + B\right) * V \quad (1)$$

Де:

- $Q, K, V \in R^{M^{2 \times d}}$ - матриці запиту (query), ключа (key) та значення (value)
- M^2 - кількість патчів у вікні;
- D - розмірність векторів запиту/ключа;

- $B \in R^{M^2 \times M^2}$ - матриця відносного позиційного зміщення (relative position bias), яка дозволяє моделі враховувати просторові зв'язки між патчами всередині локального вікна.

1.4 Методи pseudo labeling та їх застосування

Pseudo-labeling (псевдо-розмітка)[18] – це потужний напівконтрольований підхід у глибокому навчанні, який дозволяє ефективно використовувати великі обсяги нерозмічених даних. Суть методу полягає в тому, що вже навчена модель використовується для автоматичної генерації міток (псевдо-міток) для великого пулу нерозмічених даних. Ці псевдо-мітки потім комбінуються з оригінальними розміченими даними для подальшого навчання або донавчання (fine-tuning) моделі.

Мотивація та переваги використання Pseudo-Labeling:

1. **Подолання дефіциту розмічених даних:** Розмітка даних для задач комп'ютерного зору, особливо для сегментації екземплярів, де вимагається піксельна точність контурів об'єктів, є одним з найскладніших, дорогих і трудомістких етапів розробки систем. Pseudo-labeling дозволяє значно зменшити залежність від ручної праці анотаторів.
2. **Ефективне використання нерозмічених даних:** У багатьох реальних сценаріях величезні обсяги нерозмічених даних доступні "безкоштовно" (наприклад, необроблені відео з камер спостереження). Цей метод надає можливість ефективно використати цей ресурс для покращення узагальнюючої здатності моделі.
3. **Покращення узагальнюючої здатності моделі:** Додавання псевдо-розмічених даних до навчального набору може допомогти моделі краще вивчити різноманітність даних, зменшити ризик перенавчання на невеликому розміченому наборі та підвищити її робастність (стійкість) до варіацій у реальних умовах.
4. **Зменшення людської суб'єктивності:** У деяких предметних областях, де експертна ручна розмітка може містити елементи суб'єктивності або

невідповідностей, автоматично згенеровані псевдо-мітки можуть забезпечити більш послідовний та об'єктивний підхід до анотації.

Типовий алгоритм псевдо-розмітки включає такі кроки(див. рис. А.5, дод. А):

1. **Початкове навчання моделі-вчителя:** Спершу модель-вчитель навчається на невеликому початковому наборі якісно розмічених даних. Якість цієї моделі є критично важливою, оскільки вона визначатиме якість згенерованих псевдо-міток.
2. **Генерація псевдо-міток:** Навчена модель-вчитель застосовується до великого пулу нерозмічених даних для генерації передбачень, які стають "псевдо-мітками".
3. **Фільтрація та уточнення псевдо-міток):** Це найважливіший етап. Оскільки передбачення моделі можуть бути неідеальними, необхідно відфільтрувати ненадійні псевдо-мітки. Це зазвичай робиться за допомогою порогу впевненості: псевдо-мітки з впевненістю нижче встановленого порогу відкидаються. Для задач детекції та сегментації додатково застосовуються такі методи, як Non-Maximum Suppression (NMS) для усунення дублікатів об'єктів та перевірка валідності згенерованих полігонів (для сегментації).
4. **Комбінування даних:** Відфільтровані та уточнені псевдо-розмічені дані об'єднуються з оригінальним набором вручну розмічених даних.
5. **Донавчання або навчання нової моделі-учня:** На розширеному наборі даних навчається нова модель-учень або донавчається попередня модель-вчитель. Це дозволяє моделі використовувати додаткову інформацію з нерозмічених даних.
6. **Ітеративний процес:** Процес може бути повторений циклічно. Після навчання моделі-учня, вона може стати новою, покращеною моделлю-вчителем для генерації ще якісніших псевдо-міток, дозволяючи моделі поступово покращувати свою продуктивність та якість розмітки.

Незважаючи на значні переваги, використання pseudo-labeling має певні виклики:

- **Поширення помилок:** Якщо початкові псевдо-мітки містять значну кількість помилок, ці помилки можуть накопичуватися та поширюватися під час ітеративного навчання, потенційно погіршуючи кінцеву продуктивність моделі.
- **Упередженість підтвердження:** Модель може стати надмірно впевненою у власних помилкових передбаченнях, що призводить до стагнації навчання.
- **Оптимальний поріг впевненості:** Вибір відповідного порогу впевненості для фільтрації псевдо-міток є ключовим. Занадто високий поріг може відкинути багато корисних даних, а занадто низький – ввести шум. Цей параметр часто потребує експериментального підбору.

Висновки до першого розділу

У першому розділі проаналізовано теоретичні основи автоматизованого визначення вгодованості великої рогатої худоби та сучасні підходи до вирішення цієї задачі.

Встановлено, що традиційні методи оцінки BCS (візуальна оцінка, заміри, зважування) мають критичні недоліки: суб'єктивність, трудомісткість, стресовість для тварин та обмежену масштабованість, що обґрунтовує необхідність розробки автоматизованих рішень.

Огляд архітектур глибокого навчання показав, що YOLO забезпечує швидку детекцію об'єктів, Mask R-CNN - високу точність сегментації екземплярів, а Swin Transformer як backbone перевершує традиційні CNN завдяки ієрархічній структурі та механізму зсунутих вікон, що критично важливо для виявлення тонких морфологічних змін у тілі ВРХ.

Обґрунтовано доцільність застосування методів pseudo-labeling для автоматичного створення високоякісних датасетів з полігональними масками, що дозволяє мінімізувати трудомісткість ручної розмітки при навчанні моделей сегментації екземплярів.

РОЗДІЛ 2. Постановка задачі та методичні основи дослідження

2.1 Постановка задачі автоматизованого визначення вгодованості ВРХ

Задача автоматизованого визначення вгодованості великої рогатої худоби формулюється як розробка інтелектуальної системи, здатної неінвазивно, об'єктивно та з високою точністю оцінювати фізичний стан тварин на основі візуальних даних. Кінцевою метою є присвоєння кожній тварині числового значення BCS відповідно до стандартизованої шкали, що дозволить фермерам приймати обґрунтовані управлінські рішення щодо годівлі, розмноження та здоров'я стада.

Задача автоматизованого визначення вгодованості ВРХ може бути розбита на наступні підзадачі:

1. **Детекція та локалізація ВРХ:** На вхід системи подається зображення, що може містити одну або декілька тварин, а також елементи фону (обладнання, інші тварини, частини приміщення). Перша підзадача полягає у виявленні всіх екземплярів ВРХ на зображенні та визначенні їхнього точного розташування за допомогою обмежувальних рамок.
2. **Сегментація екземплярів ВРХ:** Для кожного виявленого екземпляра ВРХ необхідно створити піксельну маску, яка точно окреслює контур тіла тварини, відокремлюючи її від фону та інших об'єктів.
3. **Класифікація BCS:** На основі сегментованого зображення тіла тварини, система повинна класифікувати її вгодованість до однієї з попередньо визначених категорій BCS (3.25, 3.5, 3.75, 4.0, 4.25 за модифікованою шкалою).
4. **Вилучення та інтерпретація ознак:** Система повинна автоматично імпліцитно вилучати візуальні ознаки, релевантні для оцінки вгодованості, без необхідності їх явного програмування. Це передбачає здатність моделі фокусуватися на ключових анатомічних областях.

Виклики задачі:

- **Складні умови зйомки:** Зображення можуть бути зроблені в різних умовах освітлення (денне, штучне, тіні), з різних ракурсів (збоку, зверху, під кутом), з фоновими перешкодами та оклюзіями (часткове перекриття тварин або обладнанням).
- **Тонкі відмінності між категоріями BCS:** Особливо для середніх категорій (наприклад, між 3.5 та 3.75), візуальні відмінності можуть бути дуже незначними та вимагають високої чутливості системи.
- **Необхідність неінвазивності:** Система повинна працювати без фізичного контакту з твариною, щоб мінімізувати стрес та не впливати на її поведінку та продуктивність.

Таким чином, постановка задачі полягає у розробці робастного, точного та ефективного рішення, яке може замінити суб'єктивні та трудомісткі ручні методи оцінки BCS, сприяючи оптимізації управління стадом та підвищенню рентабельності аграрного підприємства.

2.2 Вимоги до системи та критерії ефективності (точність детекції, швидкість обробки, зручність використання)

Для того, щоб система автоматизованого визначення вгодованості ВРХ була практично цінною та конкурентоспроможною, вона повинна відповідати низці жорстких вимог та демонструвати високу ефективність за ключовими критеріями:

1. **Неінвазивність:** Система повинна функціонувати без фізичного контакту з твариною. Це забезпечує мінімізацію стресу для ВРХ, підвищує її добробут та усуває ризик передачі інфекцій.
2. **Об'єктивність та відтворюваність:** Оцінки BCS, надані системою, повинні бути незалежними від оператора та стабільними при повторних вимірюваннях за однакових умов. Це усуває суб'єктивність ручних методів.
3. **Адаптивність до різних умов:** Система має бути стійкою до варіацій умов зйомки (різні ракурси, освітлення, фони, породи тварин, їхній вік та розмір).
4. **Безпека даних:** Забезпечення конфіденційності та цілісності оброблюваних сільськогосподарських даних.

5. Простота використання: Інтерфейс та функціонал системи повинні бути інтуїтивно зрозумілими для персоналу ферм без глибоких знань у галузі комп'ютерного зору.

Критерії ефективності:

1. Точність детекції та сегментації:

- Для детекції об'єктів: Основними метриками є $mAP@0.5$ та $mAP@0.5:0.95$. Ці метрики відображають, наскільки добре система знаходить об'єкти та точно їх локалізує.
- Для сегментації екземплярів: Використовуються аналогічні метрики $mAP@0.5$ (mask) та $mAP@0.5:0.95$ (mask). Вони оцінюють якість згенерованих полігональних масок.
- Точність класифікації BCS: Визначається як відсоток правильно класифікованих тварин за їхніми категоріями BCS.
- Метрики Precision, Recall, F1-score для кожного класу BCS: Дозволяють детально оцінити ефективність розпізнавання кожної окремої категорії вгодованості.
- Матриця плутанини: Візуалізує, які категорії BCS модель плутає між собою, надаючи цінні відомості для подальшого покращення.

2. Швидкість обробки (Processing Speed):

- Цей критерій відображає, як швидко система може обробляти одне зображення або відеопотік.
- Вимірюється в секундах на зображення або в кадрах за секунду (FPS).

3. Зручність використання (Usability):

- Простота налаштування: Легкість встановлення камер, підключення до обчислювального пристрою та початкового налаштування програмного забезпечення.
- Мінімальні вимоги до калібрування: Система повинна вимагати мінімального калібрування або бути здатною до самокалібрування.

- Інтуїтивний інтерфейс: Зрозумілий для користувача інтерфейс для подачі зображень, відображення результатів та доступу до аналітичних звітів.
- Візуалізація результатів: Наявність функцій візуалізації (наприклад, накладання обмежувальних рамок, масок та міток BCS на зображення) для легкого розуміння роботи системи та її помилок.

2.3 Методи збору та підготовки даних

Якість та обсяг навчальних даних є фундаментальними для успішного навчання моделей глибокого навчання. У даному дослідженні процес підготовки даних включав використання існуючого датасету, його розширену анотацію з застосуванням методів псевдо-розмітки та подальшу аугментацію.

Для навчання моделі було використано датасет Dairy cow body condition score target detection data set[19] із відкритою ліцензією CC BY 4.0, що містить 53 566 зображень великої рогатої худоби. Зображення мають стандартну роздільну здатність 1024×576 пікселів у форматі JPEG та охоплюють п'ять категорій BCS (3.25, 3.5, 3.75, 4.0, 4.25) відповідно до модифікованої шкали. Датасет характеризується помірним дисбалансом класів та різноманітними умовами зйомки (доїльні станки, вигул, різне освітлення, ракурси, фонові елементи), що підвищує робастність моделі.

Оригінальний датасет містив лише обмежувальні рамки (Pascal VOC). Для навчання моделей instance segmentation, як Mask R-CNN, потрібні детальні полігональні маски. Для вирішення цієї ресурсоємної проблеми був розроблений та застосований pipeline ітеративної псевдо-розмітки(див. рис. А.5, дод. А). Цей процес ефективно поєднує можливості передових моделей детекції та сегментації:

Початкова детекція обмежувальних рамок за допомогою YOLOv11x: Попередньо навчена модель YOLOv11x генерувала обмежувальні рамки з Confidence threshold: 0.8 та IoU(Intersection over Union) threshold для NMS: 0.8.

Генерація полігональних масок за допомогою SAM 2.1 Large: Отримані YOLO рамки служили підказками для SAM 2.1 Large, яка генерувала високоякісні детальні полігональні маски тіла ВРХ.

Багатоступенева валідація та фільтрація масок: Згенеровані маски проходили ретельну перевірку на дублікати, валідність полігонів та нормалізацію координат, щоб забезпечити високу якість.

Ітеративне уточнення моделі YOLOv11x-seg: Отримані полігональні маски використовувалися для донавчання сегментаційної моделі YOLOv11x-seg, що дозволяло генерувати ще точніші маски на наступних ітераціях. В результаті цього ітеративного процесу, після фільтрації 2 347 дублікатів та 4 469 невалідних масок, було отримано фінальний набір з 49 097 високоякісних полігональних анотацій у форматі COCO JSON (91.6% від загального обсягу датасету).

Розбиття датасету: Фінальний датасет був розділений на Навчальну вибірку (80%), Валідаційну вибірку (10%) та Тестову вибірку (10%) із збереженням пропорційного розподілу класів (stratified split).

Аугментація даних: Для підвищення узагальнюючої здатності моделі та її стійкості до варіацій у реальних умовах, під час навчання застосовувалися різноманітні техніки "на льоту"(див. рис. А.6, дод. А):

- **RandomChoiceResize:** Випадкове масштабування зображень до одного з розмірів: (896x504), (960x540), (1024x576), (1088x612).
- **RandomFlip:** Випадкове горизонтальне віддзеркалення зображень з ймовірністю $prob=0.5$.
- **PhotoMetricDistortion:** Випадкові фотометричні трансформації, що імітують зміни освітлення (яскравості, контрасту, насиченості, відтінку).
- **Resize (для валідації та тесту):** Для валідаційної та тестової вибірок зображення масштабувалися до фіксованого розміру (1024, 576) із збереженням співвідношення сторін.

2.4. Метрики оцінювання якості моделей (mAP, IoU, F1-score)

Для об'єктивної оцінки продуктивності моделей глибокого навчання у задачах детекції та сегментації об'єктів використовуються стандартизовані метрики. Ці метрики дозволяють кількісно визначити, наскільки добре модель виявляє, локалізує та класифікує об'єкти, а також наскільки точно вона окреслює їхні контури.

Intersection over Union (IoU), або коефіцієнт Жаккара, є фундаментальною метрикою для оцінки просторового перекриття між двома областями – передбаченим обмежувальним прямокутником/маскою та відповідним об'єктом істинних даних (ground truth).

- **Визначення:** IoU обчислюється як площа перетину двох областей, поділена на площу їх об'єднання.
- **Формула:**

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of } A \cup B} = \frac{A \cap B}{A \cup B} \quad (2)$$

Де:

- A – площа передбаченої області (bounding box або маски).
- B – площа істинної області (ground truth bounding box або маски).
- $A \cap B$ – площа перетину областей A та B.
- $A \cup B$ – площа об'єднання областей A та B.
- **Інтерпретація:** Значення IoU варіюється від 0 (немає перекриття) до 1 (ідеальне перекриття). Чим вище значення IoU, тим краще передбачена область збігається з істинною. IoU використовується як поріг для визначення того, чи є передбачення "істинно позитивним" (True Positive, TP). Наприклад, якщо $IoU > 0.5$, передбачення вважається TP.

Precision, Recall та F1-score (для класифікації в контексті детекції/сегментації)

Ці метрики використовуються для оцінки ефективності класифікації всередині задачі детекції/сегментації, базуючись на визначеннях True Positive (TP), False Positive (FP) та False Negative (FN).

1. **True Positive (TP):** Кількість об'єктів, які були правильно виявлені/сегментовані, їхній клас був правильно класифікований, і їхній IoU з істинними даними перевищує заданий поріг.
 2. **False Positive (FP):** Кількість передбачень, які:
 - Не відповідають жодному істинному об'єкту (IoU < порогу).
 - Відповідають істинному об'єкту, але були класифіковані неправильно.
 - Є дублікатами вже виявленого TP.
 3. **False Negative (FN):** Кількість істинних об'єктів, які не були виявлені або були пропущені моделлю.
- **Precision (Точність):** Вимірює частку правильних передбачень серед усіх передбачень, зроблених моделлю. Висока точність означає низьку кількість хибнопозитивних спрацьовувань.

Математично Precision обчислюється за формулою:

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Де:

- TP - кількість істинно позитивних спрацьовувань;
- FP - кількість хибнопозитивних спрацьовувань;
- **Recall (Повнота/Чутливість):** Вимірює частку правильно виявлених/сегментованих істинних об'єктів серед усіх істинних об'єктів. Висока повнота означає низьку кількість хибнонегативних спрацьовувань.

Математично Recall обчислюється за формулою:

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Де:

- TP - кількість істинно позитивних спрацювань;
- FN - кількість хибнонегативних спрацювань
- **F1-score**: Є гармонійним середнім між Precision та Recall, надаючи збалансовану метрику, яка враховує обидва показники. Це особливо корисно, коли є дисбаланс між класами або коли важливі як точність, так і повнота.

Математично F1-score обчислюється за формулою:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

Де:

- Precision - частка правильно класифікованих об'єктів серед усіх, які модель віднесла до цього класу;
- Recall - частка знайдених моделлю об'єктів серед усіх істинно наявних об'єктів даного класу;

Average Precision (AP) та Mean Average Precision (mAP)

Для узагальненої оцінки продуктивності моделі в задачах детекції та сегментації, що включають багато класів і різні пороги IoU, використовуються метрики AP та mAP.

1. **Precision-Recall Curve (PR-крива)**: Для кожного класу може бути побудована PR-крива, яка відображає залежність Precision від Recall при зміні порогу впевненості передбачень моделі.

Математично Precision-Recall Curve обчислюється за формулою:

$$AP = \int_0^1 p(r) dr \quad (6)$$

Де:

- $P(r)$ - функція точності (precision) відносно рівня повноти (recall);
- R - рівень повноти (recall), що змінюється від 0 до 1.

2. **Average Precision (AP):** Це площа під PR-кривою для одного конкретного класу при заданому порозі IoU. AP є інтегральною оцінкою продуктивності моделі для певного класу.

- **Формула (інтерпольована, як у COCO):** AP зазвичай розраховується шляхом інтерполяції Precision по 11 (або більше) точках Recall (0, 0.1, ..., 1.0) для згладжування PR-кривої.

$$AP = \sum_{k=1}^N P(k) \Delta R(k) \quad (7)$$

Де:

- $P(k)$ - інтерпольована точність на k -му порозі Recall
- $\Delta R(k)$ – зміна повноти між k та $k-1$ порогом

3. **Mean Average Precision (mAP):** Є середнім значенням AP для всіх класів об'єктів, що розпізнаються моделлю. Це загальна метрика, яка узагальнює продуктивність моделі по всьому набору класів.

В контексті COCO-метрик, mAP може бути розрахований по-різному:

mAP@0.5: Це середнє значення AP для всіх класів, коли поріг IoU встановлений на 0.5. Ця метрика фокусується на тому, наскільки добре модель виявляє об'єкти без надмірно суворих вимог до їх точної локалізації.

$$mAP@0.5 = \frac{1}{C} \sum_{i=1}^C AP_{i, IoU=0.5} \quad (8)$$

Де:

- C - кількість класів
- $AP_{i, IoU=0.5}$ – Average Precision для i -го класу при $IoU = 0.5$

mAP@0.5:0.95: Це найбільш комплексна метрика, яка усереднює AP для всіх класів та для різних порогів IoU, починаючи від 0.5 до 0.95 з кроком 0.05. Це означає, що модель оцінюється за її здатністю точно виявляти об'єкти при різних рівнях строгості локалізації. Цей показник є більш репрезентативним для загальної якості моделі.

$$mAP@0.5:0.95 = \frac{1}{10 * C} \sum_{k=0.5}^{0.95(\text{step } 0.05)} \sum_{i=1}^C AP_{i, IoU=k} \quad (9)$$

Де:

- 10 – кількість порогів IoU від 0.5 до 0.95 (0.5, 0.55, ..., 0.95).

У даному дослідженні для оцінки якості детекції та сегментації тіл корів використовувалися саме СОСО-метрики (mAP@0.5 та mAP@0.5:0.95) для bounding box та масок, а також загальна точність класифікації та матриця плутанини для категорій BCS, що забезпечує всебічну та об'єктивну оцінку продуктивності розробленої системи.

Висновки до другого розділу

У другому розділі детально розглянуто постановку задачі автоматизованого визначення вгодованості ВРХ (BCS), що включає детекцію, сегментацію та класифікацію на основі морфологічних ознак.

Визначено вимоги до системи (неінвазивність, об'єктивність, адаптивність, економічність, безпека, простота) та критерії ефективності (висока точність детекції/сегментації за mAP, точність класифікації BCS за P, R, F1, Confusion Matrix, а також швидкість обробки та зручність використання).

Критично важливу роль відіграли методи збору та підготовки даних. Для датасету з 53 566 зображень ВРХ було застосовано інноваційний pipeline ітеративної псевдо-розмітки, що поєднав YOLOv11x для детекції та SAM 2.1 Large для генерації високоякісних полігональних масок. Це дозволило ефективно створити 49 097 детальних анотацій для instance segmentation, долаючи обмеження

ручної розмітки. Додатково, використовувалися різноманітні техніки аугментації для підвищення узагальнюючої здатності моделі.

Обрані метрики оцінювання якості моделей (IoU, P, R, F1, AP, mAP@0.5, mAP@0.5:0.95) забезпечили всебічну та об'єктивну оцінку продуктивності системи.

РОЗДІЛ 3. Розробка методу та реалізація програмного комплексу

3.1 Архітектура системи (клієнт-серверна модель)

Для забезпечення ефективного та масштабованого функціонування системи автоматизованого визначення вгодваності ВРХ була обрана клієнт-серверна архітектура. Такий підхід дозволяє оптимально розподілити обчислювальне навантаження, централізувати зберігання даних та забезпечити гнучкість у розширенні функціоналу. Загальна діаграма архітектури системи (див. рис. А.7, дод. А), а діаграма потоку даних (див. рис. А.8, дод. А). Система складається з трьох ключових взаємодіючих компонентів:

1. Мобільний застосунок (Клієнтська частина):

- **Призначення:** Є основним інтерфейсом для користувача (фермера), розробленим на Kotlin для Android. Він забезпечує інтуїтивно зрозумілу взаємодію з системою.
- **Основний функціонал:** Дозволяє користувачеві робити фотографії ВРХ безпосередньо з мобільного пристрою або завантажувати існуючі зображення з галереї. Мобільний застосунок відображає отримані від сервера результати оцінки вгодваності (BCS) з візуалізацією обмежувальних рамок (bounding boxes) та сегментаційних масок. Також надає доступ до агрегованих даних та звітів, дозволяє керувати профілем та історією вимірювань.
- **Взаємодія:** Обмін даними з обома серверами (Inference Server та Backend Server) відбувається через RESTful API (HTTP/HTTPS).

2. Inference Server (Сервер обробки нейронною мережею):

- **Призначення:** Виконання безпосереднього інференсу моделі глибокого навчання (Mask R-CNN з Swin Transformer) для визначення вгодованості ВРХ. Це обчислювальне ядро системи, що працює на Python з FastAPI та Uvicorn.
- **Вхід/Вихід:** Отримує зображення від мобільного застосунку. Повертає результати обробки, які включають: класифікацію ВСС, точність прогнозу, координати обмежувальних рамок для виявлених тварин та дані сегментаційних масок.
- **Оптимізації:** Для максимальної продуктивності впроваджено лінійне завантаження моделі (одноразове завантаження в пам'ять GPU при першому запиті), асинхронну обробку з пулами потоків (для паралельного виконання GPU-інференсу та CPU-інтенсивної постобробки), оптимізоване кодування сегментаційних масок у Base64-PNG та ретельну конфігурацію Uvicorn для високої пропускну здатності. Розгортається на локальному комп'ютері або виділеному сервері з потужним GPU.

3. Backend Server (Серверна частина та база даних):

- **Призначення:** Управління всією бізнес-логікою, авторизацією користувачів, зберіганням та обробкою даних. Реалізований на Node.js з Express.js та PostgreSQL як СУБД.
- **База даних:** Зберігає структуровані дані, такі як дані про тварин, історія ВСС, інформація про користувачів, метадані зображень та результати сегментації.
- **Функціонал:** Управляє реєстрацією, входом (включаючи Google-аутентифікацію) та відновленням пароля. Зберігає всі отримані результати оцінки вгодованості з позначками часу, надає дані для звітів та статистики користувача. Розгортається на локальному ПК або сервері, забезпечуючи надійне зберігання даних.

3.2. Розробка моделі глибокого навчання

Для вирішення багатозадачної проблеми, що включає детекцію, сегментацію екземплярів та класифікацію вгодованості ВРХ, обрано архітектуру Mask R-CNN. Її здатність генерувати високоточні полігональні маски є критично важливою для детального морфологічного аналізу тіла тварини.

В рамках дослідження порівнювались дві конфігурації Mask R-CNN з backbone-архітектурами Swin Transformer: Swin Transformer Tiny (Swin-T) з 47427474 параметрами (depths=[2, 2, 6, 2]) та Swin Transformer Small (Swin-S) з 68778045 параметрами (depths=[2, 2, 18, 2]). Обидві моделі використовують FPN та налаштовані на 5 класів BCS, застосовуючи CrossEntropyLoss та SmoothL1Loss. Ієрархічна структура та механізм зсунутих вікон (Shifted Window Attention) Swin Transformer є ключовими для виявлення тонких морфологічних змін.

Навчання моделей здійснювалося за допомогою фреймворку MMDetection. Використовувалося трансферне навчання (з попередньо навченими вагами з датасету COCO), max_epochs=50, оптимізатор AdamW з lr=0.0001 та комбінований планувальник швидкості навчання (LinearLR та CosineAnnealingLR). Для Swin-T використовувався batch_size=4 (ефективний 16) та для Swin-S — batch_size=2 (ефективний 12) з градієнтною акумуляцією. Ввімкнення Автоматичного змішаного прецизійного обчислення (AMP) (fp16=dict(loss_scale='dynamic')) дозволило прискорити навчання та зменшити споживання пам'яті GPU. Хук EMAHook застосовував експоненційне рухоме середнє до ваг, підвищуючи стабільність та узагальнюючу здатність. Навчання проводилося на системі з GPU NVIDIA RTX 5060, займаючи від 50 до 72 годин на модель.

Після завершення навчання обидві моделі оцінювались на відкладеній тестовій вибірці з використанням різних аугментацій для оцінки робастності.

- **Загальна точність класифікації** (див. таб. А.1, дод. А) показала перевагу Swin-Small (84.64%) над Swin-Tiny (82.20%).

- **Покласова точність** (див. таб. А.2, дод. А) також демонструє кращі результати Swin-Small у більшості категорій BCS, особливо для середніх класів.
- **Точність для різних аугментацій** (див. таб. А.3, дод. А) підтвердила вищу робастність Swin-Small, особливо для аугментацій "noise" (60.0% проти 48.7% у Swin-Tiny) та "scale_down" (81.3% проти 76.7% у Swin-Tiny).
- **Матриці плутанини** (див. рис. А.9, дод. А) візуально показують, що Swin-Small має менше помилок у розрізненні між близькими категоріями BCS.

На основі цих результатів Mask R-CNN з backbone Swin Transformer Small (Swin-S) була обрана як основна для інтеграції в систему, завдяки її вищій загальній точності та робастності.

3.3 Серверна частина та база даних

Серверна частина програмного комплексу складається з двох окремих сервісів: Backend Server (для управління користувачами та даними) та Inference Server (для обробки зображень моделями глибокого навчання). Такий розподіл оптимізує швидкість обчислень та надійність роботи з даними.

Backend Server розроблений на Node.js (Express.js) та використовує PostgreSQL як систему управління базами даних. Для аутентифікації застосовуються bcryptjs та jsonwebtoken (JWT), а також інтеграція з Google Sign-in. Цей сервер є центральним сховищем даних та реалізує більшість бізнес-логіки. Його основні функції: управління користувачами (реєстрація, вхід, відновлення пароля, Google-аутентифікація), зберігання та управління історією прогнозів вгодваності (BCS) для кожного користувача, надання даних для звітів та статистики. База даних містить дві основні таблиці: users (інформація про користувачів) та predictions (історія прогнозів BCS). Ендпоінти Backend Server (див. таб. А.4, дод. А) Детальний опис структури цих таблиць(див. таб. А.5, А.6, дод. А)

Inference Server розроблений на Python з використанням асинхронного веб-фреймворку FastAPI та веб-сервера Uvicorn (з uvloop). Для роботи з моделлю глибокого навчання застосовуються бібліотеки PyTorch та MMDetection. Цей сервер є "обчислювальним ядром" системи. Його завдання – отримання зображень, виконання інференсу моделі Mask R-CNN зі Swin Transformer та повернення результатів детекції, сегментації та класифікації BCS. Для досягнення максимальної продуктивності та низької затримки Inference Server було впроваджено низку глибоких оптимізацій:

- **Лінійне завантаження моделі:** Модель (Mask R-CNN зі Swin-S) завантажується в пам'ять GPU лише один раз при першому запиті, що значно прискорює обробку послідовних запитів.
- **Асинхронна обробка та пули потоків:** Операції, що блокують потік (інференс, завантаження зображень, постобробка), делегуються в окремі пули потоків (ThreadPoolExecutor), що дозволяє основному асинхронному циклу залишатися чутливим до інших запитів.
- **Оптимізована постобробка результатів:** Після отримання передбачень, результати швидко фільтруються за порогом впевненості, а сегментаційні маски ефективно кодуються у формат Base64-PNG для компактної передачі в JSON-відповіді.
- **Конфігурація Uvicorn:** Веб-сервер налаштований з uvloop, одним робочим процесом (--workers 1), динамічним лімітом одночасних запитів (--limit-concurrency) та іншими параметрами для максимальної пропускної здатності та стабільності.

Ендпоінти Inference Server(див. таб. А.7, дод. А).

3.4. Мобільний застосунок

Мобільний застосунок, розроблений для платформи Android з використанням Kotlin та Android Studio, є ключовим інтерфейсом взаємодії для кінцевого користувача (фермера). Його мінімалістичний та інтуїтивно зрозумілий

дизайн (візуальні приклади (див. рис. Б.1-10, дод. Б)) забезпечує зручний доступ до функціоналу автоматизованого визначення вгодованості ВРХ.

Ключові функціональні модулі:

- **Модуль аутентифікації та управління профілем:** Забезпечує реєстрацію, вхід (включаючи Google Sign-in), скидання пароля, перегляд персональної статистики (кількість класифікацій, середня точність, популярні класи BCS) та функції управління обліковим записом (видалення даних/акаунту).
- **Основний екран ("Камера" / "Нове визначення"):** Надає доступ до камери пристрою з візуальними підказками для оптимальної зйомки ВРХ, а також можливість вибору зображень з галереї. Зображення автоматично обрізаються та масштабуються.
- **Екран результатів класифікації:** Відображає результати, отримані від сервера: на зображення накладаються обмежувальні рамки та сегментаційні маски (декодовані з Base64-PNG), чітко вказується клас BCS, впевненість моделі та надаються короткі поради. Дозволяє зберегти результати до історії, експортувати фото/JSON-дані та поділитися ними.
- **Екрани історії та деталей прогнозу:** Забезпечують перегляд хронологічного списку всіх класифікацій, їх детальний аналіз (з візуалізацією, BCS, порадами) та можливості видалення окремих записів або всієї історії.
- **Екран налаштувань:** Дозволяє користувачеві контролювати відображення результатів (поріг впевненості, видимість обмежувальних рамок та масок).

Взаємодія з сервісами та API: Застосунок активно взаємодіє з Backend Server (Node.js) та Inference Server (Python) через RESTful API, використовуючи бібліотеки Retrofit та OkHttpClient. Він надсилає зображення для аналізу та отримує JSON-відповіді з результатами.

Тестування та оцінка ефективності: Проводилося функціональне тестування всіх модулів, перевірка взаємодії з серверами та оцінка UI/UX, підтверджуючи інтуїтивність, стабільність та зручність застосунку.

Таким чином, мобільний застосунок є ефективним та зручним інструментом, що інтегрує можливості глибокого навчання з повсякденною діяльністю фермера для об'єктивного моніторингу вгодованості ВРХ.

Висновки до третього розділу

У третьому розділі описано розробку та реалізацію програмного комплексу для автоматизованого визначення вгодованості ВРХ.

Архітектура системи побудована на гнучкій клієнт-серверній моделі. Вона включає мобільний застосунок, спеціалізований Inference Server для обробки нейронною мережею та Backend Server з базою даних для управління даними.

Модель глибокого навчання Mask R-CNN з backbone Swin Transformer Small (Swin-S) була обрана після порівняльного аналізу з Swin-Tiny. Swin-S продемонстрував вищу загальну точність класифікації BCS (84.64%), кращу покласову точність та робастність до аугментацій. Процес навчання включав трансферне навчання та комплекс оптимізацій для ефективного інференсу.

Серверна частина та база даних реалізовані:

- **Backend Server** (Node.js/Express.js/PostgreSQL) керує користувачами (аутентифікація, Google Sign-in) та безпечно зберігає історію прогнозів BCS.
- **Inference Server** (Python/FastAPI/Uvicorn) оптимізований для високопродуктивного інференсу моделі, використовуючи лініве завантаження, асинхронну обробку та ретельну конфігурацію Uvicorn.

Мобільний застосунок (Kotlin/Android) надає зручний інтерфейс для аутентифікації, захоплення зображень ВРХ, відображення результатів BCS (з обмежувальними рамками, масками та порадами), перегляду історії та статистики. Зображення інтерфейсу представлені у Додатку Б.

Отже, розроблений програмний комплекс є цілісним та ефективним рішенням, що поєднує можливості глибокого навчання з функціональною реалізацією для визначення вгодованості ВРХ.

Розділ 4. Експериментальні дослідження та тестування

4.1 Постановка експериментів та тестові датасети

Для всебічної оцінки розробленої системи автоматизованого визначення вгодованості великої рогатої худоби було проведено два основні блоки експериментів, спрямовані на аналіз процесу навчання моделей та оцінку їхньої кінцевої продуктивності в умовах, наближених до реальних.

1. Оцінка динаміки навчання та продуктивності на валідаційному наборі:
 - **Мета:** Проаналізувати, як моделі Mask R-CNN зі Swin Transformer Tiny (Swin-T) та Mask R-CNN зі Swin Transformer Small (Swin-S) навчаються та узагальнюють на раніше не бачених даних під час тренувального циклу. Цей етап дозволяє виявити тенденції у втратах та метриках, а також визначити оптимальні чекпоінти моделей.
 - **Валідаційний датасет:** Використовувалася валідаційна вибірка (annotations/val.json), що складає приблизно 10% від загального датасету. Цей набір використовується для періодичної оцінки моделі в ході навчання, не впливаючи на оптимізацію ваг моделі.
 - **Метрики:** Моніторилися графіки втрат (bbox_loss, seg_loss, cls_loss, dfl_loss) на тренувальному та валідаційному наборах. Для оцінки якості детекції та сегментації використовувалися COCO-метрики Average Precision (AP) та Mean Average Precision (mAP): mAP@0.5, mAP@0.75 та mAP@0.5:0.95 для bounding box (bbox) та масок (segm). Для класифікації BCS розраховувалися Precision (P), Recall (R) та F1-score для кожного класу.

- **Умови:** Оцінювання відбувалося автоматично в кінці кожної епохи навчання за допомогою MMEngine.

Експерименти з навчання, валідації та тестування моделей проводилися в наступному апаратно-програмному середовищі:

- **Операційна система:** Linux (sys.platform: linux).
- **Процесор (CPU):** Intel Core i5 14400f.
- **Оперативна пам'ять (RAM):** Ddr5 32gb 5600mhz.
- **Графічний процесор (GPU):** NVIDIA GeForce RTX 5060.
- **Програмне забезпечення:**
 - **Python:** 3.12.12 (packaged by Anaconda, Inc.).
 - **PyTorch:** 2.9.0+cu130. .
 - **OpenCV:** 4.12.0.
 - **MMEngine:** 0.10.7.
 - **MMDetection:** 3.3.0.
 - **Компілятор GCC:** gcc (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0.

Для оцінки продуктивності та стабільності Inference Server був розроблений спеціалізований Python-скрипт MaskRCNNStressTest. Цей інструмент дозволяє імітувати поведінку численних одночасних користувачів, надсилаючи запити до API сервера в різних режимах навантаження, що дозволяє комплексно оцінити його пропускну здатність, час відгуку та стійкість до відмов.

4.2. Аналіз результатів роботи моделей

Якість навчального датасету є фундаментальною для продуктивності моделі. Для створення полігональних масок було застосовано інноваційний pipeline ітеративної псевдо-розмітки, що поєднує можливості YOLOv11x для детекції обмежувальних рамок та SAM 2.1 Large для генерації піксельних масок. Цей процес дозволив отримати високоякісні сегментаційні анотації для 49 097 зображень ВРХ, приклад сегментації(див. рис. А.10, дод. А).

Графіки(див. рис. A.11, A.12 дод. А) відображаються динаміку втрат на моделі Swin-S та Swin-T.

Відповідно до (див. рис. A.11, A.12 дод. А) видно,що всі компоненти втрат (для bounding box, сегментації, класифікації та DFL) демонструють стабільне зниження протягом процесу навчання. Це вказує на успішну збіжність моделі та її здатність до мінімізації помилок. Коливання на валідаційних кривих є типовими, але загальна тенденція до зниження свідчить про ефективне узагальнення.

Метрики mAP для bound box та масок, які були отримані на найкращій епозі під час навчання (див. таб. А.8, дод. А).

Значення Precision (P), Recall (R) та F1-score для класифікації кожної категорії BCS, отримані в ході валідації моделей(див. таб. А.9, дод. А).

Обидві моделі демонструють дуже високі F1-score для класифікації BCS у всіх категоріях на валідаційному наборі. Swin-Small показує незначне, але послідовне покращення F1-score для більшості класів порівняно зі Swin-Tiny.

Продемонстровано роботу моделей(див. рис. А.13, дод. А), та візуалізацію сегментації та класифікації.

На основі детального аналізу динаміки навчання, високих показників COCO-метрик на валідаційному наборі, модель Mask R-CNN з backbone Swin Transformer Small (Swin-S) була визнана оптимальною. Вона демонструє вищу загальну точність класифікації BCS (84.64%), кращі показники для окремих класів та значно більшу робастність до різноманітних умов зйомки та візуальних перешкод на тестовій вибірці. Її висока точність детекції та піксельної сегментації, підтверджена метриками AP та F1-score, робить її надійним інструментом для автоматизованої оцінки вгодваності ВРХ.

4.3. Стрес-тестування серверної частини

Стрес-тестування Inference Server є вирішальним етапом для оцінки його продуктивності, стабільності та здатності витримувати різні рівні навантаження в умовах реальної експлуатації. Це дозволяє визначити максимальну пропускну

здатність системи та її поведінку при наближенні до критичних навантажень. Для цього був розроблений спеціалізований Python-скрипт (MaskRCNNStressTest), що імітує поведінку численних одночасних користувачів.

1. **Інструмент:** Розроблений скрипт MaskRCNNStressTest дозволяє конфігурувати тести з різними режимами навантаження. Для оцінки меж продуктивності використовувався режим find-limit, який поступово збільшує кількість одночасних "користувачів" протягом заданої тривалості тесту.
2. **Сервер, що тестується:** Inference Server, розгорнутий на Python/FastAPI/Uvicorn (з моделлю Mask R-CNN Swin-Small) на системі з GPU NVIDIA GeForce RTX 5060. Детальний опис середовища надано в Розділі 4.1.
3. **Параметри тестування:** Було проведено тест у режимі find-limit з наступними параметрами:
 - duration: 300 секунд
 - min_users: 20 (початкова кількість одночасних користувачів)
 - max_users: 350 (максимальна кількість користувачів, до якої наростає навантаження)
4. **Механізм навантаження:** Скрипт лінійно збільшував кількість активних потоків (target_users) від 20 до 350 протягом 300 секунд. Кожен потік імітував користувача, який постійно надсилає запити до API /predict, використовуючи випадкові зображення з тестової папки.
5. **Метрики моніторингу:** Під час тестування відстежувалися та виводилися в консоль кожні 5 секунд такі показники:
 - Кількість активних "користувачів".
 - Загальна кількість оброблених запитів.
 - Відсоток успішних запитів.
 - Пропускна здатність (запитів на секунду, req/s).
 - Середній час відгуку (avg_time) останніх запитів.
 - Кількість помилок HTTP 503.

Результати стрес тесту з різний проміжок часу(див. таб. А.10, дод. А).

Протягом усієї тривалості тесту (300 секунд), при наростаючому навантаженні від 20 до 344 одночасних користувачів, сервер зберігав 100% успішність обробки запитів без єдиної помилки HTTP 503 Service Unavailable. Пропускна здатність стабілізувалася на рівні близько 29.7 запитів на секунду. Середній час відгуку поступово зростає зі збільшенням кількості користувачів (від 0.78с на початку до 11.06с до кінця тестування), що є очікуваним при вичерпанні ресурсів.

Стрес-тестування в режимі find-limit успішно продемонструвало високу продуктивність, стабільність та відмовостійкість Inference Server. Система, розгорнута на одному GPU NVIDIA RTX 5060, здатна безперебійно обробляти майже 30 запитів на секунду, підтримуючи 100% успішність навіть при навантаженні до 344 одночасних користувачів. Це підтверджує, що реалізовані глибокі оптимізації дозволили досягти виняткової ефективності та дозволяють системі витримувати значні навантаження.

Висновки до четвертого розділу

У четвертому розділі були представлені та проаналізовані результати комплексних експериментальних досліджень, що охоплюють як якість розроблених моделей глибокого навчання, так і продуктивність серверної частини системи.

Проведено комплексні експериментальні дослідження розробленої системи автоматизованого визначення вгодваності ВРХ.

Аналіз динаміки навчання показав стабільну збіжність моделі Mask R-CNN зі Swin Transformer Small (Swin-S) та високі показники Average Precision (AP) на валідаційному наборі. Тестування на окремому наборі з аугментаціями підтвердило, що Swin-S є оптимальною моделлю, досягнувши вищої загальної точності класифікації BCS (84.64%) та робастності до візуальних спотворень.

Стрес-тестування серверної частини показало, що Inference Server на одному GPU NVIDIA RTX 5060 успішно обробив 100% запитів при навантаженні до 344 одночасних користувачів, досягнувши пропускну здатності близько 29.73 req/s. Це підтверджує ефективність впроваджених оптимізацій.

Результати експериментів підтвердили ефективність та надійність розробленої системи для практичного застосування в аграрних підприємствах.

Висновки

У представлений магістерській роботі успішно вирішено актуальну науково-прикладну задачу автоматизованого визначення вгодованості великої рогатої худоби (ВРХ) на основі технологій глибокого навчання. Це дослідження є важливим кроком у цифровій трансформації аграрного сектору України, спрямованим на підвищення ефективності тваринництва та забезпечення його сталого розвитку в умовах поствоєнного відновлення.

В основі роботи лежало розуміння обмежень традиційних, суб'єктивних та трудомістких методів оцінки вгодованості (BCS). Це спонукало до розробки об'єктивного та автоматизованого рішення за допомогою передових архітектур глибокого навчання та комп'ютерного зору. Методологічний підхід охоплював детекцію, сегментацію та класифікацію ВРХ.

Значним досягненням стала розробка та успішне застосування інноваційного pipeline ітеративної псевдо-розмітки. Цей підхід, поєднуючи можливості моделей YOLOv11x та SAM 2.1 Large, дозволив автоматично створити високоякісний сегментаційний датасет з полігональними масками для 49 097 зображень ВРХ, ефективно долаючи проблему значних витрат на ручну анотацію.

Центральним елементом програмного комплексу стала розроблена та оптимізована модель глибокого навчання Mask R-CNN з backbone Swin Transformer Small (Swin-S). Порівняльний аналіз показав, що Swin-S демонструє вищу загальну точність класифікації BCS (84.64%) та кращу робастність до різноманітних

візуальних трансформацій порівняно зі Swin Transformer Tiny. Для забезпечення високої продуктивності Inference Server було впроваджено комплекс оптимізацій, включаючи лінійне завантаження моделі, асинхронну обробку запитів з пулами потоків та ефективну конфігурацію веб-сервера.

Програмний комплекс реалізовано за клієнт-серверною архітектурою, яка включає:

- **Мобільний застосунок** (Kotlin/Android) для взаємодії з користувачем, що надає інтерфейс для захоплення зображень, візуалізації результатів BCS (з обмежувальними рамками та сегментаційними масками) та управління даними.
- **Inference Server** (Python/FastAPI/Uvicorn) для обробки зображень та виконання передбачень моделі.
- **Backend Server** (Node.js/Express.js з PostgreSQL) для управління користувачами та даними прогнозів.

Проведені комплексні експериментальні дослідження та тестування повністю підтвердили ефективність та надійність розробленої системи:

- Аналіз динаміки навчання та валідації моделі Swin-S продемонстрував стабільне зниження втрат та досягнення виняткових значень СОСО-метрики Average Precision (AP) на валідаційному наборі (понад 0.94 для mAP@0.5:0.95), а також високий F1-score (понад 0.95) для всіх класів BCS.
- Стрес-тестування Inference Server у режимі find-limit показало його виняткову відмовостійкість та продуктивність: сервер на одному GPU NVIDIA RTX 5060 успішно обробив 100% запитів при наростаючому навантаженні до 344 одночасних користувачів протягом 300 секунд, досягнувши стабільної пропускної здатності близько 29.73 запитів на секунду.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Heinrichs J. Precision Feed Heifers. Dairy Herd. 2020. November 20. URL: <https://www.dairyherd.com/news-news/precision-feed-heifers>
2. Bewley J. Body Condition Scoring as a Tool for Dairy Herd Management / J. Bewley, R. Boyce, D. Hockin, L. Munksgaard, S. Eicher, M. Einstein, K. Schutz // Penn State Extension. – URL: <https://extension.psu.edu/body-condition-scoring-as-a-tool-for-dairy-herd-management>
3. Wangchuk K. Comparison and reliability of techniques to estimate live cattle body weight / K. Wangchuk, C. Wangdi, T. Mindu // Journal of Applied Animal Research. – 2017. – Vol. 46, No. 1. – P. 349–352. DOI: 10.1080/09712119.2017.1302876
4. Brandl N. Stress-free cattle weighing systems / N. Brandl, E. Jørgensen // BOVINE project. – URL: <https://hub.bovine-eu.net/training-in-animal-welfare-for-operatorsfarmers-handlers-transporters-and-slaughterhouses-and-stress-free-drive-systems-during-weighing-and-transport-in-beef-cattle-/stress-free-cattle-weighing-systems->
5. Dandıl E. Body Condition Score (BCS) Segmentation and Classification in Dairy Cows using R-CNN Deep Learning Architecture / E. Dandıl, K. K. Çevik // Turkish Journal of Agriculture - Food Science and Technology. – 2019. – Vol. 7, No. 12. – P. 2212–2217.
6. Etsemeskel Tadele, Destaw Worku, Dessie Yigzaw, Teshager Muluneh & Abebe Melese (2025). *Precision of dairy farming: navigating challenges and seizing*

- opportunities for sustainable dairy production in Africa*. *Frontiers in Animal Science*. DOI: 10.3389/fanim.2025.1541838.
7. Bezen, R., Edan, Y., & Halachmi, I. (2020). Computer vision system for measuring individual cow feed intake using RGB-D camera and deep learning algorithms. *Computers and Electronics in Agriculture*, 172, 105345. <https://doi.org/10.1016/j.compag.2020.105345>
 8. Zhao, K., Bewley, J. M., He, D., & Jin, X. (2018). Automatic lameness detection in dairy cattle based on leg swing analysis with an image processing technique. *Computers and Electronics in Agriculture*, 148, 226-236. <https://doi.org/10.1016/j.compag.2018.03.014>
 9. Song, X., Bokkers, E. A., van der Tol, P. P., Koerkamp, P. W., & van Mourik, S. (2018). Automated body weight prediction of dairy cows using 3-dimensional vision. *Journal of Dairy Science*, 101(5), 4448-4459. <https://doi.org/10.3168/jds.2017-13094>
 10. Borchers, M. R., Chang, Y. M., Tsai, I. C., Wadsworth, B. A., & Bewley, J. M. (2016). A validation of technologies monitoring dairy cow feeding, ruminating, and lying behaviors. *Journal of Dairy Science*, 99(9), 7458-7466. <https://doi.org/10.3168/jds.2015-10843>
 11. Andrew, W., Greatwood, C., & Burghardt, T. (2017). Visual identification of individual Holstein-Friesian cattle via deep metric learning. *Computers and Electronics in Agriculture*, 141, 245-257. <https://doi.org/10.1016/j.compag.2017.07.017>
 12. Borchers, M. R., Chang, Y. M., Proudfoot, K. L., Wadsworth, B. A., Stone, A. E., & Bewley, J. M. (2017). Machine-learning-based calving prediction from activity, lying, and ruminating behaviors in dairy cattle. *Journal of Dairy Science*, 100(7), 5664-5674. <https://doi.org/10.3168/jds.2016-11526>
 13. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR)*, 779-788.
<https://doi.org/10.1109/CVPR.2016.91>
14. The evolution of YOLO: Object detection algorithms. URL: YOLO object detection: Evolution and algorithms | SuperAnnotate
 15. Ultralytics YOLO11. URL: <https://docs.ultralytics.com/ru/models/yolo11/>
 16. What is Mask R-CNN? The Ultimate Guide. URL: <https://blog.roboflow.com/mask-rcnn/>
 17. Liu Z., Lin Y., Cao Y., Hu H., Wei Y., Zhang Z., Lin S., Guo B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. P. 10012-10022. DOI: 10.1109/ICCV48922.2021.00986
 18. Semi-Supervised Semantic Segmentation Based on Pseudo-Labels: A Survey. arXiv:2403.01909v1. URL: <https://arxiv.org/html/2403.01909v1>
 19. Huang xiao ping, Dou Zihao, Huang Fei, et al. Dairy cow body condition score target detection data set[DS/OL]. V3. Science Data Bank, 2025[2025-12-18]. <https://cstr.cn/31253.11.sciencedb.16704>. CSTR:31253.11.sciencedb.16704.
 20. Angel T.R. et al. Comparison of manual and automated body condition scoring in dairy cows on a commercial dairy farm. *Veterinary Record*. 2024. DOI: 10.1002/vetr.3891
 21. Bercovich A., Edan Y., Alchanatis V., Moallem U., Parmet Y., Honig H., Maltz E., Antler A., Halachmi I. Development of an automatic cow body condition scoring using body shape signature and Fourier descriptors. *Journal of Dairy Science*. 2013. Vol. 96, No. 12. P. 8047-8059. DOI: 10.3168/jds.2013-6568
 22. Dandıl E., Çevik K.K., Boğa M. Automated Classification System Based on YOLO Architecture for Body Condition Score in Dairy Cows. *Veterinary Sciences*. 2024. Vol. 11, No. 9. P. 399. DOI: 10.3390/vetsci11090399
 23. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly S., Uszkoreit J., Houlsby N. An

- image is worth 16x16 words: Transformers for Image Recognition at Scale. ICLR 2021. URL: <https://arxiv.org/abs/2010.11929>
24. Ferguson J.D., Galligan D.T., Thomsen N. Principal descriptors of body condition score in Holstein cows. *Journal of Dairy Science*. 1994. Vol. 77, No. 9. P. 2695-2703. DOI: 10.3168/jds.S0022-0302(94)77212-X
 25. Jordan T., Bailey E. Body Condition Scoring of Beef Cattle. University of Missouri Extension Publication G2230. URL: <https://extension.missouri.edu/publications/g2230>
 26. Kirillov A., Mintun E., Ravi N., Mao H., Rolland C., Gustafson L., Xiao T., Whitehead S., Berg A.C., Lo W.Y., Dollár P., Girshick R. Segment Anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023. P. 4015-4026. DOI: 10.1109/ICCV51070.2023.00371
 27. Lewis R., Kostermans T., Brovold J.W., Laique T., Ocepek M. Automated Body Condition Scoring in Dairy Cows Using 2D Imaging and Deep Learning. *AgriEngineering*. 2025. Vol. 7, No. 7. P. 241. DOI: 10.3390/agriengineering7070241
 28. Ruan J., Xie M. A review of Segment Anything Model (SAM): Strengths, applications and limitations in image segmentation. *Computer Vision and Image Understanding*. 2024. Vol. 238. P. 103876. DOI: 10.1016/j.cviu.2023.103876
 29. Rodríguez Alvarez J., Arroqui M., Mangudo P., Toloza J., Jatip D., Rodríguez J.M., Downing A. et al. Body condition estimation on cows from depth images using Convolutional Neural Networks. *Computers and Electronics in Agriculture*. 2018. Vol. 155. P. 12-22. DOI: 10.1016/j.compag.2018.09.039
 30. Siachos N. et al. Development and validation of a fully automated 2D surveillance system for BCS monitoring of dairy cows. *Journal of Dairy Science*. 2024. DOI: 10.3168/jds.2024-24567
 31. Song X., Bokkers E.A., van der Tol P.P., Koerkamp P.W.G., van Mourik S. Automated body condition scoring of dairy cows using 3D feature extraction from

- multiple body regions. *Journal of Dairy Science*. 2019. Vol. 102, No. 5. P. 4294-4308. DOI: 10.3168/jds.2018-15238
32. Spoliansky R., Edan Y., Parmet Y., Halachmi I. Development of automatic body condition scoring using a low-cost 3D Kinect camera. *Journal of Dairy Science*. 2016. Vol. 99, No. 9. P. 7714-7725. DOI: 10.3168/jds.2015-10607
33. Wagener M.G. et al. The influence of different examiners on the Body Condition Score (BCS) in dairy cows. *Journal of Dairy Research*. 2023. Vol. 90, No. 2. P. 156-161. DOI: 10.1017/S0022029923000212
34. Wang C.Y., Bochkovskiy A., Liao H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023. P. 7464-7475. DOI: 10.1109/CVPR52729.2023.00721
35. Виклюк Я.І., Камінський Р.М., Пасічник В.В. Моделювання складних систем: посібник. Львів: Видавництво «Новий Світ – 2000», 2020. 404 с.
36. Jocher G., Chaurasia A., Qiu J. YOLO by Ultralytics (Version 11.0.0). 2024. URL: <https://github.com/ultralytics/ultralytics>
37. Lin T.Y., Dollár P., Girshick R., He K., Hariharan B., Belongie S. Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. P. 2117-2125. DOI: 10.1109/CVPR.2017.106
38. Zin T.T., Seint P.T., Tin P., Horii Y., Kobayashi I. Body Condition Score Estimation Based on Regression Analysis Using a 3D Camera. *Sensors*. 2020. Vol. 20, No. 13. P. 3705. DOI: 10.3390/s20133705
39. Python Software Foundation. Python Documentation. URL: <https://docs.python.org/>
40. PyTorch. PyTorch Documentation. URL: <https://pytorch.org/docs/stable/index.html>
41. OpenMMLab. MMDetection Documentation. URL: <https://mmdetection.readthedocs.io/en/latest/>

42. FastAPI. FastAPI Documentation. URL: <https://fastapi.tiangolo.com/>
43. Node.js. Node.js Documentation. URL: <https://nodejs.org/en/docs>
44. PostgreSQL. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/>
45. JetBrains. Kotlin Documentation. URL: <https://kotlinlang.org/docs/home.html>
46. Google Developers. Android Developers Documentation. URL: <https://developer.android.com/docs>

ДОДАТКИ

ДОДАТОК А. Технічні дані та результати експериментів

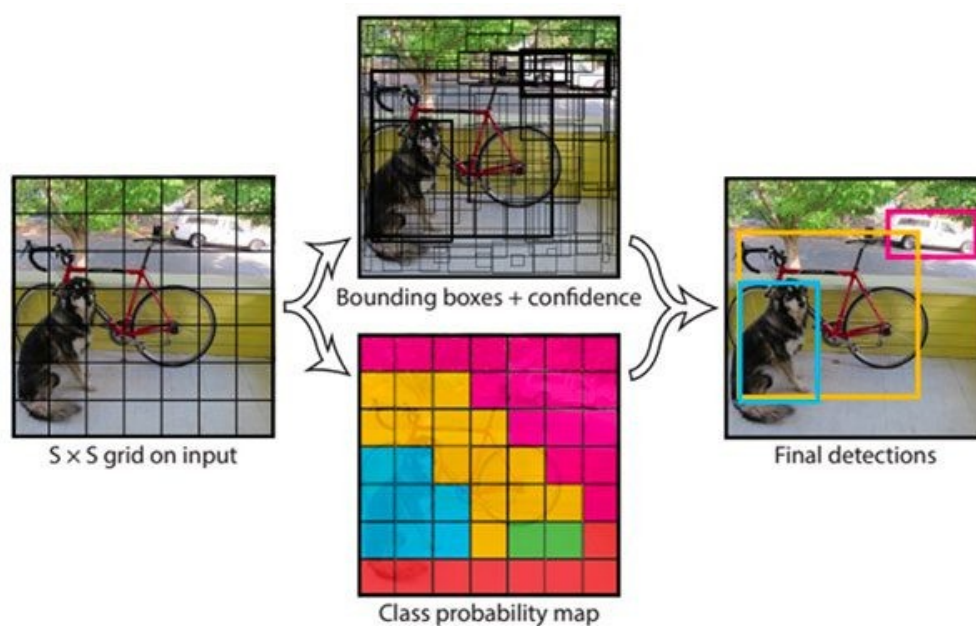


Рисунок А.1 – Принцип роботи Yolo[14]

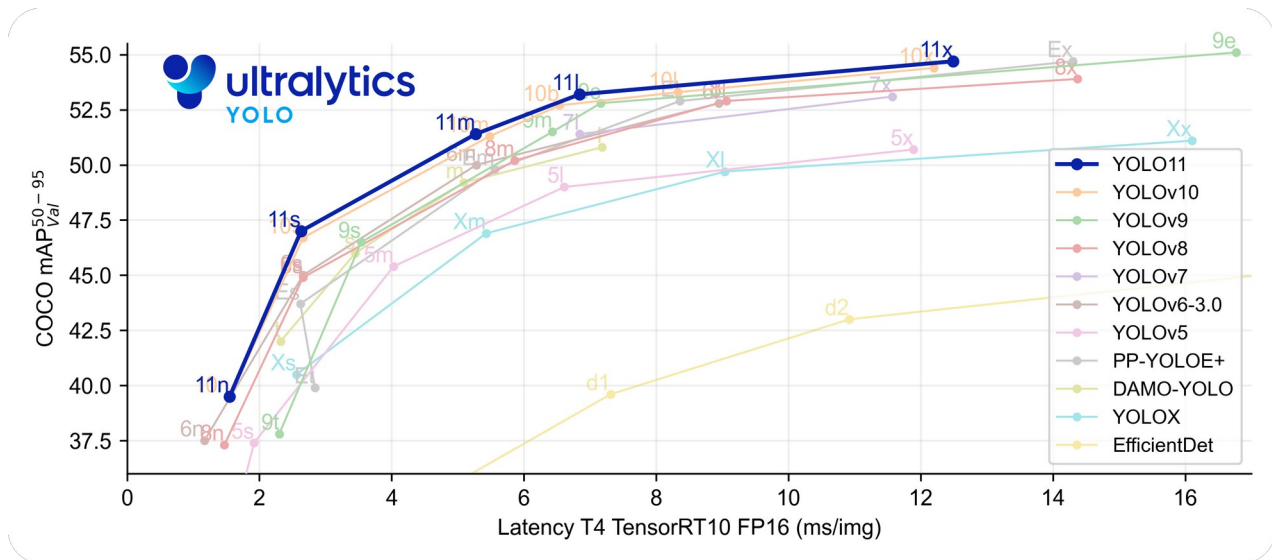


Рисунок А.2 – Порівняння моделей[15]

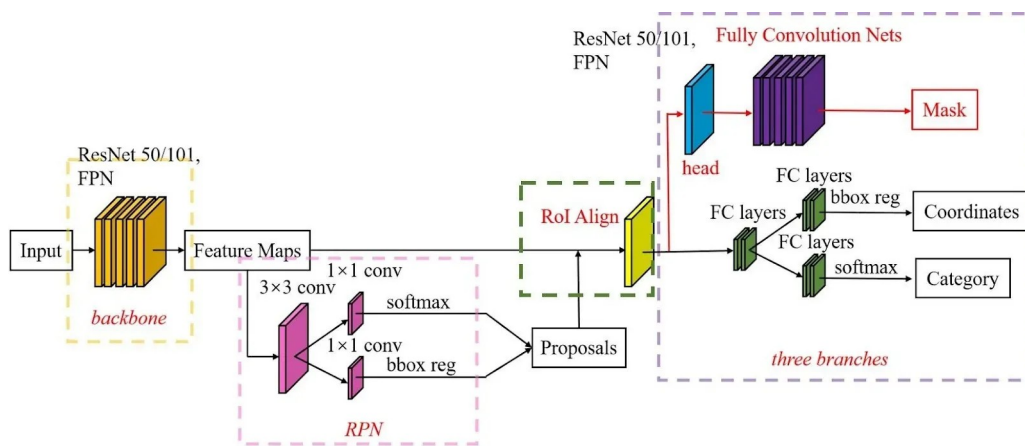


Рисунок А.3 – Принцип роботи Mask R-CNN[16]

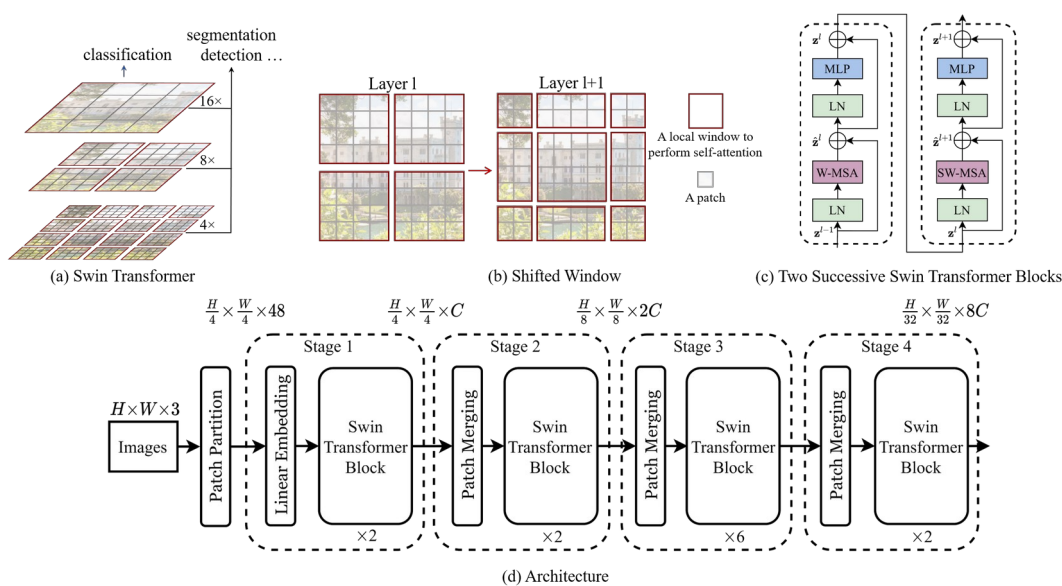


Рис. А.4 - Схема зміщених вікон та архітектура swin transformer[17]

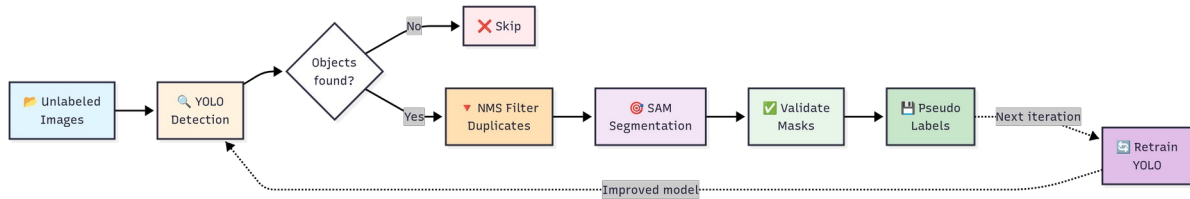


Рисунок А.5 - Процес Iterative pseudo labeling

Колаж аугментацій: Фотометричні спотворення та Геометрія

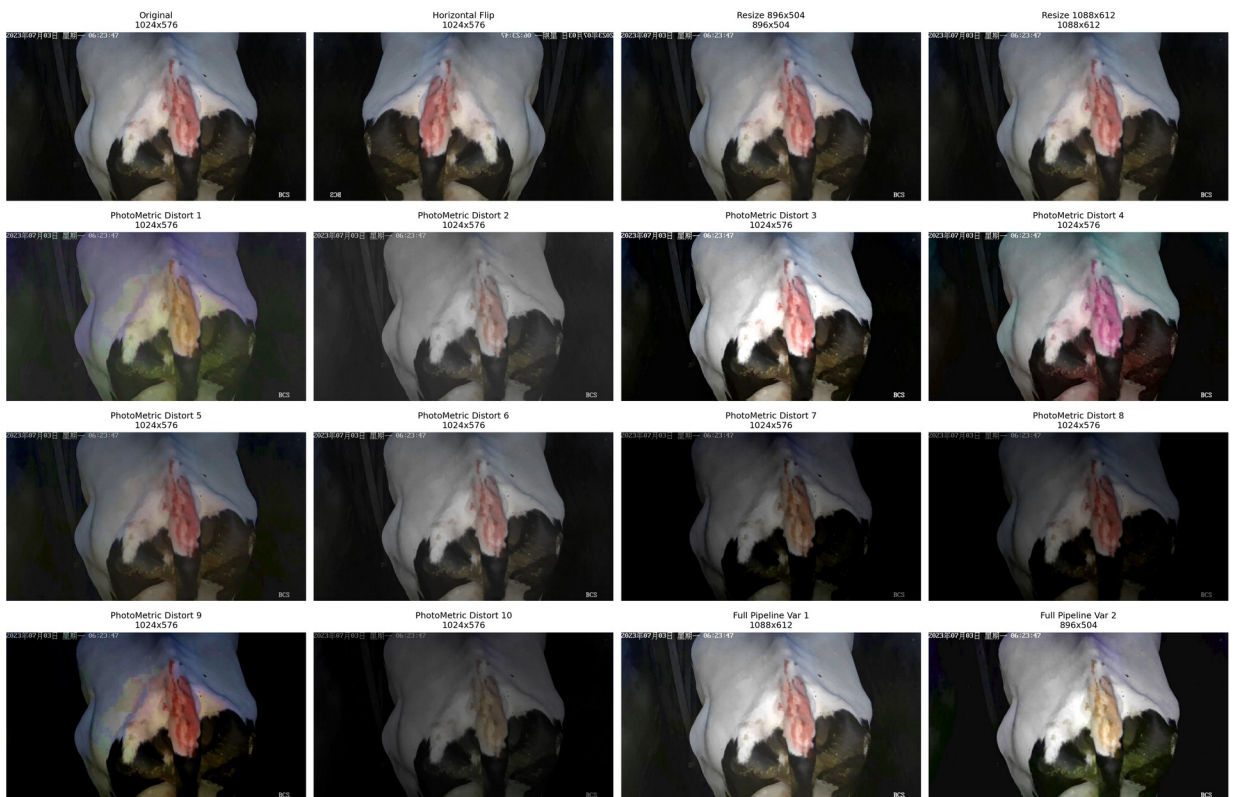


Рисунок А.6 – Демонстрація аугментацій

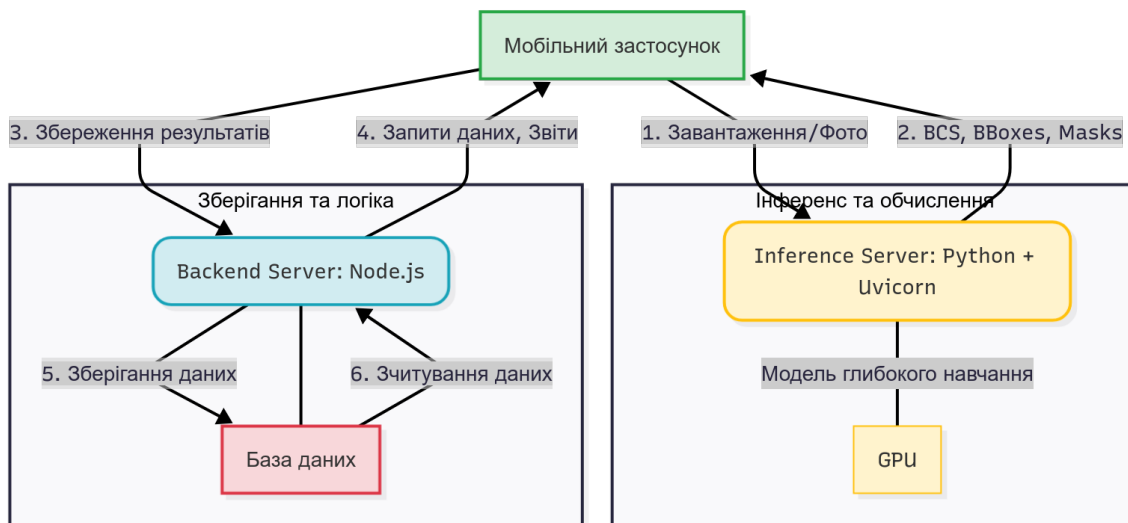


Рисунок А.7 – Архітектура системи

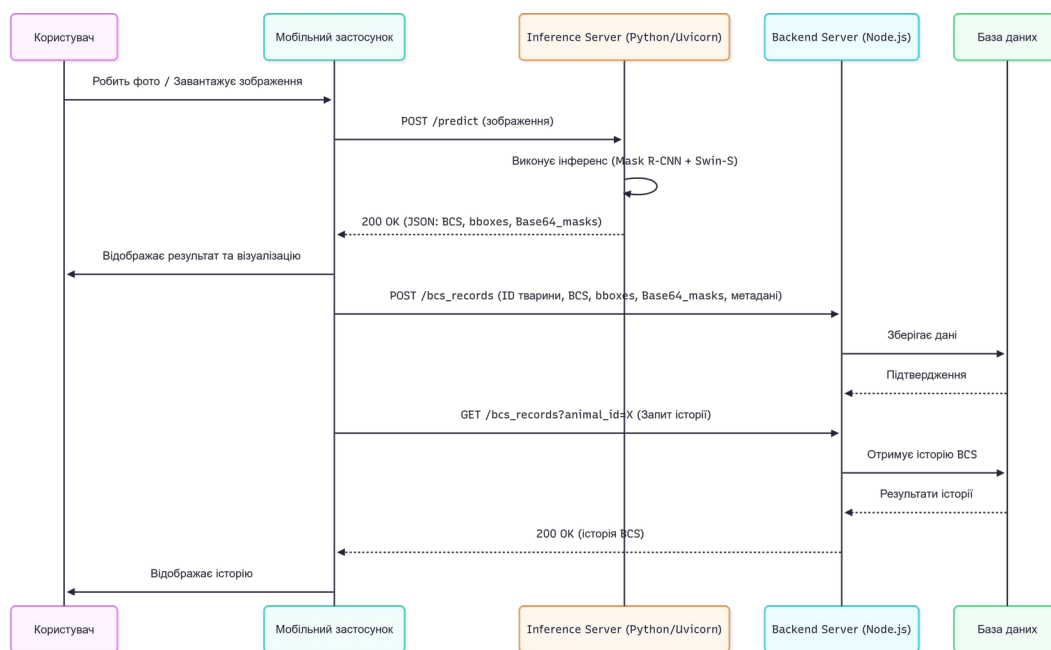


Рисунок А.8 – Потік даних в системі

Таблиця А.1. Загальна точність класифікації BCS

Модель	Кількість правильних	Всього зображень	Точність %
Swin-Tiny	22127	26920	82.20
Swin-Small	22784	26920	84.64

Таблиця А.2. Точність класифікації для кожного класу BCS (%)

Модель	3.25	3.5	3.75	4.0	4.25

Swin-Tiny	79.3	82.3	83.3	80.7	86.0
Swin-Small	83.3	85.4	85.4	84.8	85.7

Таблиця А.3. Точність класифікації для різних аугментацій (%)

Аугментація/Модель	Swin-Tiny	Swin-Small
Original	90.2	90.4
Brightness+	87.9	89.3
Brightness-	89.3	90.5
Contrast+	86.0	85.9
Blur	87.4	88.8
H flip	89.5	90.8
Rotate+10	82.9	84.1
Rotate-10	83.4	85.5
Noise	48.7	60.0
Scale down	76.7	81.3

Таблиця А.4. API Endpoints Backend Server (Node.js)

Endpoint	HTTP Метод	Опис	Захист (JWT)
/api/register	POST	Реєстрація нового користувача.	Ні
/api/login	POST	Вхід користувача, повертає JWT-токен.	Ні
/api/auth/google	POST	Вхід/реєстрація через Google ID токен	Ні
/api/auth/forgot-password	POST	Ініціація скидання пароля, надсилає лист з посиланням.	Ні
/api/auth/reset-password	POST	Зміна пароля за допомогою токена скидання.	Ні
/reset-password	GET	Перенаправлення з посилання для скидання пароля в мобільний застосунок.	Ні
/api/user	GET	Отримання інформації про поточного користувача.	Так
/api/predictions	POST	Збереження результату прогнозу (BCS, bbox, mask, image).	Так
/api/predictions	GET	Отримання історії прогнозів поточного користувача.	Так
/api/predictions/:id	DELETE	Видалення конкретного прогнозу.	Так
/api/user/stats	GET	Отримання статистики прогнозів користувача (загальна кількість, середній confidence, топ-класи).	Так
/api/user/data	DELETE	Видалення всіх прогнозів поточного користувача.	Так
/api/user/account	DELETE	Видалення акаунту користувача та всіх пов'язаних даних.	Так
/api/health	GET	Перевірка стану сервера.	Ні

Таблиця А.5. СУБД Users

Поле	Тип	Опис
id	SERIAL, PRIMARY KEY	Унікальний ідентифікатор користувача
email	VARCHAR, UNIQUE	Електронна пошта для входу
name	VARCHAR	Ім'я користувача
password	VARCHAR	Захешований пароль (bcryptjs)
google_id	VARCHAR, UNIQUE	Google ID (для входу через Google)
reset_token	VARCHAR	Токен для скидання пароля
reset_token_expires	TIMESTAMP	Термін дії токена
created_at	TIMESTAMP	Дата та час реєстрації

Таблиця А.6. СУБД Predicitons

Поле	Тип	Опис
id	SERIAL, PRIMARY KEY	Унікальний ідентифікатор прогнозу
user_id	INTEGER, FOREIGN KEY → users.id	Ідентифікатор користувача
class_name	VARCHAR	Передбачений клас BCS (напр. "3.75")
confidence	NUMERIC	Впевненість моделі (0–1)
bbox	JSONB	Обмежувальна рамка у JSON-форматі
mask	TEXT	Base64 PNG маски (може бути NULL)
image_base64	TEXT	Base64 вихідного зображення (може бути NULL)
created_at	TIMESTAMP	Дата та час створення прогнозу

Таблиця А.7. API Endpoints Inference Server (Python)

Endpoint	HTTP Метод	Опис
/predict	POST	Приймає зображення у вигляді UploadFile, виконує інференс моделі Mask R-CNN та повертає JSON-відповідь, що містить передбачений клас BCS, впевненість, координати обмежувальних рамок та Base64-кодовану PNG-сегментаційну маску.
/	GET	Базовий endpoint, повертає інформацію про сервер, пристрій та активні оптимізації.
/health	GET	Перевірка стану сервера та доступності GPU, показує зайняту пам'ять GPU.

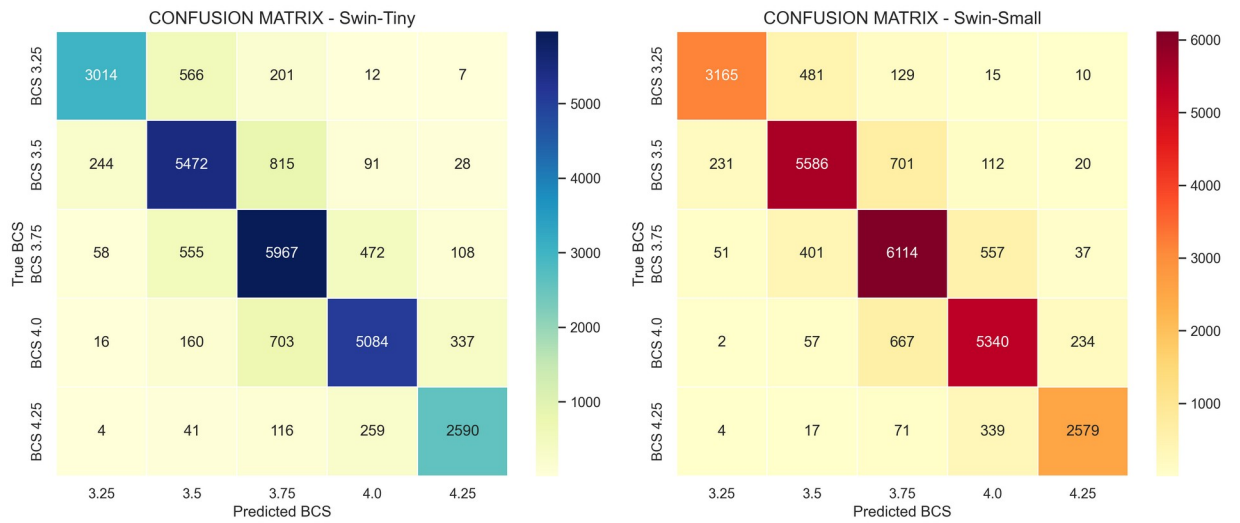


Рисунок А.9 – Матриці плутанини



Рисунок А.10 - Візуалізації масок отриманих через псевдо-розмітку

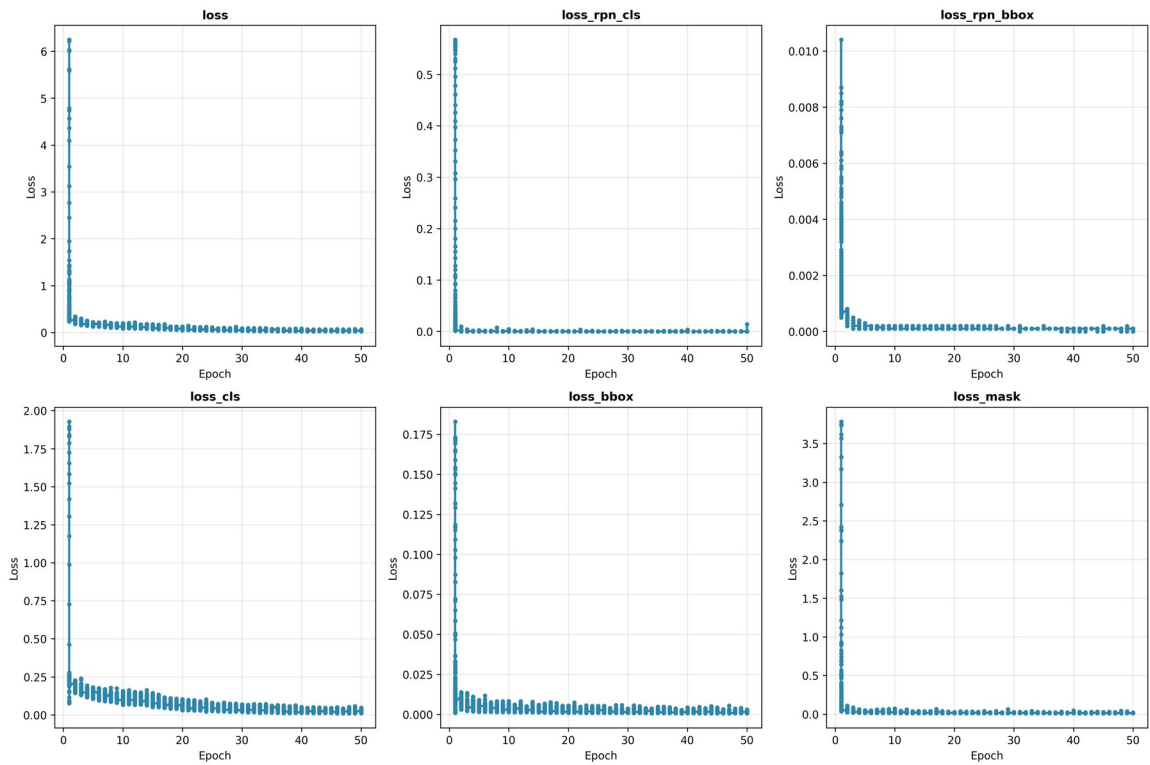


Рисунок А.11 - Графіки втрат Swin-T

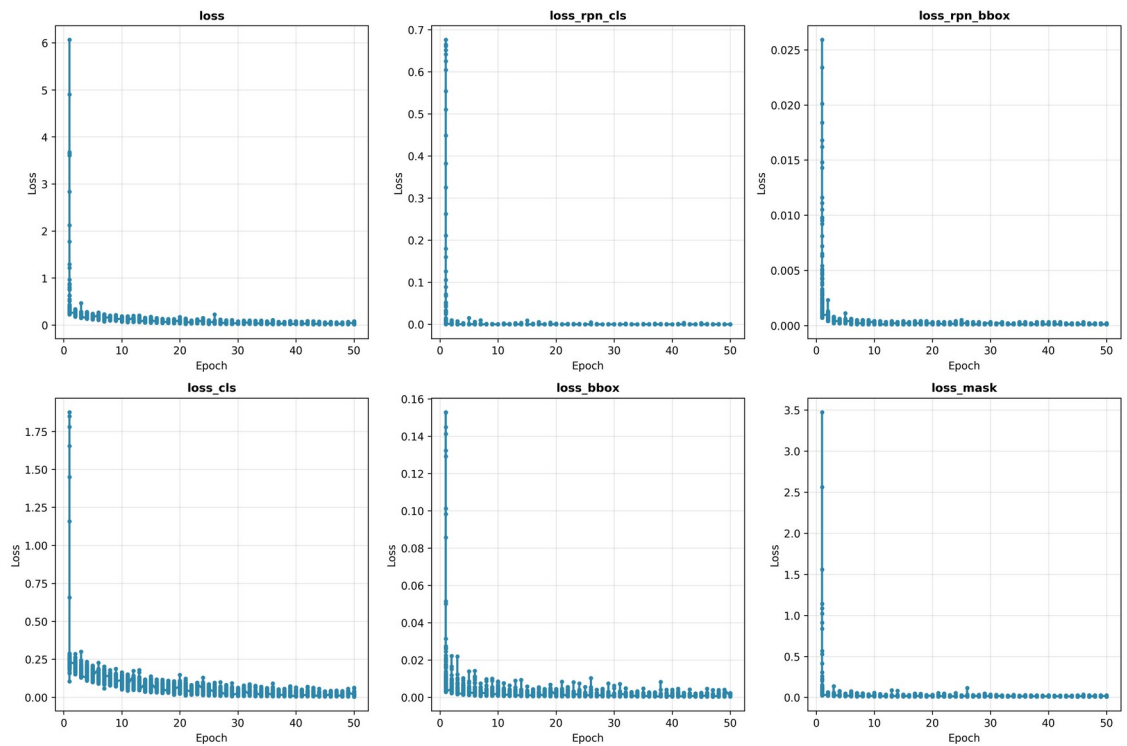


Рисунок А.12 - Графіки втрат Swin-S

Таблиця А.8. Метрики Average Precision (AP)

Модель/ Метрика	bbox_mAP @[.5:.95]	bbox_mAP @0.50	bbox_mAP @0.75	segm_mAP @[.5:.95]	segm_mAP @0.50	segm_mAP @0.75
Swin-T	0.943	0.952	0.951	0.947	0.952	0.952
Swin-S	0.946	0.951	0.949	0.947	0.951	0.951

Таблиця А.9. Метрики класифікації (F1-score Per Class)

Модель	Клас BCS	Precision	Recall	F1-Score
Swin-T	3.25	0.966	0.983	0.974
	3.5	0.945	0.979	0.962
	3.75	0.940	0.981	0.960
	4.0	0.949	0.987	0.968
	4.25	0.959	0.982	0.970
Swin-S	3.25	0.970	0.982	0.976
	3.5	0.943	0.967	0.955
	3.75	0.940	0.978	0.958
	4.0	0.946	0.979	0.962
	4.25	0.955	0.971	0.963

Model Comparison: Segmentation & Classification

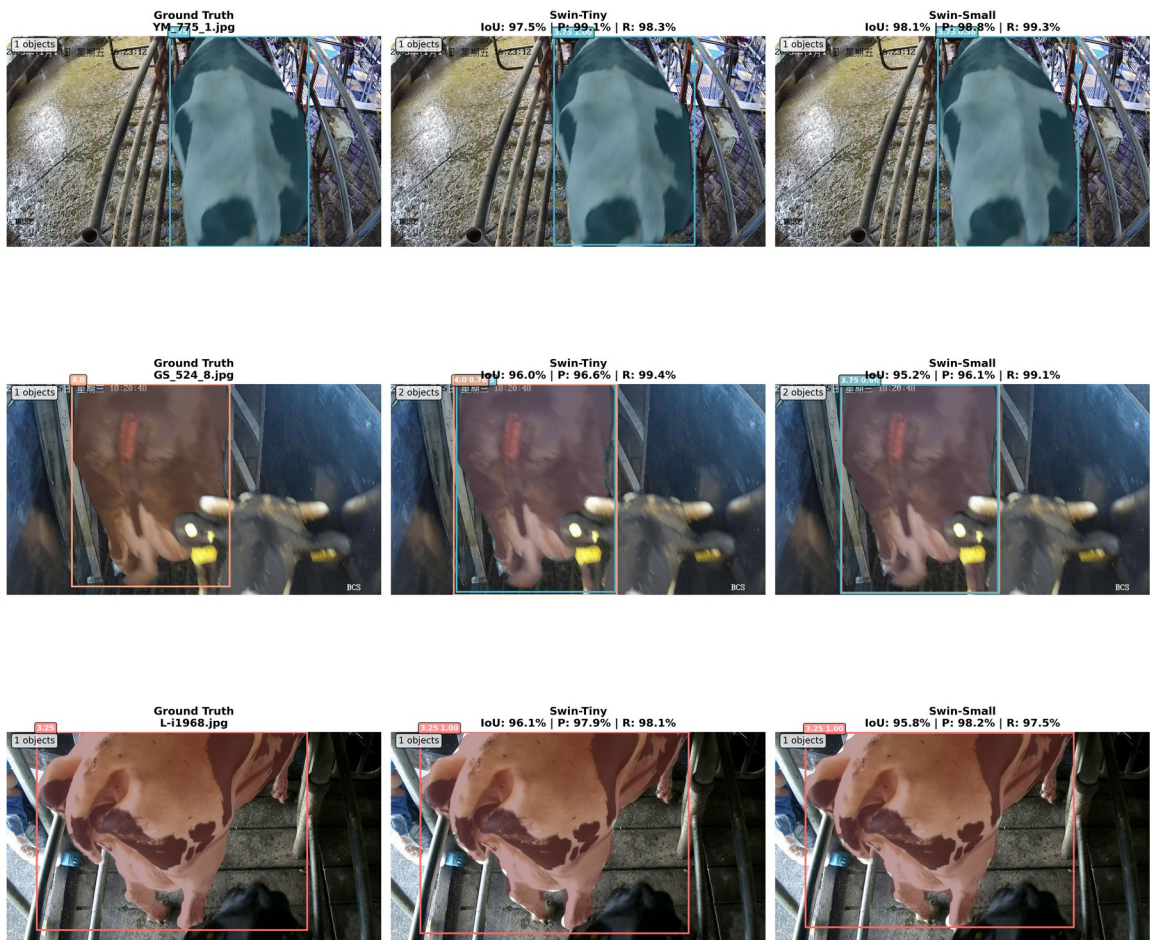


Рисунок 13- Результат роботи моделей

Таблиця А.10. Стрес тест моделі

Кількість користувачів	Кількість запитів	Запити в секунду	Час інференсу секунди	Кількість помилок
25	140	27.9	0.78	0
91	1912	29.4	2.88	0
163	3851	29.6	5.19	0
262	6539	29.7	8.40	0
344	8775	29.7	11.06	0

ДОДАТОК Б. Візуалізація роботи системи на мобільному пристрої

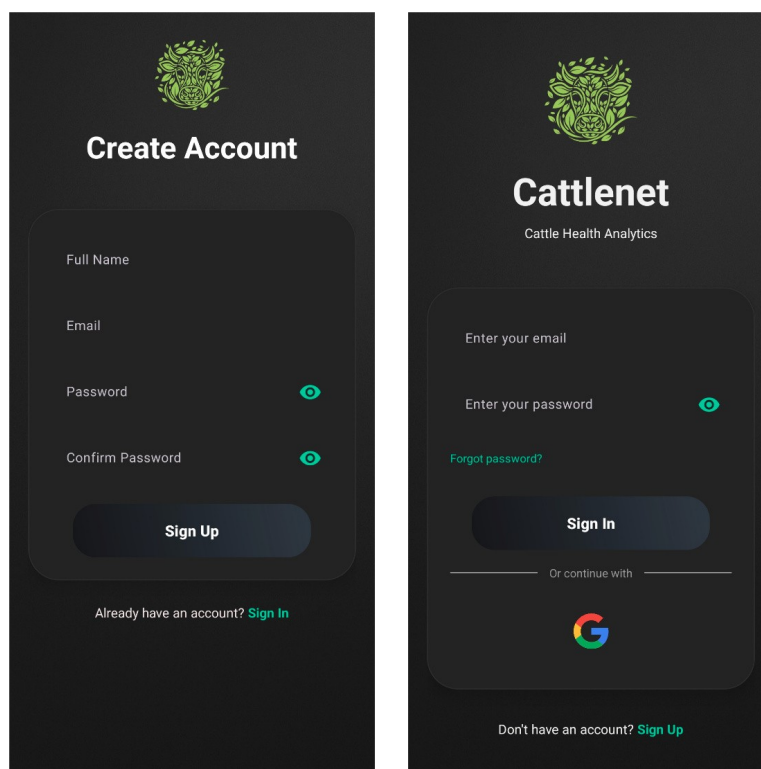


Рисунок Б.1 – Вхід, реєстрація

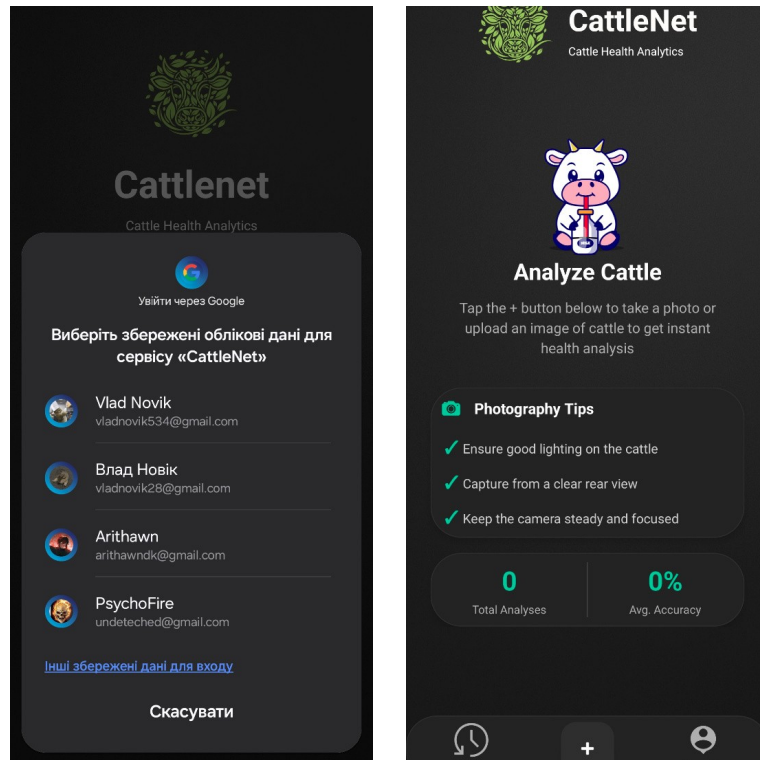


Рисунок Б.2 – Google Sign-in та Головна

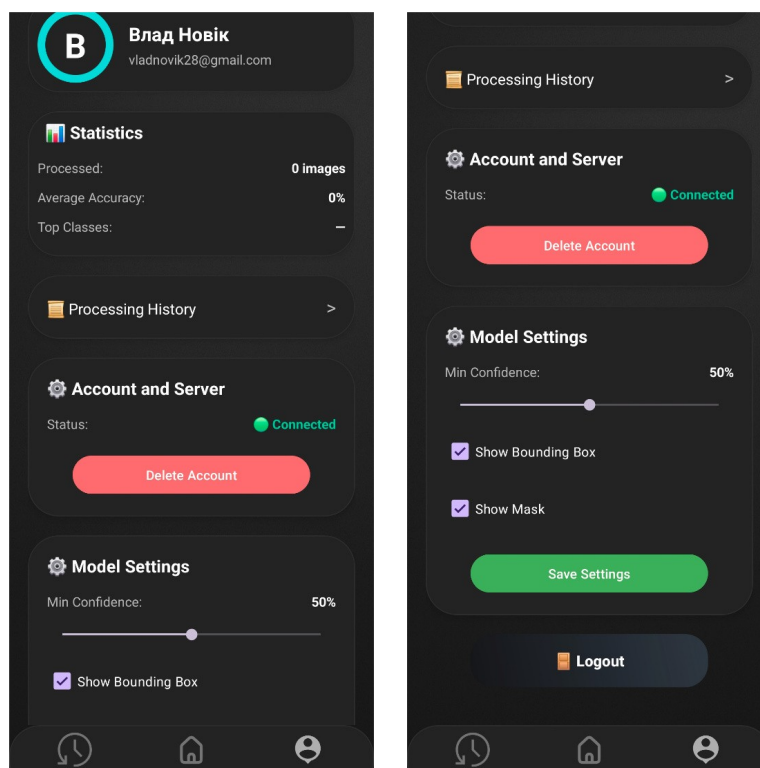


Рисунок Б.3 – Профіль користувача

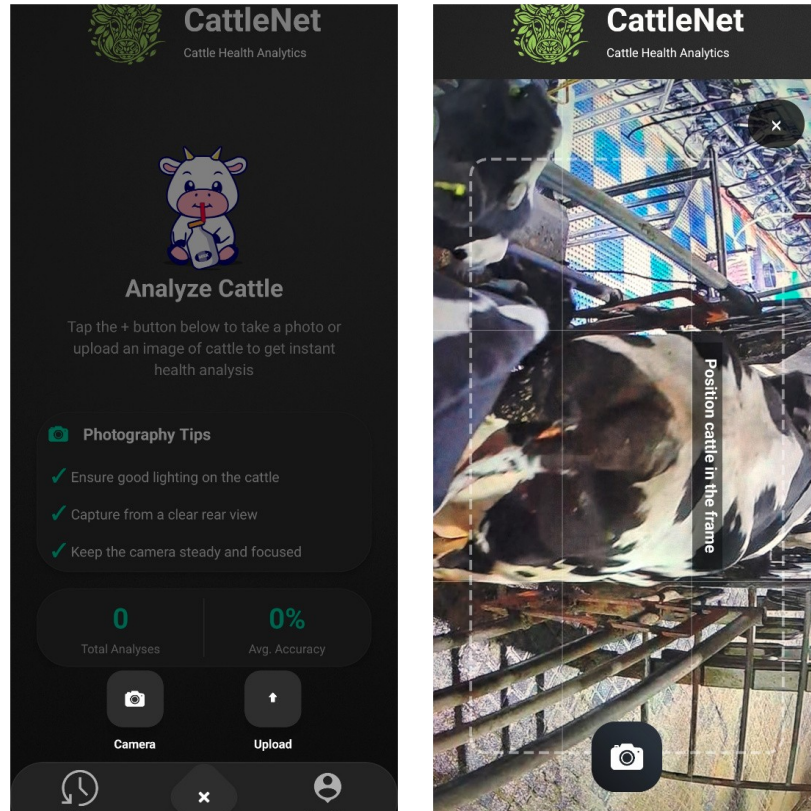


Рисунок Б.4 – Завантаження зображення та зйомка

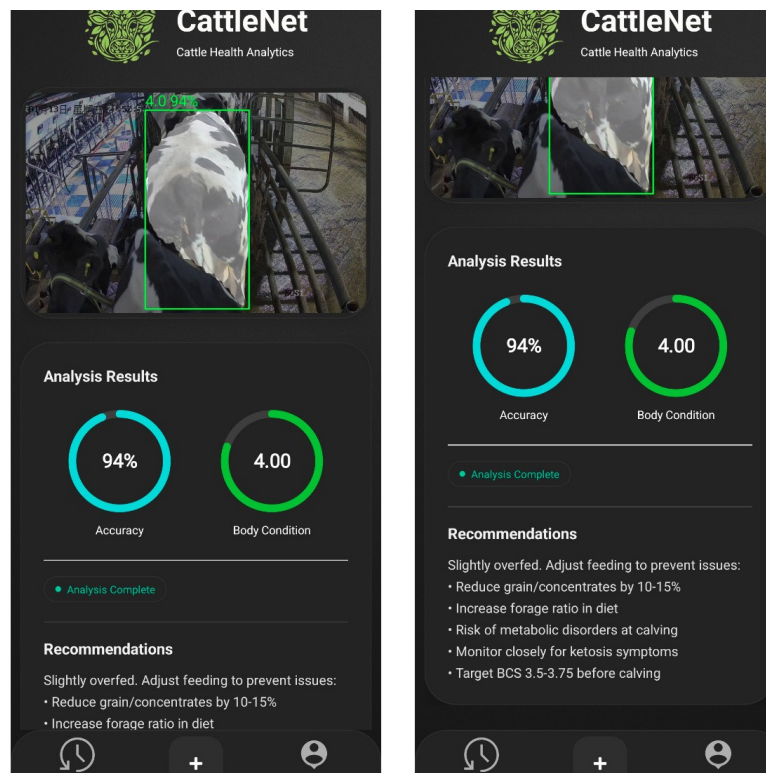


Рисунок Б.5 – Відображення результатів

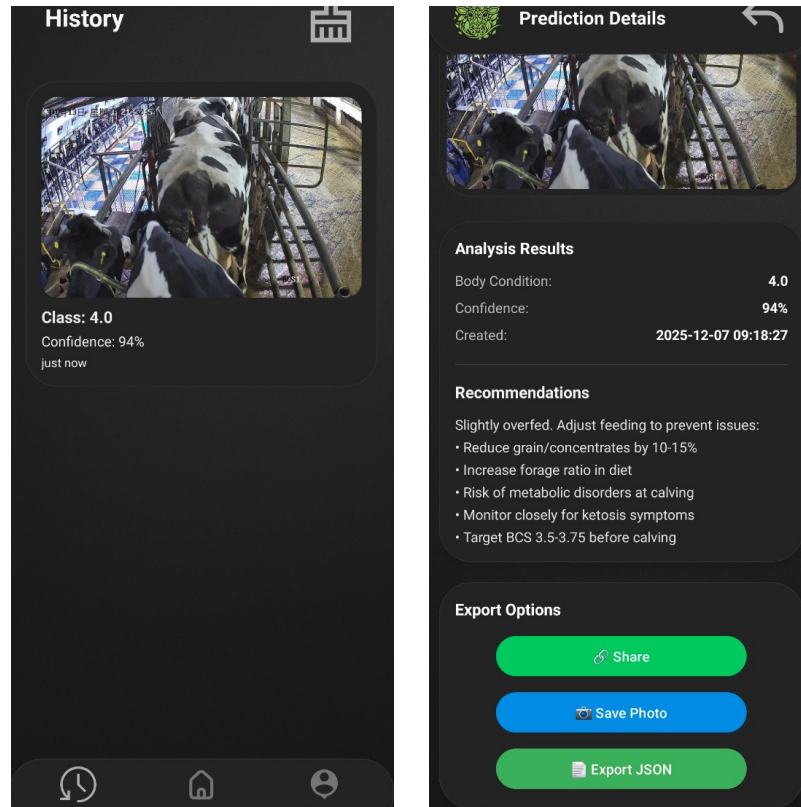


Рисунок Б.6 – Історія та детальний аналіз

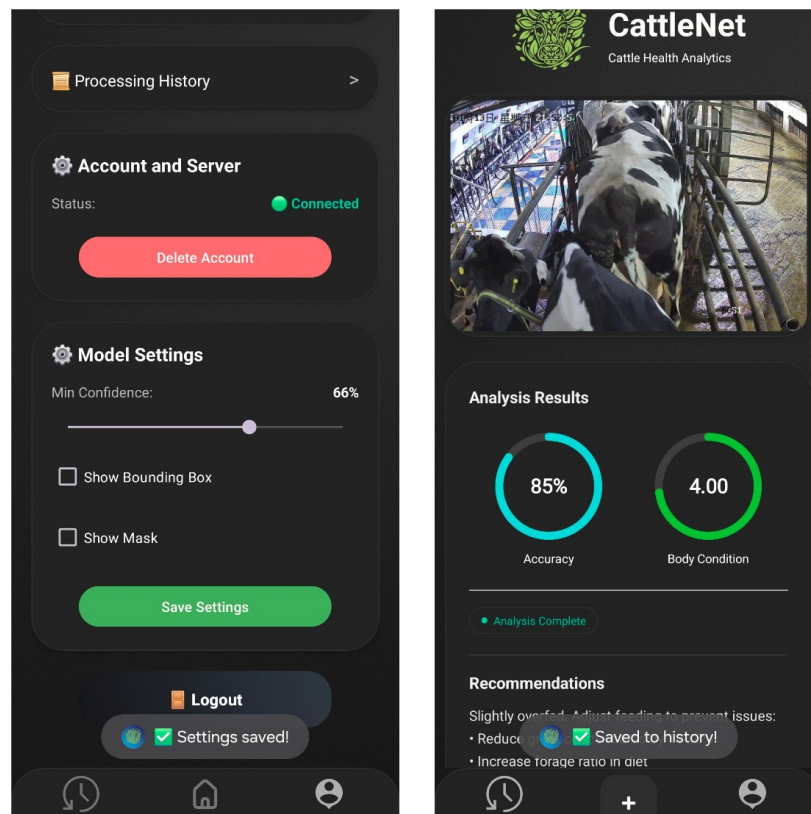


Рисунок Б.7 – Налаштування моделі, відображення маски та bounding box

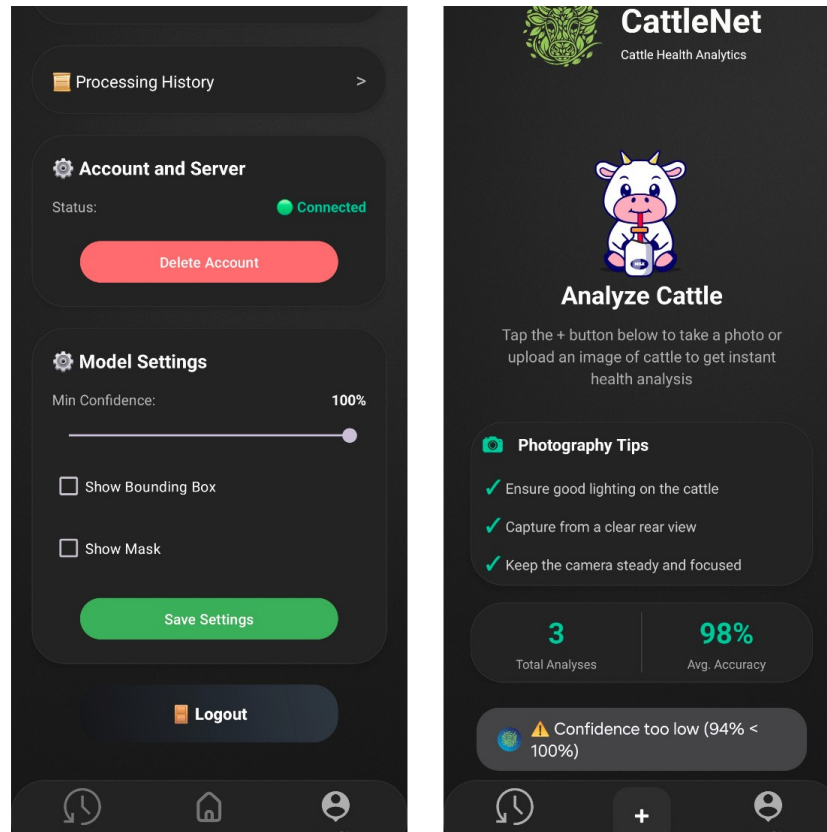


Рисунок Б.8 – Налаштування моделі, точність

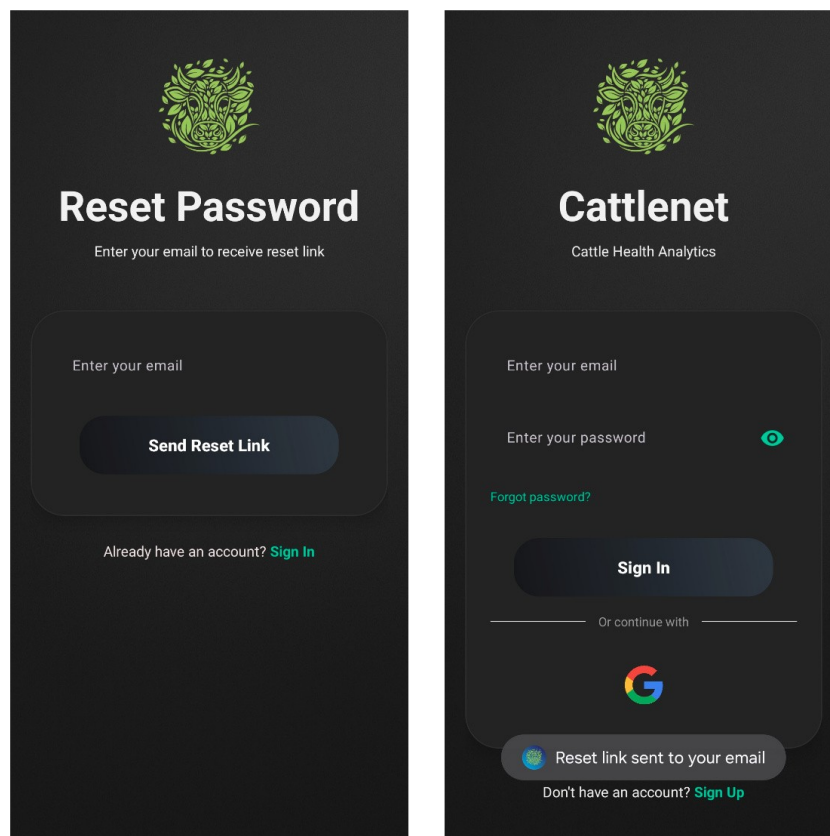


Рисунок Б.9 – Скидання пароля

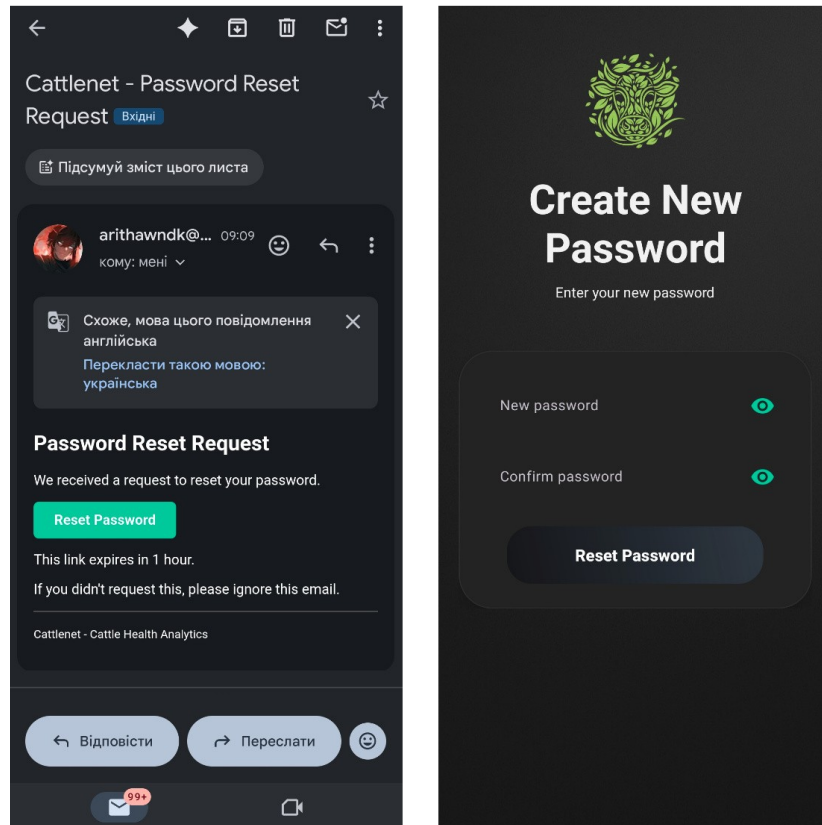


Рисунок Б.10 – Скидання пароля

