

## YOLO-БАЗОВАНА АВТОМАТИЧНА ІДЕНТИФІКАЦІЯ РОГІВ ОЛЕНІВ ДЛЯ ОЦІНКИ СТАНУ ПОПУЛЯЦІЙ

Воронова Н.В.<sup>1</sup>, Горбань В.В.<sup>1</sup>, Литвиненко А.В.<sup>2</sup>, Височін Д.В.<sup>1</sup>

<sup>1</sup>Запорізький національний університет

вул. Університетська, 66, 69011, м. Запоріжжя

<sup>2</sup>Національний університет «Запорізька політехніка»

вул. Університетська, 64, 69011, м. Запоріжжя

180270n@gmail.com, valera251077@gmail.com, lytvart@ukr.net, dmytrovdv7@gmail.com

У сучасних дослідженнях дикої природи значну роль відіграють технології комп'ютерного зору та методи детекції об'єктів, зокрема із застосуванням фотопасток. Ці інструменти дозволяють автоматизувати процеси моніторингу, які раніше вимагали значних витрат часу та людських ресурсів. Автоматизований аналіз зображень сприяє глибшому розумінню екологічних факторів, що впливають на розподіл і поведінку тварин, а також забезпечує ефективне управління природними ресурсами та збереження біорізноманіття. Дослідження присвячене розробці та застосуванню алгоритму автоматичного розпізнавання морфологічних характеристик оленів, зокрема будови їхніх рогів, на основі зображень із фотопасток. Роги оленів є важливими індикаторами віку, статусу домінування, якості харчування та загального здоров'я особин. Автоматизований аналіз цих ознак дозволяє отримувати цінну інформацію про стан популяції без необхідності вилову тварин чи тривалого спостереження, що має практичне значення для мисливських господарств України. Для вирішення задачі детекції об'єктів використано алгоритм YOLO (You Only Look Once), який характеризується високою швидкістю та точністю завдяки одночасному передбаченню обмежувальних рамок і ймовірностей класів об'єктів за допомогою єдиної згорткової нейронної мережі. На відміну від традиційних методів, YOLO аналізує зображення цілком, враховуючи контекст, що забезпечує надійне розпізнавання об'єктів у складних природних умовах. У дослідженні алгоритм адаптовано для виявлення рогів оленів, що демонструє його універсальність і придатність до вузькоспеціалізованих задач екологічного моніторингу. Результати дослідження показують, що автоматичне розпізнавання морфологічних ознак рогів дає змогу швидко виявляти аномалії, оцінювати генетичну якість і стан здоров'я оленів, а також прогнозувати ризики для популяції. Це сприяє розробці ефективних стратегій збереження та управління популяціями копитних. Отримані дані можуть бути використані для оптимізації роботи мисливських господарств, зокрема для планування заходів зі збереження біорізноманіття та раціонального використання природних ресурсів. Перспективи подальших досліджень включають інтеграцію розробленого алгоритму в автоматизовані системи моніторингу, розширення аналізу на інші фенотипічні ознаки тварин, а також поширення цього підходу на інші регіони України. Подальший розвиток методології може сприяти створенню універсальних інструментів для моніторингу дикої природи, що матиме значення не лише для екології, але й для суміжних галузей, таких як біоінформатика та управління природокористуванням. Застосування алгоритмів комп'ютерного зору, подібних до YOLO, відкриває нові можливості для швидкої та точної оцінки стану популяцій у реальному часі, що є критично важливим для сталого розвитку екосистем. *Ключові слова:* моніторинг, фотопастка, аналіз розподілу, ратичні, популяції, моделювання, комп'ютерний зір, алгоритм YOLO.

**YOLO-Based Automatic Deer Antler Identification for Population Assessment. Voronova N., Gorban V., Lytvynenko A., Vysochin D.**

In modern wildlife research, computer vision technologies and object detection methods, particularly those employing camera traps, play a pivotal role. These tools facilitate the automation of monitoring processes that traditionally demanded considerable time and human resources. Automated image analysis enhances the understanding of ecological factors influencing animal distribution and behavior, while also supporting effective natural resource management and biodiversity conservation. This study focuses on the development and implementation of an algorithm for the automatic recognition of morphological characteristics of deer, specifically the structure of their antlers, using images captured by camera traps. Deer antlers serve as critical indicators of age, dominance status, nutritional quality, and overall individual health. The automated analysis of these traits provides valuable insights into population conditions without requiring animal capture or extended observation periods, which is particularly relevant for hunting farms in Ukraine. To tackle the object detection challenge, the YOLO (You Only Look Once) algorithm was utilized. This algorithm is distinguished by its high speed and accuracy, achieved through the simultaneous prediction of bounding boxes and class probabilities within a single convolutional neural network. Unlike conventional approaches, YOLO processes the entire image holistically, incorporating contextual information, which ensures robust object recognition under complex natural conditions. In this research, the algorithm was tailored to detect deer antlers, showcasing its adaptability and effectiveness for specialized tasks in ecological monitoring. The findings demonstrate that the automatic recognition of antler morphology enables rapid identification of anomalies, evaluation of genetic quality and health status of deer, and prediction of population risks. These outcomes support the formulation of effective conservation strategies and the management of ungulate populations. The resulting data can optimize the operations of hunting farms, particularly in planning biodiversity conservation measures and the sustainable use of natural resources. Future research prospects include integrating the developed algorithm into automated monitoring systems, extending the analysis to additional phenotypic traits of animals, and applying this approach to other regions of Ukraine. Further refinement of the methodology could lead to the creation of versatile tools for wildlife monitoring, with implications not only for ecology but also for related disciplines such as bioinformatics and natural resource management. The deployment of computer vision algorithms like YOLO unlocks new possibilities for rapid and precise real-time assessment of population status, a factor of critical importance for the sustainable development of ecosystems. *Key words:* monitoring, camera trap, distribution analysis, ungulates, populations, modeling, computer vision, YOLO algorithm.

**Постановка проблеми.** Застосування технологій комп'ютерного зображення та сучасних методів виявлення об'єктів за допомогою фотопасток набуває більшого значення в дослідженнях дикої природи. Ці технології дозволяють автоматизувати процеси моніторингу, які раніше вимагали значної участі людини, що сприяє оптимізації витрат часу та ресурсів. Наприклад, у роботі Бельда та співавт. [1] продемонстровано можливість використання даних фотопасток у поєднанні з ландшафтними індексами для моделювання розподілу копитних у середньоморському ландшафті. Отримані результати підкреслюють важливість таких даних для з'ясування екологічних чинників, що виділяють розподіл ратичних, і дають цінну інформацію для розробки стратегій управління дикою фауною та лісовими ресурсами.

Проте залишаються виклики, пов'язані з використанням цих технологій для точного й автоматизованого фенотипування ознак диких тварин, зокрема морфологічних характеристик рогів оленів, які є ключовими показниками віку, стану здоров'я та динаміки популяції. У контексті мисливських господарств України можливість ефективного моніторингу морфології рогів за допомогою зображення із фотопасток могла б спростити оцінку стану популяції, дослідження поведінки та аналіз здоров'я тварин. Ручний аналіз таких ознак є трудомістким і часто суб'єктивним, що зумовлює потребу в автоматизованих рішеннях, здатних швидко виявити аномалії та оцінити ризики для підтримки зусиль зі збереження популяції [2, 3].

Перспективним підходом до розв'язання цих проблем є використання передових алгоритмів виявлення об'єктів, таких як YOLO (You Only Look Once) [4]. На відміну від традиційних методів, які аналізують окремі фрагменти зображення, YOLO обробляє зображення цілісно, враховуючи контекст, що забезпечує точність розпізнавання об'єктів у складних природних умовах. Ця робота спрямована на подолання цієї прогалини шляхом застосування алгоритму YOLO для автоматизованого виявлення характеристик рогів на зображеннях із фотопасток, демонструючи адаптивність сучасних методів машинного навчання до специфічних потреб моніторингу популяції ратичних. Розв'язання цих викликів сприятиме розробці надійних інструментів для управління дикою природою та збереження біорізноманіття.

**Актуальність дослідження.** Дослідження, спрямовані на автоматизований аналіз морфологічних характеристик рогів оленів, мають високу актуальність у контексті сучасних екологічних і природоохоронних викликів. Роги оленів є важливими індикаторами віку, статусу домінування, якості харчування, генетичної здатності до виживання та загального здоров'я особин. Наприклад, розмір і симетрія рогів можуть відображати рівень доступу до поживних ресурсів і генетичну стійкість, що

є ключовими факторами для оцінки життєздатності популяцій. У сучасних умовах, коли антропогенний тиск на природні екосистеми зростає, а ресурси для традиційних методів моніторингу обмежені, автоматизовані підходи до аналізу зображень із фотопасток забезпечують швидке й ефективне отримання даних без необхідності вилову тварин чи тривалого польового спостереження.

**Зв'язок авторського доробку із важливими науковими та практичними завданнями.** Застосування комп'ютерного зору, зокрема алгоритмів YOLO, дозволяє дослідникам, екологам і працівникам мисливських господарств отримувати цінну інформацію про стан популяцій оленів, включаючи виявлення аномалій, оцінку генетичної якості та прогнозування ризиків для популяцій. Це сприяє розробці обґрунтованих стратегій управління природними ресурсами та збереження біорізноманіття. В Україні, де мисливські господарства відіграють значну роль у підтримці популяцій ратичних, такі дослідження мають особливе значення, оскільки вони дають змогу оптимізувати заходи зі збереження видів і раціонального використання ресурсів. Крім того, автоматизований аналіз фенотипічних ознак відкриває перспективи для масштабування методології на інші види та регіони, що робить такі дослідження актуальними не лише для екології, але й для суміжних дисциплін, таких як біоінформатика та управління природокористуванням. Алгоритм YOLO, на відміну від вільних двоетапних методів, використовує єдину згортову нейронну мережу для одночасного визначення об'єктів та їх класів, значно прискорюючи детекцію [5].

**Аналіз останніх досліджень та публікацій.** Виявлення та оцінка стану об'єктів у реальному часі є ключовим елементом сучасних технологій у таких галузях, як автономні транспортні засоби, робототехніка, відеоспостереження та доповнена реальність. Серед різних алгоритмів виявлення об'єктів, фреймворк YOLO забезпечує швидку та надійну ідентифікацію об'єктів на зображеннях. Можливості YOLO щодо виявлення об'єктів у режимі реального часу описані в автономних транспортних системах, вони дозволяють швидко ідентифікувати та відстежувати різні об'єкти, такі як транспортні засоби, пішоходи [6, 7]. Ці можливості вже застосовані в багатьох галузях, включаючи розпізнавання дій [8]. Моделі YOLO використовувалися в сільському господарстві для виявлення та класифікації сільськогосподарських культур [9, 10], шкідників та хвороб [11]. Вони також були адаптовані для завдань розпізнавання обличчя в біометрії та системах безпеки [12, 13].

На рисунку 1 показано бібліометричну мережеву візуалізацію всіх статей, знайдених у Scopus, зі словом YOLO в назві та відфільтрованих за ключовим словом «виявлення об'єктів». Потім ми вручну відфільтрували всі статті, пов'язані із застосуваннями [14].

Архітектура YOLO розділяє зображення на сітку розміром  $S \times S$  (наприклад,  $13 \times 13$  клітин). Для кожної секції сітки алгоритм передбачає: в обмежувальних рамок (кожна рамка описується координатами центру, шириною, висотою); впевненість у наявності об'єкта в кожній рамці; ймовірності приналежності об'єкта до кожного з класів.

Основними перевагами YOLO є: швидкість роботи (YOLO обробляє зображення в режимі реального часу, що важливо для аналізу відео або роботи на мобільних пристроях у польових умовах); глобальне розуміння контексту (оскільки алгоритм аналізує все зображення цілком, він краще враховує контекст, що зменшує кількість помилкових спрацьовувань (наприклад, менше ймовірно, що гілка дерева буде помилково прийнята за ріг оленя); висока точність детекції [15-16]. Функція втрат YOLO (loss function) оцінює, наскільки передбачення моделі відрізняються від фактичних даних. Вона включає компоненти для оцінки точності локалізації, впевненості та класифікації. Під час навчання модель намагається мінімізувати цю функцію втрат, поступово покращуючи свої передбачення.

**Виділення невирішених раніше частин загальної проблеми.** Для роботи з алгоритмом YOLO для розпізнавання рогів оленя, розроблено середовище на базі Python із використанням бібліотек PyTorch

(фреймворк глибокого навчання), OpenCV (обробка зображення), Matplotlib і NumPy (візуалізація та обчислення), а також Ultralytics (реалізація YOLO v8). Використання графічної карти використовується для прискорення навчання нейронних мереж. Ключовою проблемою є формування репрезентативного набору даних. Для цього зібрано понад 500 зображень оленів із різноманітними умовами: різне освітлення (сонце, хмарність, сутінки), пори року (літній і зимовий фон), ракурси (фронтальний, профіль, поза), вид оленів (благородний, лань, північний) та якість зображення (професійні фото, любительські знімки, фотопастки). Це забезпечує різноманітність даних, необхідну для ефективного навчання моделі [17].

Розроблено та адаптовано алгоритм YOLO v8 для розпізнавання рогів оленів, що є новим у контексті управління популяціями диких тварин в Україні. Створено набір даних (>500 зображень) із різноманітними умовами (освітлення, пори року, ракурси, види оленів, якість зображень) для підвищення узагальнюючої здатності моделі. Використано Python із бібліотеками PyTorch, OpenCV, Matplotlib, NumPy та Ultralytics, оптимізованими для GPU, що забезпечує ефективне навчання моделі. Створені анотації, які вказують точне розташування рогів оленя на кожному зображенні та розмітку обмежувальних рамок

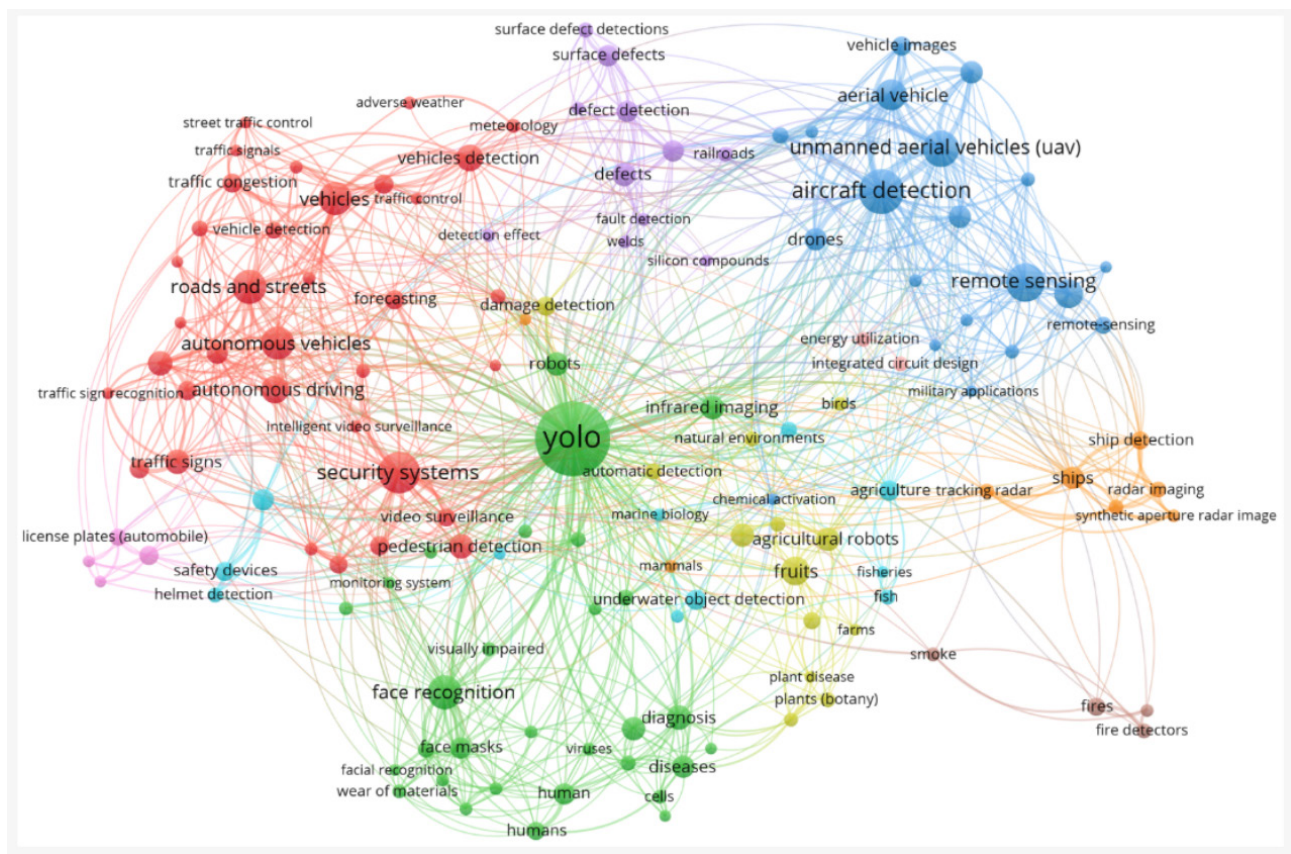


Рис. 1. Бібліометрична мережева візуалізація статей, знайдених у Scopus, зі словом YOLO



(bounding boxes) навколо рогів. Результати дають змогу автоматизувати моніторинг популяцій, оцінювати генетичну якість та прогнозувати ризики, сприяючи збереженню біорізноманіття та раціональному природокористуванню.

#### Методологічне або загальнонаукове значення.

Використано Labellmg для створення анотацій у форматі YOLO. Принципи розмітки: щільне охоплення рогів без включення інших частин тіла, окрема розмітка для кожного набору рогів, врахування частково перекритих рогів.

Анотації зберігаються у текстових файлах із відносними координатами (class\_id, x\_center, y\_center, width, height), що забезпечує універсальність.

Дані структуровано у папки: "images": зображення (.jpg, .png). та "labels": анотації у форматі YOLO.

Обґрунтовано поділ на:

1. тренувальну вибірку (70–80%) — для навчання моделі;
2. валідаційну вибірку (10–15%) — для оцінки та запобігання перенавчанню;
3. тестову вибірку (10–15%) — для фінальної оцінки.

Такий підхід забезпечує точність детекції, узагальнюючу здатність моделі та має значення для методології машинного навчання.

**Виклад основного матеріалу.** Для визначення класів, які модель повинна розпізнавати, було створено файл конфігурації у форматі YAML. У нашому випадку це досить простий файл, оскільки ми маємо лише один клас — «роги оленя». Цей файл визна-

час не тільки класи, але й шляхи до різних вибірок даних, що спрощує процес навчання моделі.

На рисунку 2 елементи відмічені різними кольорами для позначення контексту. Наприклад, роги оленя були відмічені червоним кольором. Також, для додаткової точності були додані відмітки контексту очей та носа, щоб додатково було знайдено голова оленя. Це допомагає уникнути неточностей, коли в кадрі є елементи дерев, які можна сплутати з рогами.

Нейронна мережа YOLO складається з багатьох шарів нейронів, кожен з яких виконує певні математичні операції над даними. Перші шари мережі виявляють прості особливості зображення (лінії, кути, текстур), а глибші шари комбінують ці особливості для розпізнавання складніших патернів і, зрештою, цілих об'єктів. Навчання нейронної мережі — це процес підбору оптимальних значень для мільйонів параметрів (ваг) мережі. Цей процес відбувається за допомогою алгоритму зворотного поширення помилки (backpropagation) та оптимізатора (наприклад, стохастичного градієнтного спуску), який коригує ваги для мінімізації функції втрат.

Для нашого проєкту було обрано модель YOLOv8s — це варіант алгоритму YOLO восьмої версії з «маленькою» (small) архітектурою, яка забезпечує хороший баланс між швидкістю роботи та точністю детекції. Важливим аспектом навчання є використання попередньо навченої моделі (pre-trained model). Це модель, яка вже була навчена на великому наборі даних (наприклад, MS COCO, що містить понад 200,000 зображень з 80 категоріями

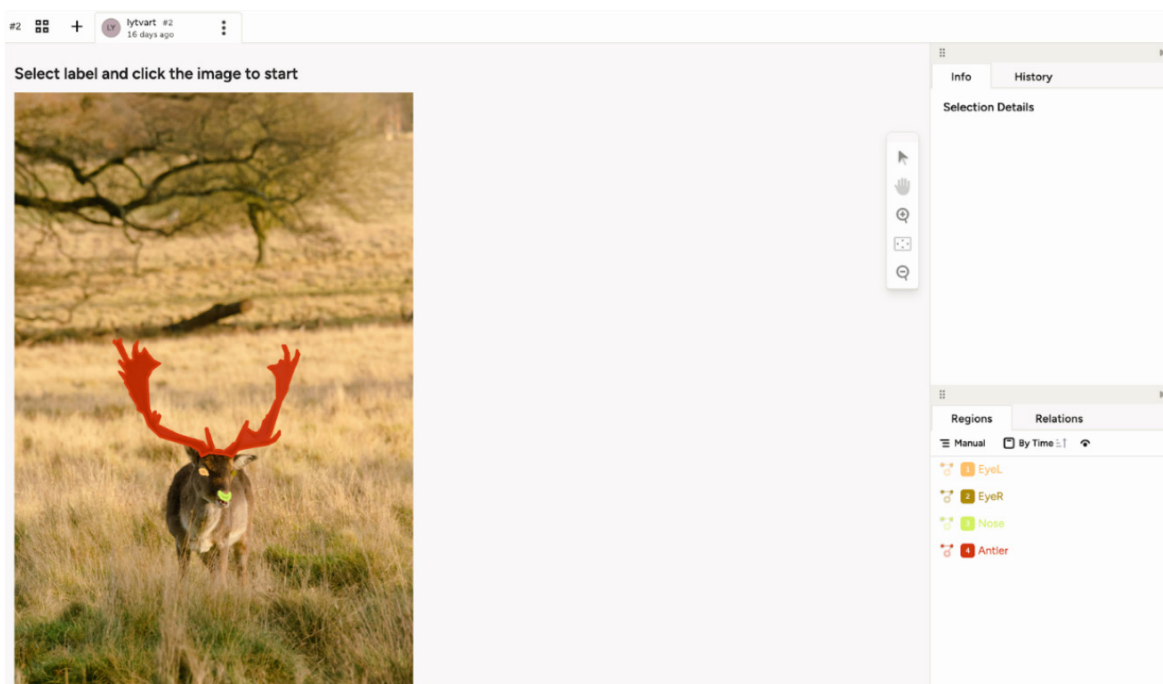


Рис. 2. Особливості позначення елементів для ефективного розпізнавання рогів оленя

об'єктів). Використання такої моделі як відправної точки дозволяє значно скоротити час навчання та покращити результати, особливо при обмеженому наборі даних.

Ми застосували підхід «трансферного навчання» (transfer learning), який базується на ідеї, що нижні шари нейронної мережі, які виявляють базові візуальні особливості (лінії, кути, текстури), можуть бути корисними для багатьох завдань комп'ютерного зору, не тільки для тих, на яких модель була початково навчена.

Для нашого проєкту було налаштовано наступні ключові гіпер параметри:

1. Швидкість навчання (learning rate): 0.01 з експоненційним зменшенням, яка визначає, наскільки сильно параметри моделі змінюються після кожної ітерації. Занадто висока швидкість може призвести до «перестрибування» оптимальних значень, а занадто низька – до повільної збіжності. Експоненційне зменшення означає, що швидкість навчання поступово зменшується з часом, що дозволяє спочатку швидко наблизитися до оптимуму, а потім точно його налаштувати.

2. Розмір групи (batch size): набір з 16 зображеннями – це група зображень, які обробляються одночасно. Більший розмір групи забезпечує стабільне навчання, але вимагає більше пам'яті. Значення 16 було обрано як компроміс між стабільністю та ефективністю використання ресурсів.

3. Кількість епох: 100 Епоха – це один повний прохід через усі тренувальні дані. Чим більше епох, тим довше відбувається навчання і тим краще модель може пристосуватися до даних. Однак занадто багато епох може призвести до перенавчання. Значення 100 було обрано з урахуванням специфіки задачі та обсягу даних.

4. Аугментація даних: включено Аугментація даних – це техніка, яка штучно збільшує різноманітність тренувальних даних шляхом застосування різних трансформацій до наявних зображень (горизонтальне відображення, зміна яскравості, контрасту, насиченості тощо). Це допомагає моделі краще узагальнювати та розпізнавати об'єкти в різних умовах.

Нижче наведено приклад коду для навчання моделі YOLO з використанням бібліотеки Ultralytics:

```
from ultralytics import YOLO
import os
import torch
# Перевіряємо доступність графічної карти
device = torch.device('cuda' if torch.cuda.is_
available() else 'mps')
# Шлях до конфігураційного файлу
config_path = 'data.yaml'
# Завантаження базової моделі
model = YOLO('yolov8s.pt')
# Навчання моделі
results = model.train(
    data=config_path, # шлях до конфігурацій-
ного файлу
```

```
epochs=100, # кількість епох
imgsz=640, # розмір вхідних зображень
batch=16, # розмір батчу
patience=15, # раннє зупинення навчання,
якщо немає покращення впродовж 15 епох
save_period=10, # збереження проміжних
моделей кожні 10 епох
name='deer_antlers_detection', # назва експе-
рименту
augment=True, # застосування аугмен-
тації даних
lr0=0.01, # початкова швидкість
навчання
lrf=0.001, # фінальна швидкість
навчання
cos_lr=True, # використання косинусного
розкладу швидкості навчання
cache=True, # кешування зображень для
прискорення навчання
device=device, # пристрій для навчання
(процесор або відеокарта)
pretrained=True, # використання попе-
редньо навченої моделі
optimizer='SGD', # тип оптимізатора
weight_decay=0.0005, # регуляризація L2
momentum=0.937, # момент для оптимі-
затора SGD
warmup_epochs=3, # кількість епох для
розігріву
close_mosaic=10, # вимкнення мозаїчної
аугментації на останніх епохах
val=True # проведення валідації після
кожної епохи)
# Виведення підсумкових метрик
print(f»Найкраще значення mAP@0.5: {results.
maps[0]}»)
print(f»Найкраще значення mAP@0.5:0.95:
{results.maps[1]}»)

```

Під час навчання модель проходить через наступні етапи для кожної епохи:

1. Прямий прохід (Forward pass): Модель обробляє батч зображень і робить передбачення щодо розташування рогів оленя.

2. Обчислення втрат (Loss calculation): Передбачення порівнюються з фактичними анотаціями, і обчислюється значення функції втрат, яке відображає, наскільки точними були передбачення.

3. Зворотне поширення (Backward pass): Обчислюються градієнти функції втрат відносно всіх параметрів моделі.

4. Оновлення параметрів (Parameter update): Параметри моделі оновлюються за допомогою оптимізатора для мінімізації функції втрат.

5. Валідація: Після обробки всіх тренувальних даних модель оцінюється на валідаційному наборі даних для моніторингу прогресу та запобігання перенавчанню.

Після завершення навчання модель зберігається у файл, який містить всю необхідну інформацію для розпізнавання рогів оленя на нових зображеннях:

1. Архітектура моделі. Це визначення структури нейронної мережі – кількість і типи шарів, як вони з'єднані між собою. У випадку YOLOv8s це специфічна архітектура, оптимізована для задач детекції об'єктів.

2. Ваги моделі. Це мільйони числових параметрів, які модель «вивчила» під час навчання. Саме ці параметри визначають, як модель реагуватиме на різні вхідні дані. Ваги – це основний результат процесу навчання, і вони займають найбільшу частину збереженого файлу.

3. Конфігурація. Це додаткова інформація, необхідна для правильної інтерпретації вхідних даних та вихідних результатів, наприклад: інформація про класи (у нашому випадку один клас – «роги оленя»), параметри попередньої обробки зображень (нормалізація, зміна розміру), параметри постобробки результатів (фільтрація передбачень за порогом впевненості, non-max suppression)

4. Метадані. Інформація про версію моделі, дату навчання, використані гіперпараметри, досягнуті показники якості тощо.

Файл моделі може мати різні формати (.pt, .pth для PyTorch; .h5 для Keras; .onnx для відкритого формату ONNX). У випадку YOLOv8 від Ultralytics моделі зазвичай зберігаються у форматі .pt (PyTorch).

**Головні висновки.** Після навчання, оптимізації та тонкого налаштування модель продемонструвала наступні показники на тестовому наборі даних:

1. Precision (точність): 91.4% – це означає, що з усіх об'єктів, які модель позначила як роги оленя, 91.4% дійсно були рогами оленя. Висока точність свідчить про низький рівень хибно-позитивних результатів.

2. Recall (повнота): 88.7% – це означає, що модель виявила 88.7% усіх рогів оленя, присутніх на тестових зображеннях. Цей показник вказує на те, що модель дуже рідко пропускає роги оленя.

3. F1-Score: 90.0% – це гармонічне середнє точності та повноти, яке дає збалансовану оцінку якості моделі. Високе значення F1-Score (близьке до 90%) свідчить про хороший баланс між точністю та повнотою.

4. mAP@0.5: 0.912 – це середня точність при порозі IoU 0.5. Це означає, що модель досягає середньої точності 91.2%, коли обмежувальна рамка вважається правильною, якщо вона перекривається з фактичною рамкою не менше ніж на 50%.

5. Швидкість інференсу: 45 кадрів/с на GPU Mac M3 MAX – це показник того, як швидко модель може обробляти зображення. Швидкість 45 кадрів в секунду означає, що модель може аналізувати відеопотік у режимі реального часу, що важливо для багатьох практичних застосувань.

**Перспективи використання результатів дослідження.** Для практичної демонстрації роботи системи було розроблено простий інтерфейс, що дозволяє обробляти як окремі зображення, так і відеопотік. Нижче наведено приклад коду для такої демонстрації:

```
from ultralytics import YOLO
model = YOLO('yolo8/deer-seg.pt')
model = YOLO('Users/artem/env/deer-detect/runs/segment/train3/weights/best.pt')
results = model('dataset/deer1.jpg')
results[0].show()
```

І маємо такий результат детекції рогів оленів, рисунок 3:

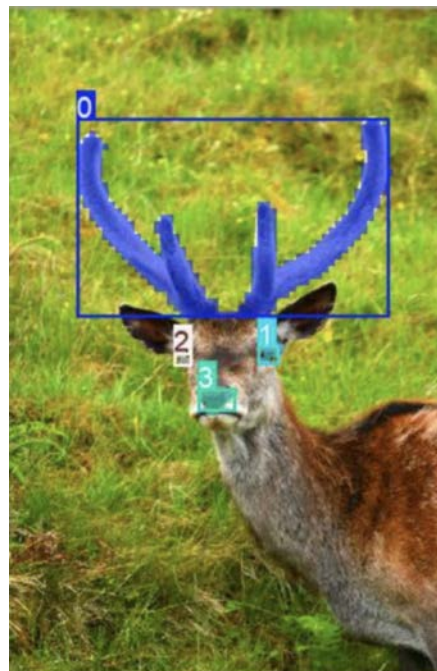


Рис. 3. Результат детекції рогів оленя для обробки фото- та відео

Майбутні напрямки досліджень включають інтеграцію алгоритму в автоматизовані системи моніторингу, аналіз інших фенотипічних ознак та поширення цього підходу на інші мисливські господарства в Україні.

### Література

1. Belda A., Oltra-Crespo S., Miró-Martínez P., Zaragoza B. Can spatial distribution of ungulates be predicted by modeling camera trap data related to landscape indices? A case study in a fragmented mediterranean landscape. *Caldasia*. 42(1). 2020. P. 96–104. <https://doi.org/10.15446/caldasia.v42n1.76384>
2. Височін Д.В., Горбань В.В. Популяційна динаміка та використання ресурсів ратичних у Полтавському регіоні (2010–2023). *XI Регіональна, I Всеукраїнська науково-практична конференція «Актуальні проблеми та перспективи розвитку природничих, медичних та фармацевтичних наук»*. 26 квітня 2025 р. 2025. С.113–114.

3. Vysochyn D.V., Lytvynenko A.V. Assessment of the ecological status of deer populations in the «taganchanske» hunting estate, divytsia tract, Cherkasy region, using antler phenotype analysis. XI Регіональна, I Всеукраїнська науково-практична конференції «Актуальні проблеми та перспективи розвитку природничих, медичних та фармацевтичних наук». 26 квітня 2025 р. 2025. С.115-116.
4. Казіонов М., Скрипник Т., Пасічник О., Вознюк, Л. Вдосконалений метод виявлення БПЛА за тепловізійними зображеннями на основі моделі глибокого навчання YOLO. Herald of Khmelnytskyi National University. Technical Sciences. 2025. 347(1), P. 571-575. <https://doi.org/10.31891/2307-5732-2025-347-79>
5. Тервен Х., Кордова-Еспарза Д.М., Ромеро-Гонсалес Х.-А. Повний огляд архітектур YOLO у комп'ютерному зорі: від YOLOv1 до YOLOv8 та YOLO-NAS. Mach. Learn. Knowl. Extr., 2023. 5(4), С. 1680–1716. DOI: <https://doi.org/10.3390/make5040083>.
6. Lan W.; Dang J.; Wang Y.; Wang S. Pedestrian detection based on YOLO network model. In *Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA)*. Changchun, China, 5–8 August 2018; P. 1547–1551.
7. Hsu W.Y.; Lin W.Y. Adaptive fusion of multi-scale YOLO for pedestrian detection. IEEE Access 2021. 9, P. 110063–110073.
8. Shinde S.; Kothari A.; Gupta V. YOLO-based human action recognition and localization. *Procedia Comput. Sci.* 2018. 133. P. 831–838.
9. Tian Y.; Yang G.; Wang Z.; Wang H.; Li E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* 2019. 157. P. 417–426.
10. Wu D.; Lv S.; Jiang M.; Song H. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments. *Comput. Electron. Agric.* 2020. P. 178, 105742.
11. Lippi M.; Bonucci N.; Carpio R.F.; Contarini M.; Speranza S.; Gasparri A. A. Yolo-based pest detection system for precision agriculture. In *Proceedings of the 2021 29th Mediterranean Conference on Control and Automation (MED)*. Puglia, Italy, 22–25 June 2021; P. 342–347.
12. Wang Y.; Zheng J. Real-time face detection based on YOLO. In *Proceedings of the 2018 1st IEEE International Conference on knowledge innovation and Invention (ICKII)*, Jeju, Republic of Korea. 23–27 July 2018; P. 221–224.
13. Chen W.; Huang H.; Peng S.; Zhou C.; Zhang C. YOLO-face: A real-time face detector. *Vis. Comput.* 2021. 37. P. 805–813.
14. VOSviewer. VOSviewer: Візуалізація наукових ландшафтів. 2023. Доступно онлайн: <https://www.vosviewer.com/> (дата звернення: 11.04.2025 р.).
15. Веб-технології для обробки відеоданих. URL: <https://surl.li/ydktej> (дата звернення: 07.04.2025).
16. Нейромережі, штучний інтелект, машинне навчання розробка і впровадження. URL: <https://surl.lu/cfuspj> (дата звернення: 10.04.2025).
17. Карпаті А., Фей-Фей Л. Глибокі візуально-семантичні вирівнювання для генерації описів зображень. *CVPR*. 2015. С. 3128–3137.