

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук
(повна назва)

Кафедра _____ Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Система автоматичного розпізнавання видів птахів
на основі глибокого навчання
(тема)

Виконав:
здобувач _____ четвертого _____ року навчання,
групи _____ ІТШ-21-5

Владислав Цимбалістенко
(власне ім'я, прізвище)

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

Тип програми _____ освітньо-професійна
Освітня програма _____ Штучний інтелект
(повна назва освітньої програми)

Керівник _____ доц. Марія Головянко
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____
(підпис)

Олег ЗОЛОТУХІН
(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ перший (бакалаврський) _____

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

Тип програми _____ освітньо-професійна _____

Освітня програма _____ Штучний інтелект _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

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

1. Тема роботи _____ Система автоматичного розпізнавання видів птахів на основі глибинного навчання _____

затверджена наказом університету від 19 травня 2025 р. № 378Ст

2. Термін подання студентом роботи до екзаменаційної комісії 25 червня 2025 р.

3. Вихідні дані до роботи Науково-технічні публікації з обробки зображень, глибинного навчання та комп'ютерного зору, дані з інтернет-джерел про класифікацію видів птахів і орнітологічні набори даних, документація з розробки веб-додатків та фреймворків для Python

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі _____

2) Алгоритм глибинного навчання для розпізнавання птахів _____

3) Реалізація програмного забезпечення _____

4) Демонстрація роботи застосунку _____

РЕФЕРАТ

Пояснювальна записка: 63 с., 11 рис., 1 дод., 21 джерело.

АНСАМБЛЬ МОДЕЛЕЙ, ГЛИБИННЕ НАВЧАННЯ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ВИДІВ ПТАХІВ, COMPUTER VISION, FLASK, PYTORCH, WEB APPLICATION.

Об'єкт дослідження – процес автоматичного розпізнавання об'єктів на зображеннях за допомогою методів глибинного навчання.

Предмет дослідження – алгоритми та архітектури глибинного навчання, що застосовуються для розпізнавання та класифікації видів птахів на основі зображень.

Мета роботи – розробка системи автоматичного розпізнавання видів птахів за зображенням на основі сучасних методів глибинного навчання.

Методи дослідження – огляд наукової літератури з глибинного навчання та орнітологічних даних для аналізу предметної галузі, оцінка існуючих алгоритмів розпізнавання для розробки системи.

ABSTRACT

Bachelor's thesis contains: 63 pp., 11 fig., 1 ann., 21 references.

COMPUTER VISION, CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, ENSEMBLE LEARNING, FLASK, IMAGE PREPROCESSING, PYTORCH, WEB APPLICATION.

Object of Study – the process of automatic object recognition on images using deep learning methods.

Subject of Study – algorithms and architectures of deep learning applied to the recognition and classification of bird species based on images.

Aim of the Work – development of a system for automatic recognition of bird species from images based on modern deep learning methods.

Research methods – analysis of the subject area, review of scientific literature on deep learning and ornithology, and evaluation of existing recognition systems for the development of the system.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ.....	9
1 Аналіз предметної галузі	11
1.1 Вступ до розпізнавання видів птахів	12
1.1.1 Різноманіття та класифікація видів птахів	13
1.1.2 Значення систем автоматичного розпізнавання	14
1.2 Проблеми автоматичного розпізнавання видів птахів.....	15
1.2.1 Мінливість зовнішнього вигляду та вокалу птахів	16
1.2.2 Екологічні та контекстуальні виклики	17
1.3 Огляд глибинного навчання.....	17
1.3.1 Фундаментальні принципи глибинного навчання.....	18
1.3.2 Глибинне навчання в розпізнаванні образів	19
1.4 Поточний стан розпізнавання видів птахів	20
1.4.1 Традиційні методи ідентифікації	21
1.4.2 Підходи розпізнавання з використанням машинного навчання.....	21
1.4.3 Обмеження існуючих систем.....	22
1.5 Потенціал глибинного навчання для розпізнавання видів птахів	23
1.5.1 Переваги над традиційними методами	23
1.5.2 Доречні архітектури глибинного навчання.....	24
1.6 Актуальність проблеми	25
1.6.1 Екологічні та природоохоронні наслідки.....	25
1.6.2 Технологічне та освітнє використання.....	26
1.7 Мета і постановка задачі	27
2 Алгоритм глибинного навчання для розпізнавання птахів	29
2.1 Основи обробки зображень для розпізнавання.....	29
2.2 Моделі для розпізнавання видів птахів	31
2.2.1 Модель ResNet.....	31
2.2.2 Модель DenseNet.....	32

2.2.3	Модель InceptionV3	33
2.2.4	Модель EfficientNet.....	34
2.3	Метод ансамблю.....	35
3	Реалізація програмного забезпечення	37
3.1	Фреймворки для веб-додатків на Python	37
3.1.1	Легкі фреймворки	38
3.1.2	Повностекові фреймворки	39
3.1.3	Вибір фреймворку.....	40
3.2	Стратегії розгортання моделей глибокого навчання	40
3.2.1	Пряме завантаження	41
3.2.2	Окремий сервер.....	43
3.2.3	Вибір стратегії розгортання моделей.....	45
3.3	Підходи до обробки та інтеграції даних	46
3.3.1	Підходи до обробки зображень	46
3.3.2	Методи інтеграції зовнішніх наборів даних	48
3.4	Технології бекенду	51
3.4.1	Фреймворк Flask	51
3.4.2	Бібліотеки PyTorch та torchvision.....	52
3.4.3	Бібліотека Pillow	52
3.4.4	Бібліотека NumPy.....	53
3.5	Технології фронтенду	53
4	Демонстрація роботи застосунку	55
	Висновки	60
	Перелік джерел посилання	61
	Додаток А Відомість кваліфікаційної роботи	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AI – Artificial Intelligence – штучний інтелект.

API – Application Programming Interface – прикладний програмний інтерфейс;

CNN – Convolutional Neural Network – згорткові нейронні мережі;

DL – Deep Learning – глибинне навчання;

FLOP – Floating Point Operations – операції з плаваючою комою;

HTML – HyperText Markup Language – мова розмітки гіпертексту;

UI – User Interface – користувацький інтерфейс.

ВСТУП

Автоматичне розпізнавання видів птахів залишається ключовим викликом в екологічних дослідженнях, зумовленим нагальною потребою моніторингу пташиного біорізноманіття в умовах зростаючого екологічного тиску. Традиційні методи ідентифікації, що покладаються на людський досвід, використовують польові путівники, біноклі та слуховий аналіз для розрізнення видів за оперенням, морфологією чи вокалізацією, але ці підходи мають проблеми з масштабуванням, коли стикаються з величезними масивами даних, що генеруються сучасними екологічними дослідженнями. Ранні обчислювальні методи, включаючи базові алгоритми машинного навчання, такі як машини опорних векторів, запровадили автоматизацію, але вимагали значної ручної обробки даних, що обмежувало їхню здатність впоратися зі складністю та мінливістю ознак птахів у різних середовищах існування.

Історично ідентифікація птахів почалася з ретельних спостережень натуралістів, що спиралися на детальні замальовки та нотатки, розвинувшись у 20 столітті з появою аудіозаписів і таксономічних баз даних. Впровадження машинного навчання в 1990-х роках ознаменувало перехід до автоматизації, хоча й обмежений обчислювальними обмеженнями. Нещодавні досягнення в галузі нейронних мереж зробили революцію в цій сфері, дозволивши системам обробляти складні набори даних. Етичні міркування, зокрема прозорість результатів роботи штучного інтелекту та рекомендації щодо відповідального використання в дослідницьких і комерційних контекстах, забезпечують відповідність технології екологічним і соціальним цілям, спираючись на цей багатий історичний фундамент для вирішення сучасних викликів.

Важливість вдосконалення розпізнавання видів птахів полягає в його здатності підтримувати широкомасштабний моніторинг біорізноманіття, необхідний для розуміння здоров'я екосистем і формування стратегій

збереження. Швидке зростання кількості даних, отриманих з таких джерел, як фотопастки, біоакустичні реєстратори та громадські наукові платформи, такі як eBird або iNaturalist, підкреслює попит на автоматизовані системи, здатні обробляти різноманітні вхідні дані з високою точністю. Глибинне навчання з його здатністю вивчати складні закономірності з необроблених візуальних і акустичних даних стає трансформаційним рішенням, пропонуючи неперевершену точність і ефективність порівняно з людським спостереженням або більш ранніми обчислювальними методами.

Отже, застосування глибинного навчання у сфері розпізнавання птахів не лише підвищує точність і ефективність досліджень, але й формує нову парадигму збереження довкілля, в якій технології виступають надійними союзниками у боротьбі за стале майбутнє біорізноманіття.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Розпізнавання видів птахів є важливим завданням у різних галузях, включаючи орнітологію, екологію, природоохоронну біологію та науку про навколишнє середовище. Своєчасна і точна ідентифікація видів птахів є життєво важливою для відстеження динаміки популяцій, оцінки здоров'я екосистем та розробки ефективних стратегій збереження, особливо у світлі зростаючих глобальних проблем, таких як руйнування оселищ, зміна клімату та зменшення біорізноманіття [1]. Птахи слугують важливими індикаторами екологічного благополуччя, пропонуючи цінну інформацію про стан екосистем, що робить їх вивчення пріоритетним як для дослідників, так і для політиків. Проте, велике різноманіття видів птахів, а також складнощі, пов'язані з їхньою ідентифікацією, створюють значні труднощі для традиційних методів, які значною мірою залежать від людського досвіду. Хоча ці традиційні підходи є корисними, вони часто є трудомісткими, суб'єктивними і нелегко масштабуються, особливо коли йдеться про масштабні екологічні дослідження або про зростаючу роль громадської науки. Поява автоматизованих систем розпізнавання видів птахів, створених завдяки розвитку комп'ютерних технологій, таких як глибоке навчання (DL), є революційним рішенням цих проблем. Очікується, що ці системи підвищать ефективність, точність і доступність ідентифікації видів, дозволяючи дослідникам аналізувати великі масиви даних і даючи можливість неспеціалістам долучатися до моніторингу біорізноманіття.

У цьому розділі пропонується поглиблений огляд галузі, досліджуються наукові, технологічні та соціальні аспекти автоматичної ідентифікації видів птахів. У ньому розглядаються виклики, пов'язані з різноманіттям птахів, недоліки сучасних методів ідентифікації та трансформаційний потенціал глибокого навчання в цій галузі. Розглядаючи ці системи в ширшому контексті екологічного та

технологічного розвитку, цей розділ створює основу для розуміння їхньої важливості та обґрунтування цієї тези. Дискурс охоплює історичний розвиток ідентифікації птахів, сучасний стан дисципліни та наслідки впровадження автоматизованих систем для наукових досліджень і залучення громадськості.

1.1 Вступ до розпізнавання видів птахів

Ідентифікація видів птахів передбачає розпізнавання їх за унікальними ознаками, зокрема, за візерунками пір'я, фізичними характеристиками, голосовими звуками чи поведінковими особливостями. Традиційно цим завданням займаються орнітологи, польові біологи та досвідчені б'ордвотчери, які використовують вичерпні польові путівники, таксономічну літературу та свої широкі знання для розрізнення видів. Наприклад, для того, щоб відрізнити вербову славку від зяблика, необхідно ретельно вивчити тонкі відмінності у пісенних візерунках або будові крил, а це вимагає значних знань і досвіду. Хоча людська ідентифікація залишається важливою, вона дедалі більше ускладнюється вимогами сучасної оцінки біорізноманіття.

Зростання доступності даних екологічних оцінок, ініціатив зі спостереження за дикою природою та громадських наукових платформ, таких як eBird та iNaturalist, призводить до того, що ми отримуємо мільйони зареєстрованих спостережень [2]. Ця кількість значно перевершує можливості ручних методів, які часто є трудомісткими і схильними до неточностей через втому спостерігача, упередженість або недостатню підготовку, особливо в складних польових умовах. Автоматизовані системи розпізнавання, які використовують алгоритми глибинного навчання, ефективно усувають ці обмеження. Вони дозволяють швидко і точно аналізувати великі масиви даних, виявляючи закономірності у візуальній або слуховій інформації з надзвичайною точністю. Крім того, ці системи

підвищують доступність ідентифікації видів, дозволяючи орнітологам-аматорам, освітянам і громадським науковцям долучатися до моніторингу біорізноманіття за допомогою інтуїтивно зрозумілих інструментів, таких як мобільні додатки. Інтегруючи традиційні орнітологічні методології з сучасними технологіями, автоматизовані системи революціонізують дослідження птахів. Вони сприяють як науковим дослідженням, так і залученню громадськості до вирішення екологічних питань, сприяючи більш інклюзивному та результативному підходу до досліджень біорізноманіття.

1.1.1 Різноманіття та класифікація видів птахів

Птахи, яких налічується понад 10000 видів, є однією з найрізноманітніших груп хребетних тварин. Вони населяють широкий спектр екосистем, включаючи тропічні ліси, водно-болотні угіддя, арктичні тундри та міське середовище [3]. Таксономічно вони поділяються на близько 40 рядів і 250 родин. Класифікація передбачає інтеграцію морфологічних ознак, генетичної інформації, отриманої за допомогою секвенування ДНК, та поведінкових характеристик, таких як способи добування їжі та шлюбні демонстрації. Ряд Горобцеподібні, до якого відносяться сидячі птахи, такі як горобці, зяблики і солов'ї, представляє більше половини всіх видів птахів і демонструє вражаючу різноманітність розмірів, забарвлення і вокалізації.

Процес класифікації є складним через тонкі відмінності між видами. Наприклад, мухоловку вільхову та мухоловку вербову можна відрізнити за незначними відмінностями у їхніх криках, що вимагає експертних знань для точної ідентифікації. Явище конвергентної еволюції, коли неспоріднені види набувають схожих характеристик, ще більше ускладнює це завдання. Очевидна схожість між європейською та американською вільшанками, які належать до різних родин, ілюструє цю складність. Варіації оперення в різні

пори року, пристосування до конкретних регіонів і міграційні схеми додають складнощів. Крім того, тропічні райони є домівкою для різноманітних видів, які дуже схожі один на одного, а мігруючі види, такі як полярний крячок, вносять додаткові часові коливання.

Гібридизація, яка відбувається при схрещуванні різних видів, призводить до того, що потомство має проміжні характеристики, тим самим ускладнюючи таксономічну класифікацію. Ці складнощі підкреслюють необхідність створення комплексних систем розпізнавання на основі даних. Такі системи повинні бути здатні аналізувати широкий спектр візуальних і акустичних даних, щоб полегшити точну ідентифікацію видів у різноманітних екологічних і таксономічних сценаріях.

1.1.2 Значення систем автоматичного розпізнавання

Автоматизовані системи розпізнавання видів птахів трансформують сферу орнітологічних досліджень та охорони природи. Вони усувають недоліки традиційних методів ідентифікації, значно мінімізуючи час і зусилля, що витрачаються на це [4]. Це є значною перевагою для масштабних екологічних досліджень, таких як відстеження мігруючих популяцій вздовж глобальних маршрутів, таких як Тихоокеанський пролітний шлях. Швидко аналізуючи великі масиви даних, які можуть включати зображення з високою роздільною здатністю, відеокліпи або аудіозаписи, ці системи сприяють оперативному вжиттю природоохоронних заходів. Вони допомагають реагувати на зменшення чисельності популяцій або зміни в оселищах, забезпечуючи тим самим ефективне реагування. Використання алгоритмів глибокого навчання підвищує точність, визначаючи тонкі характеристики.

Наприклад, ці системи можуть розрізняти особливості оперення білоогокого сипухи та вокалізацію дроздів, перевершуючи людських спостерігачів у складних умовах, таких як високий рівень шуму або низька

освітленість. У сфері охорони природи вони сприяють моніторингу видів, що перебувають під загрозою зникнення, в тому числі кулика-лопатня в ізольованих арктичних місцях розмноження. Крім того, вони допомагають виявляти інвазивні види, такі як звичайна мина, яка становить ризик для корінних екосистем. На суспільному рівні ці системи розширюють можливості громадських науковців через такі платформи, як Merlin та eBird, тим самим розширюючи зусилля зі збору даних і сприяючи залученню громадськості до ініціатив зі збереження біорізноманіття.

З освітньої точки зору, ці інструменти залучають як студентів, так і ентузіастів, сприяючи кращому розумінню екологічних проблем. Впровадження технологій в екологічні дослідження за допомогою систем автоматичного розпізнавання дає масштабовані, точні та комплексні рішення. Ці системи сприяють ініціативам, спрямованим на дослідження, збереження та підтримку біорізноманіття птахів в умовах глобальних екологічних викликів.

1.2 Проблеми автоматичного розпізнавання видів птахів

Автоматизовані системи розпізнавання видів птахів мають значний потенціал для революційних змін в екологічних дослідженнях і природоохоронних заходах, проте вони стикаються зі значними перешкодами, які гальмують їхню розробку і впровадження. Ці системи необхідні для точної ідентифікації видів з різних джерел даних, таких як зображення, відео або аудіозаписи, за різних умов – завдання, яке по суті ускладнюється біологічною та екологічною мінливістю.

Основні виклики виникають через різноманітність ознак птахів та нестабільні умови, що зустрічаються в польових умовах, і потребують вдосконалених алгоритмічних підходів. Хоча моделі глибинного навчання є високоефективними, вони потребують всебічного навчання і точного налаштування, щоб вміло орієнтуватися в цих складнощах, гарантуючи тим

самим надійну роботу в практичних сценаріях. У цьому розділі розглядаються ці питання, з акцентом на коливаннях у появі птахів та їхніх голосах, а також на екологічних та контекстуальних елементах, які впливають на точність розпізнавання.

1.2.1 Мінливість зовнішнього вигляду та вокалу птахів

Широке розмаїття понад 10 000 видів птахів у всьому світі має значні відмінності у візуальних і слухових характеристиках, що створює значні труднощі для автоматизованих систем розпізнавання [5]. Візуальні ознаки, включаючи візерунки пір'я, розміри тіла і забарвлення, варіюються не тільки між видами, але і в межах одного виду через вплив таких факторів, як стать, вік або сезонні зміни.

Наприклад, червона крижень змінює своє яскраво-червоне шлюбне оперення на більш приглушені оливкові відтінки в період між розмноженням, що вимагає від моделей врахування цих часових змін. Аналогічно, близькоспоріднені види, такі як чорнодзьобик і вівсянка звичайна, мають незначні відмінності в оперенні, які вимагають точного визначення ознак, щоб запобігти помилковій ідентифікації.

Акустичні характеристики демонструють значну мінливість, оскільки пташині пісні та крики відрізняються залежно від виду, географічного розташування та поведінкового контексту. Дрізд лісовий видає складні, схожі на флейти трелі, тоді як мухоловка вербова видає більш простий, хрипкий поклик, проте регіональні діалекти або індивідуальні відмінності можуть перешкоджати точному розпізнаванню.

Крім того, вокалізація може змінюватися під час залицання або у відповідь на загрозу, що ускладнює аналіз. Щоб ефективно розпізнавати ці тонкі патерни, системи глибинного навчання повинні використовувати великі та різноманітні набори даних, що полегшує точну ідентифікацію видів у різних біологічних та екологічних контекстах.

1.2.2 Екологічні та контекстуальні виклики

На ефективність автоматичних систем розпізнавання птахів значною мірою впливають фактори навколишнього середовища та контексту в природних біотопах, які часто погіршують якість вхідних даних. Фоновий шум, зокрема вітер, вода, що тече, або крики інших тварин, можуть маскувати вокалізацію птахів в аудіозаписах, особливо в густому середовищі, наприклад, у тропічних лісах [6]. Аналогічно, візуальні дані можуть бути ускладнені перешкодами, такими як листя, тіні або недостатнє освітлення, що потенційно приховують важливі ознаки, необхідні для ідентифікації, такі як унікальне маркування білоокої вивірки.

Фактори навколишнього середовища, такі як коливання погоди та зміна часу доби, додають складності до збору даних. Неприятливі умови, такі як дощ або туман, можуть затьмарювати візуальні дані, а записи, зроблені в сутінках, можуть спотворювати звукові частоти, тим самим ставлячи під сумнів надійність моделей. Крім того, наявність непостійної якості датчиків, наприклад, пристроїв з низькою роздільною здатністю або мікрофонів, розташованих на великій відстані, ще більше знижує чіткість даних, що вимагає складних методів попередньої обробки для поліпшення якості вхідних даних. Ці виклики, як екологічні, так і технічні, підкреслюють потребу в стійких і гнучких алгоритмах, які можуть підтримувати надійну роботу розпізнавання в різних реальних сценаріях.

1.3 Огляд глибинного навчання

Глибинне навчання значно покращує автоматичну ідентифікацію видів птахів, дозволяючи системам аналізувати складні набори даних, включаючи зображення високої роздільної здатності та аудіозаписи, з винятковою точністю. Здатність розпізнавати складні закономірності в необроблених даних робить його фундаментальною технологією для

екологічних застосувань, особливо для розпізнавання різних видів птахів за різних умов. Нейронні мережі з декількома шарами лежать в основі глибинного навчання, дозволяючи моделям автономно виокремлювати такі ознаки, як візерунки пір'я або вокальні характеристики. Ця функціональність покращує моніторинг біорізноманіття, допомагаючи дослідникам і захисникам природи вирішувати проблеми, пов'язані з мінливістю видів і втручанням навколишнього середовища.

1.3.1 Фундаментальні принципи глибинного навчання

Глибине навчання використовує нейронні мережі з декількома шарами для представлення складних взаємозв'язків даних, що полегшує виконання складних завдань, таких як ідентифікація видів птахів [7]. Ці мережі складаються з взаємопов'язаних вузлів, розташованих у вхідному, прихованому та вихідному шарах, де кожен вузол обробляє інформацію за допомогою зважених зв'язків і функцій активації для перетворення необроблених вхідних даних у значущі вихідні (рисунок 1.1).

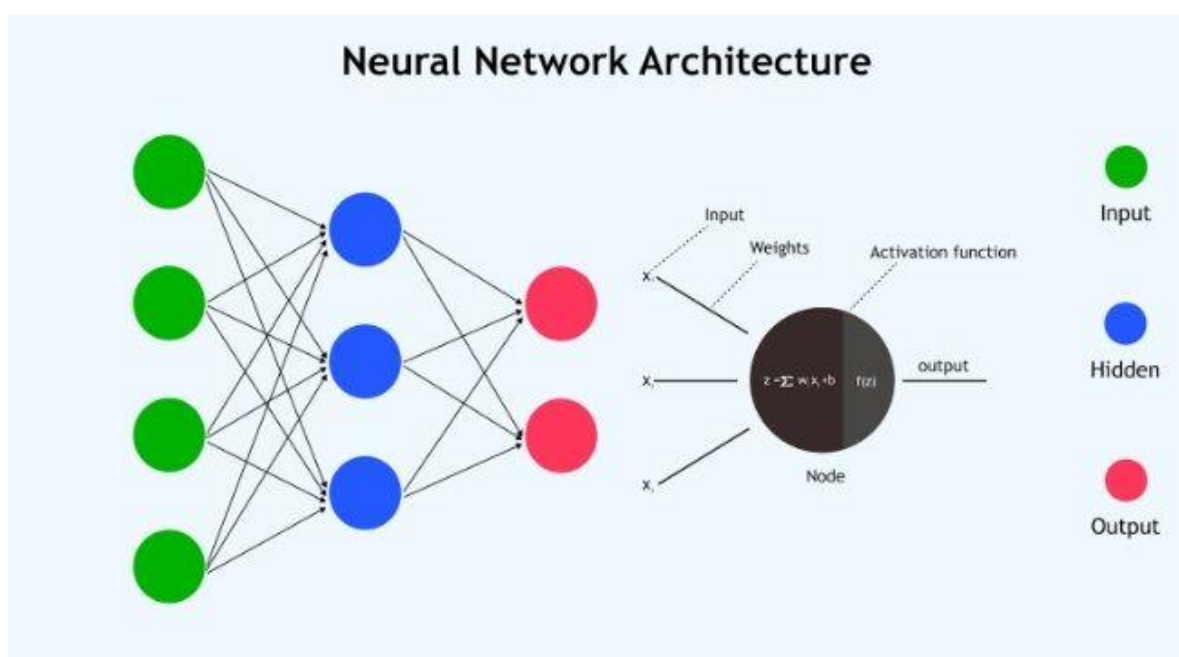


Рисунок 1.1 – Архітектура нейронної мережі

Під час навчання моделі змінюють свої ваги, щоб зменшити неточність прогнозування, використовуючи великі набори даних як орієнтир. Для розпізнавання птахів нейронні мережі набувають здатності ідентифікувати такі характеристики, як візерунки пір'я та частоти вокалізації, використовуючи зворотне поширення та градієнтний спуск для підвищення своєї ефективності. Хоча ця методологія вимагає значних обчислювальних ресурсів, вона дає стійкі моделі, які можуть ефективно узагальнювати різноманітні вхідні дані. Багаторівнева архітектура, притаманна глибинному навчанню, полегшує поступове набуття все більш складних характеристик, починаючи від фундаментальних форм і закінчуючи складними патернами. Така гнучкість необхідна для врахування різноманітності зовнішності та криків птахів, забезпечуючи таким чином точну ідентифікацію в екологічних дослідженнях.

1.3.2 Глибинне навчання в розпізнаванні образів

Глибинне навчання демонструє виняткові можливості в розпізнаванні образів, що робить його особливо вправним в ідентифікації видів птахів як за візуальними, так і за слуховими даними.

Найвідоміші архітектури, зокрема згорткові нейронні мережі (CNN) та рекурентні нейронні мережі, спеціально розроблені для обробки просторових і послідовних даних, відповідно, задовольняючи тим самим чіткі вимоги завдань ідентифікації птахів. Згорткові нейронні мережі зосереджені на аналізі зображень шляхом вилучення просторових характеристик, таких як унікальне очне кільце білоокої вивре або вражаюче забарвлення червоного танагера (рисунок 1.2) [8]. Ці мережі використовують згорткові шари для ідентифікації країв, текстур і специфічних особливостей, ефективно зменшуючи розмірність даних, зберігаючи при цьому важливу інформацію, навіть на захаращеному або зашумленому фоні.

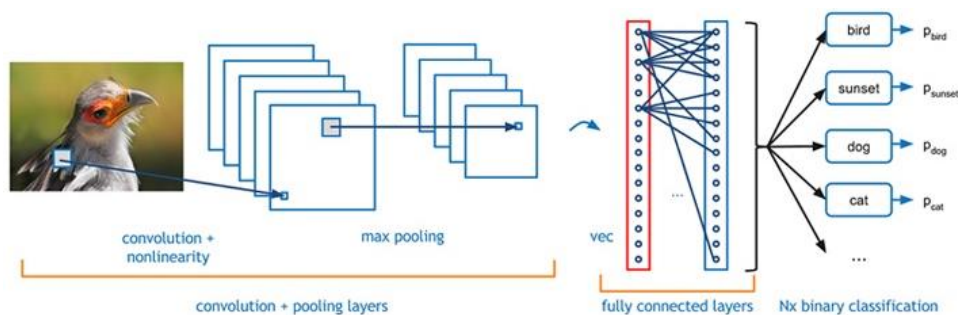


Рисунок 1.2 – Згортовка нейронна мережа

Рекурентні нейронні мережі призначені для аналізу аудіопослідовностей, ефективно ідентифікуючи часові патерни в пташиних вокалізаціях, таких як пташині пісні або дзвінки. Наприклад, вони можуть відрізнити складні трелі дроздів від більш простих покликів вербової мухоловки, вправно керуючи вхідними даними різної тривалості та регіональними вокальними відмінностями. Ці моделі використовують великі набори даних для створення надійних репрезентацій, долаючи такі перешкоди, як фоновий шум і незначні видові відмінності. Їх гнучкість гарантує стабільну роботу в практичних екологічних контекстах, тим самим підвищуючи точність систем, що використовуються для ідентифікації видів птахів.

1.4 Поточний стан розпізнавання видів птахів

Розпізнавання видів птахів є життєво важливим для екологічних досліджень та охорони природи, дозволяючи здійснювати моніторинг біорізноманіття та популяційних тенденцій. Сучасні методи варіюються від традиційних підходів, заснованих на спостереженнях людини, до передових комп'ютерних технологій, кожен з яких робить свій внесок у розвиток цієї галузі.

Технологічний прогрес розширив сферу застосування систем розпізнавання, проте залишаються проблеми з досягненням високої

точності в різних умовах. Розуміння цих методів та їхніх обмежень сприяє розробці більш ефективних рішень для автоматичної ідентифікації видів птахів.

1.4.1 Традиційні методи ідентифікації

Традиційна ідентифікація видів птахів залежить від людського досвіду, а орнітологи та б'юродвотчери використовують методи спостереження для розрізнення видів. Такі інструменти, як польові визначники, таксономічні ключі та біноклі, дозволяють ідентифікувати птахів за візуальними ознаками, такими як оперення, розмір чи поведінка, які часто доповнюються слуховими сигналами з пісень чи криків, як, наприклад, при розрізненні мелодійної пісні вербової колібрі від повторюваних нот зяблика. Для виявлення тонких відмінностей експерти потребують тривалої підготовки, а цей процес є кропітким, але обмеженим у часі та масштабах. Маломасштабні дослідження виграють від такої точності, але метод дає збої при обробці великих масивів даних, отриманих в результаті екологічних досліджень або на платформах громадської науки, таких як eBird, де спостерігачі-аматори надають дані різної якості.

Громадянська наука розширює можливості збору даних, але вносить неузгодженості через різний рівень кваліфікації. Ці методи, хоча і є основоположними для орнітології, не можуть задовольнити потреби швидкого, широкомасштабного моніторингу біорізноманіття, що вимагає автоматизованих альтернатив.

1.4.2 Підходи розпізнавання з використанням машинного навчання

Машинне навчання зробило революцію в розпізнаванні видів птахів, автоматизувавши ідентифікацію за допомогою обчислювальних моделей, що значно підвищило ефективність. Ранні методи використовували

алгоритми на кшталт машин опорних векторів або дерев рішень, покладаючись на створені вручну ознаки, такі як кольорові гістограми для зображень або спектральні частоти для звуку, що вимагало ретельної розробки ознак для досягнення достатньої точності. Сучасні підходи використовують глибинне навчання: згорткові нейронні мережі обробляють зображення, щоб виявити такі ознаки, як очне кільце білоокої вивірки або оперення червоного кроншнепа, а рекурентні нейронні мережі аналізують аудіо, щоб відрізнити трелі дроздів від дзвінків вербової мухоловки. Ці моделі вивчають складні закономірності безпосередньо з необроблених даних, покращуючи продуктивність порівняно з попередніми методами.

Такі додатки, як Merlin Bird ID, інтегрують глибинне навчання, дозволяючи користувачам ідентифікувати види за допомогою додатків для смартфонів, а такі платформи, як iNaturalist, надають великі набори даних для навчання [9]. Ці системи ефективно масштабуються, обробляючи дані з фотопасток, біоакустичних рекордерів або громадянської науки, підтримуючи екологічні дослідження з більшою точністю. Адаптивність глибинного навчання до різних типів даних підвищує його корисність, хоча воно вимагає значних обчислювальних ресурсів і маркованих наборів даних для досягнення оптимальних результатів у реальних сценаріях.

1.4.3 Обмеження існуючих систем

Сучасні системи розпізнавання видів птахів мають суттєві недоліки, які обмежують їхню ефективність. Традиційним методам бракує масштабованості, оскільки люди-спостерігачі намагаються обробити великий обсяг даних сучасного моніторингу, що затримує отримання екологічних знань, критично важливих для збереження природи. Системи машинного навчання, хоч і масштабовані, але не можуть впоратися з мінливістю ознак птахів і шумом навколишнього середовища. Моделі можуть помилково ідентифікувати види зі схожим зовнішнім виглядом, як-

от чорнодзьобик і пеночка, або ж давати збої в шумному середовищі, наприклад, у тропічних лісах, де звуки накладаються один на одного.

Проблеми з якістю даних, включаючи зображення з низькою роздільною здатністю або неповні набори даних, особливо для рідкісних видів, перешкоджають навчанню моделі. Високі обчислювальні витрати і потреба у великих обсягах маркованих даних ще більше обмежують доступність, підкреслюючи необхідність вдосконалення алгоритмів і ресурсів.

1.5 Потенціал глибинного навчання для розпізнавання видів птахів

Глибинне навчання трансформує розпізнавання видів птахів, пропонуючи масштабовані та точні рішення для екологічних досліджень. Його здатність обробляти великі масиви даних покращує моніторинг біорізноманіття та зусилля зі збереження. Удосконалені нейронні мережі дозволяють системам ідентифікувати види за зображеннями або аудіозаписами з високою точністю, що перевершує традиційні методи.

1.5.1 Переваги над традиційними методами

Глибинне навчання перевершує традиційні методи ідентифікації птахів, які залежать від людських навичок і ручного спостереження. Воно ефективно аналізує величезні обсяги даних, включаючи зображення, захоплені фотопастками, та аудіозаписи з біоакустичних пристроїв, зі швидкістю, яку людина не може порівняти, полегшуючи відстеження видів в реальному часі [10]. На відміну від людських методів, які схильні до втоми та упередженості, глибинне навчання забезпечує надійну точність. Ці моделі можуть ідентифікувати складні характеристики навіть за складних обставин, тим самим мінімізуючи помилки, пов'язані з традиційною ідентифікацією за допомогою польових гідів.

Системи глибинного навчання ефективно масштабуються для глобальних наборів даних, підтримуючи такі платформи, як eBird. Вони розширюють можливості громадських науковців за допомогою зручних інструментів, розширюючи участь у дослідженнях біорізноманіття.

Ця технологія також адаптується до різних типів даних, обробляючи як візуальні, так і акустичні дані, на відміну від традиційних методів, обмежених навичками спостерігача або екологічними обмеженнями.

1.5.2 Доречні архітектури глибинного навчання

Ефективність систем розпізнавання видів птахів ґрунтується на архітектурі глибинного навчання, яка використовує передові нейронні мережі для аналізу складних візуальних і слухових даних. Найчастіше використовуються згорткові нейронні мережі та рекурентні нейронні мережі, тоді як інноваційні моделі, такі як трансформатори, отримують все більше визнання за свою адаптивність в екологічному контексті.

Згорткові нейронні мережі є високоефективними в аналізі зображень, вправно виділяючи ієрархію ознак, починаючи від простих країв і закінчуючи складними візерунками, такими як яскраве оперення червонодзьобого кроншнепа та нюансовані позначки чорнодзьобого співочого птаха. Вони використовують згорнуті шари для виявлення просторових зв'язків, об'єднання шарів для мінімізації розмірності та повністю з'єднані шари для класифікації видів, демонструючи стійкість до спотворень зображення та захаращеного фону, що часто зустрічається в природному середовищі.

Рекурентні нейронні мережі спеціально розроблені для роботи з аудіоданими, що дозволяє аналізувати часові послідовності для розпізнавання пташиних пісень або дзвінків. Вони ефективно розпізнають закономірності у вокалізації, дозволяючи відрізнити складні трелі дроздів

від більш простих покликів вербової мухоловки, навіть за наявності регіональних діалектів або звуків, що перекриваються.

Мережі з довгою короткочасною пам'яттю, підтип рекурентних нейронних мереж, покращують продуктивність, підтримуючи довготривалі залежності між довгими аудіопослідовностями.

Архітектури на основі трансформаторів, такі як трансформатори зору для зображень або трансформатори звуку для спектрограм, інтегрують просторовий і часовий аналіз, пропонуючи підвищену точність для мультимодальних даних.

Ці моделі використовують механізми уваги, щоб зосередитися на критичних особливостях, що дозволяє розпізнавати рідкісні види, такі як кулик-лопатень, у різноманітних наборах даних з таких платформ, як iNaturalist. Їх адаптивність до великомасштабних, гетерогенних даних робить їх ідеальними для вивчення мінливості ознак птахів і умов навколишнього середовища.

1.6 Актуальність проблеми

Розпізнавання видів птахів на основі глибинного навчання має трансформаційний потенціал для екологічних досліджень, природоохоронних стратегій та залучення громадськості.

Автоматизуючи точну ідентифікацію видів, ця технологія вирішує нагальні екологічні проблеми, такі як втрата біорізноманіття та зміна клімату, підтримуючи прийняття рішень на основі даних. Її застосування виходить за межі науки, сприяючи технологічним інноваціям та освітній діяльності.

1.6.1 Екологічні та природоохоронні наслідки

Системи глибинного навчання для розпізнавання видів птахів значно просувають екологічні дослідження, уможливаючи швидкий,

широкомасштабний моніторинг пташиних популяцій. Вони обробляють великі масиви даних з фотопасток, біоакустичних реєстраторів або громадських наукових платформ, таких як eBird, відстежуючи такі види, як крячок полярний, на міграційних шляхах [11], щоб виявити популяційні тенденції, переваги середовища проживання та реакцію на зміни навколишнього середовища, що є критично важливими для розуміння здоров'я екосистем. Ці системи підтримують охорону природи, сприяючи захисту зникаючих видів, таких як кулик-лопатень, шляхом раннього виявлення у віддалених оселищах. Вони також виявляють інвазивні види, такі як міна звичайна, які загрожують місцевим екосистемам, що дозволяє вчасно втручатися для відновлення екологічного балансу.

Надаючи точні дані, глибинне навчання зменшує залежність від трудомістких польових досліджень, оптимізуючи таким чином розподіл ресурсів на природоохоронні ініціативи. Ця ефективність має вирішальне значення для вирішення нагальних проблем, таких як деградація оселищ, надаючи дієві ідеї для політиків і природоохоронців. Масштабованість цієї технології покращує глобальні оцінки біорізноманіття, підтримуючи міжнародні угоди, такі як Конвенція про біологічне різноманіття. Вона сприяє зусиллям, спрямованим на пом'якшення антропогенного впливу, захищаючи пташине розмаїття для майбутніх поколінь.

1.6.2 Технологічне та освітнє використання

Глибинне навчання для розпізнавання видів птахів стимулює технологічні інновації, інтегруючи передові алгоритми у зручні для користувача інструменти, такі як додатки для смартфонів, такі як Merlin Bird ID. Ці платформи використовують згорткові нейронні мережі для ідентифікації видів за зображеннями або рекурентні нейронні мережі для аналізу вокалізації, наприклад, трелі дроздів, що робить їх більш доступними для дослідників та ентузіастів.

Технологія підтримує розробку автоматизованих систем моніторингу, таких як біоакустичні мережі або зйомка з дронів, які масштабують збір даних у різних середовищах існування. Ці досягнення покращують екологічний аналіз у режимі реального часу, сприяючи інтеграції з іншими екологічними інструментами на основі штучного інтелекту для комплексних досліджень біорізноманіття.

В освіті системи поглибленого навчання пропонують інтерактивні ресурси для студентів та викладачів, сприяючи залученню до екологічної науки. Додатки та онлайн-платформи дозволяють користувачам досліджувати різноманіття птахів, вивчати методи ідентифікації та робити свій внесок у громадську науку, як це видно на прикладі зростаючих наборів даних iNaturalist.

Ці інструменти сприяють підвищенню екологічної обізнаності, особливо серед молодшої аудиторії, через гейміфіковане навчання або шкільні програми. Демократизуючи доступ до розпізнавання видів, дослідження розвиває глобальну спільноту, яка зацікавлена у збереженні природи, об'єднуючи технології та освіту для підтримки сталих практик.

1.7 Мета і постановка задачі

Автоматичне розпізнавання видів птахів за допомогою глибинного навчання задовольняє гостру потребу в ефективних, масштабованих інструментах для моніторингу біорізноманіття птахів. Ця технологія має на меті покращити екологічні дослідження та охорону природи шляхом автоматизації ідентифікації видів, зменшуючи залежність від трудомістких людських спостережень [12]. Розроблена система використовуватиме нейронні мережі для аналізу візуальних та акустичних даних, забезпечуючи точну ідентифікацію різних видів та умов. Вона підтримуватиме дослідників, природоохоронців та громадських науковців, сприяючи

ширшому залученню до вирішення екологічних проблем, таких як втрата оселищ та зміна клімату.

Метою цієї роботи є розробка системи на основі глибинного навчання для автоматичного розпізнавання видів птахів. Система оброблятиме складні набори даних, наприклад, з фотопасток або біоакустичних реєстраторів, для отримання швидких і надійних результатів, що сприятиме екологічному моніторингу та збереженню різноманітних оселищ.

Ця система не замінить людський досвід, але доповнить його, надаючи масштабований інструмент для аналізу великих обсягів даних, наприклад, тих, що надаються такими платформами, як eBird або iNaturalist. Вона враховуватиме етичні міркування, забезпечуючи прозорість результатів роботи штучного інтелекту і сприяючи відповідальному використанню в дослідженнях і суспільних програмах, а також вирішуючи такі проблеми, як мінливість видів і шум навколишнього середовища.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- аналіз предметної галузі;
- вибір технологій для програмної реалізації;
- програмна реалізація;
- тестування;
- виявлення перспектив розробленого рішення.

2 АЛГОРИТМ ГЛИБИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ПТАХІВ

2.1 Основи обробки зображень для розпізнавання

Точна класифікація зображень птахів, що базується на детальному аналізі візуальних ознак, таких як візерунки оперення, форми тіла, кольорові градієнти, дрібні деталі дзьоба, очей чи навіть текстура пір'я в різних умовах освітлення, є центральним елементом функціонування веб-додатку для розпізнавання видів птахів.

Глибинне навчання, зокрема згорткові нейронні мережі, стало провідним підходом у цій сфері завдяки своїй унікальній здатності автоматично виявляти ієрархічні ознаки з необроблених даних, що дозволяє моделям адаптуватися до складних і різноманітних візуальних патернів, включаючи сезонні зміни в забарвленні чи пози птахів [16].

Цей процес розпочинається з операцій згортки, де спеціально розроблені фільтри сканують зображення, виділяючи базові елементи, такі як краї, текстури, градієнти кольорів чи контури, які можуть варіюватися залежно від фону чи природного оточення.

На подальших шарах ці прості патерни поступово комбінуються, утворюючи складніші структури, які відображають унікальні характеристики конкретних видів птахів, наприклад, відмінності в оперенні між близькоспорідненими видами, такими як різні види горобців чи голубів.

Такий підхід особливо цінний у задачах із великою кількістю класів, як-от ідентифікація сотень видів птахів, що мешкають у різних регіонах, де традиційне ручне визначення ознак стає не лише трудомістким, а й неефективним через суб'єктивність людського сприйняття та його обмеження в розпізнаванні тонких відмінностей.

Згорткові нейронні мережі вирішують цю проблему, використовуючи багат шарові архітектури, де кожен шар додає нові рівні абстракції – від

простих геометричних форм, таких як прямокутники чи кола, до високорівневих концепцій, таких як типові візуальні маркери певного виду, наприклад, характерні плями на крилах чи кільця навколо очей.

Ця гнучкість робить згорткові нейронні мережі ідеальним інструментом для автоматичної ідентифікації птахів, особливо коли зображення можуть варіюватися за якістю (наприклад, розмиті чи чіткі), кутом зйомки (згори, збоку чи знизу) чи умовами освітлення (день, сутінки, тінь).

Крім того, інтеграція згорткових нейронних мереж із великими наборами даних, такими як ImageNet, що містить мільйони зображень різних об'єктів, чи CUB-200-2011, спеціально розробленим для класифікації птахів, дозволяє моделям навчатися на широкому спектрі прикладів, включаючи різноманітні ракурси та природні сцени. Це підвищує їхню здатність узагальнювати знання й адаптуватися до нових даних, забезпечуючи високу точність у реальних умовах використання веб-додатку.

Важливим аспектом є також попередня обробка зображень, яка включає нормалізацію яскравості та контрасту, щоб усунути вплив зовнішніх факторів, і аугментацію, наприклад, обертання чи зміну масштабів, що імітує різні перспективи, що зустрічаються в польових умовах.

Такий підхід не лише покращує продуктивність моделей, але й робить їх стійкішими до помилок, викликаних шумом чи артефактами зображень, що є типовим для фотографій, зроблених аматорами в природному середовищі.

Усе це робить згорткові нейронні мережі ключовою технологією для створення надійного й адаптивного інструменту для орнітологів та ентузіастів, які прагнуть досліджувати біорізноманіття птахів через цифрові засоби.

2.2 Моделі для розпізнавання видів птахів

2.2.1 Модель ResNet

Архітектура ResNet, зокрема її версія ResNet50, розроблена для подолання проблем, пов'язаних із глибокими нейронними мережами, таких як зникнення градієнта під час навчання (рисунок 2.1).

Вона використовує залишкові зв'язки, які дозволяють сигналу проходити через мережу коротшими шляхами, зберігаючи важливу інформацію. ResNet50 складається з чотирьох етапів конволюційних шарів, де кількість каналів поступово зростає, а завершальним етапом є повноз'єднаний шар для класифікації.

Для розпізнавання птахів модель ResNet50 ефективно виділяє загальні риси, такі як контури тіла, форми крил, а також базові візуальні патерни, наприклад, основні градієнти оперення, завдяки чому її можна гнучко адаптувати до специфічних наборів даних.

Ця здатність до адаптації дозволяє моделі ефективно працювати з різноманітними умовами зйомки та варіаціями зовнішнього вигляду видів. Завдяки 25,6 мільйонам параметрів, які визначають її архітектурну складність, і приблизно 4,1 мільярда FLOP (Floating Point Operations), що відображають обчислювальну навантаженість, ResNet50 є оптимальним вибором для веб-розгортання на стандартному обладнанні, такому як середньопродуктивні сервери чи локальні комп'ютери. Її архітектура забезпечує баланс між точністю та швидкістю інференсу, що критично важливо для інтерактивних веб-додатків. Крім того, підтримка попередньо навчених ваг значно спрощує інтеграцію моделі у виробниче середовище.

Проте вона може мати труднощі з розпізнаванням видів із незначними відмінностями, наприклад, у кольорі оперення, через акцент на глобальних ознаках. Для підвищення її ефективності потрібне ретельне налаштування, великий обсяг даних і врахування локальних деталей.

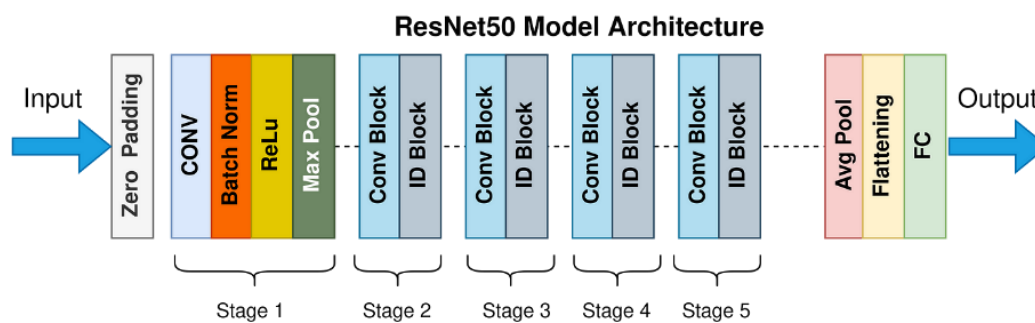


Рисунок 2.1 – Архітектура ResNet

2.2.2 Модель DenseNet

DenseNet, зокрема версія DenseNet201, вирізняється щільною архітектурою, де кожен шар з'єднується з усіма наступними, що сприяє повторному використанню ознак і зменшує кількість параметрів. Ця модель складається з щільних блоків (рисунок 2.2), де кожен шар отримує вхідні дані від усіх попередніх у межах блоку, що дозволяє ефективно обробляти складні деталі, такі як тонкі візерунки оперення чи текстури пір'я птахів. Завдяки такому підходу DenseNet201 оптимізує передачу інформації між шарами, що підвищує її здатність розпізнавати навіть незначні відмінності між видами.

Для розпізнавання птахів модель DenseNet201 демонструє виняткову ефективність у роботі з тонкими відмінностями, такими як текстура пір'я, форма дзьоба, візерунки на хвості чи навіть дрібні деталі навколо очей, що є ключовим фактором для точного розрізнення близькоспоріднених видів, які часто мають схожі зовнішні ознаки [17]. Завдяки приблизно 20 мільйонам параметрів, які визначають її структуру, і перехідним шарам, що регулюють розміри зображень та кількість каналів, модель оптимізує обчислювальні витрати, зберігаючи високу продуктивність при відносно помірних ресурсах.

Ці перехідні шари допомагають зменшувати просторові розміри даних на кожному етапі, що знижує навантаження на пам'ять і обчислювальну потужність, що є важливим для обробки великих наборів даних.

Однак її висока пам'ятна складність (близько 7,2 мільярда FLOP) і значне використання ресурсів під час навчання чи інференції можуть ускладнити розгортання на веб-серверах із обмеженими можливостями. Незважаючи на це, її здатність детально аналізувати зображення робить її цінним інструментом для складних задач класифікації.

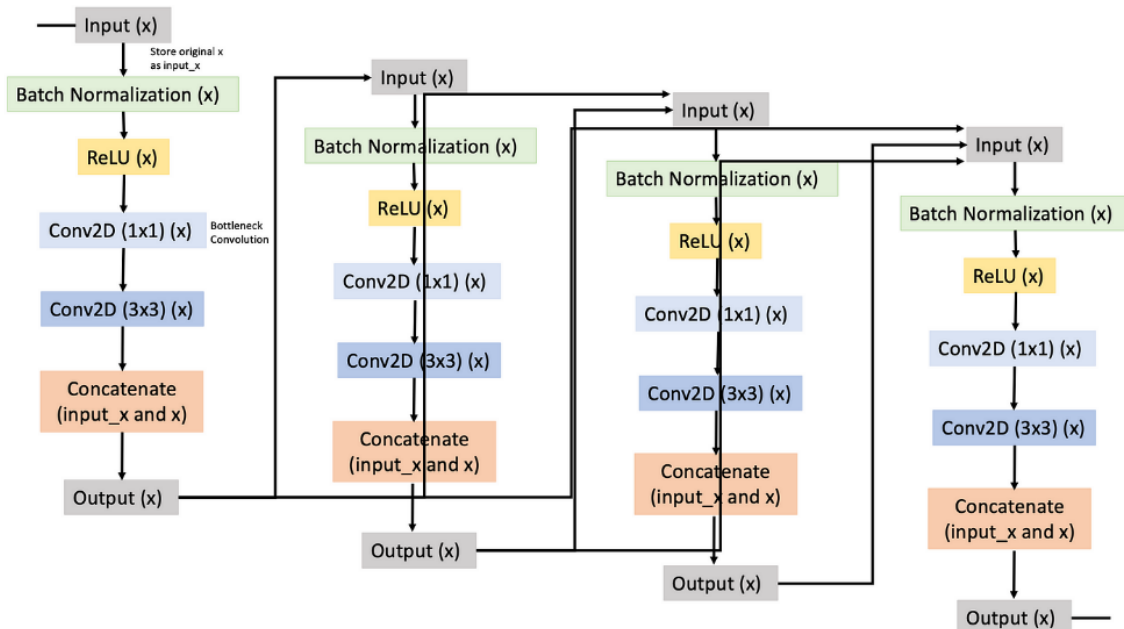


Рисунок 2.2 – Архітектура DenseNet

2.2.3 Модель InceptionV3

InceptionV3 вирізняється модульною структурою з блоками inception, які одночасно застосовують згортки різних масштабів, об'єднуючи їхні результати для кращого аналізу деталей (рисунок 2.3).

Ця модель включає допоміжні класифікатори на проміжних шарах, що стабілізують навчання глибоких мереж, а факторизовані згортки та

згортки 1×1 зменшують обчислювальні витрати, досягаючи приблизно 5,7 мільярда FLOP.

Для розпізнавання птахів InceptionV3 ефективно обробляє різноманітні візуальні патерни – від великих форм тіла до дрібних текстур оперення, завдяки своєму підходу до багатомасштабного аналізу [17]. Її помірна кількість параметрів полегшує розгортання на стандартних веб-серверах.

Проте складність архітектури може подовжувати час інференції, а на малих наборах даних можливе перенавчання, що потребує регуляризації, таких як відсіювання.

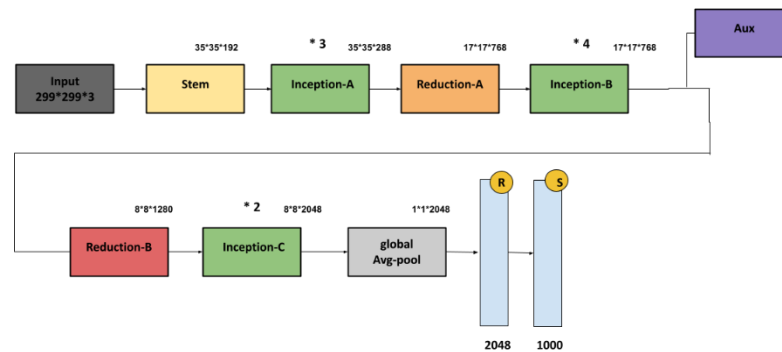


Рисунок 2.3 – Архітектура InceptionV3

2.2.4 Модель EfficientNet

EfficientNet, зокрема версія EfficientNet-B7, базується на систематичному масштабуванні глибини, ширини та роздільної здатності мережі для оптимального балансу продуктивності та ресурсів (рисунок 2.4).

Ця модель використовує інвертовані вузькі місця та блоки стиснення й збудження, що підвищують виразність ознак. З 66 мільйонами параметрів EfficientNet-B7 демонструє виняткову точність у детальних задачах, таких як розпізнавання птахів, фіксуючи як глобальні, так і локальні особливості, наприклад, градієнти оперення чи маркування обличчя [17]. Попередньо

навчені ваги з ImageNet дозволяють ефективно адаптуватися до специфічних наборів даних.

Однак висока обчислювальна вартість (понад 10 мільярдів FLOP) і тривалий час інференції вимагають потужного обладнання чи методів оптимізації, таких як квантування, що може бути викликом для веб-розгортання.

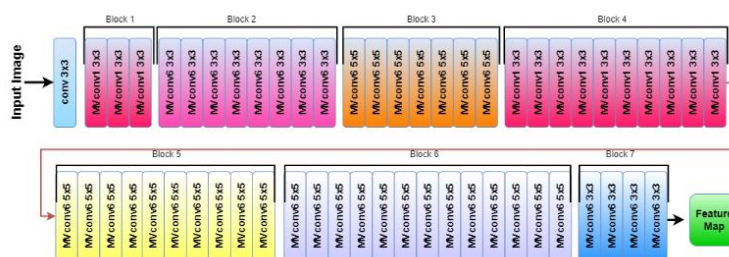


Рисунок 2.4 – Архітектура EfficientNet

2.3 Метод ансамблю

Ансамблеві методи об'єднують прогнози від декількох моделей для підвищення точності та надійності, використовуючи різні архітектури для охоплення ширшого спектру ознак [18]. Зменшуючи помилки, спричинені упередженнями окремих моделей, ансамблі особливо ефективні для дрібнозернистих задач, таких як розпізнавання видів птахів, де тонкі візуальні відмінності кидають виклик окремим моделям. До поширених методів ансамблювання належать:

- усереднення обчислює середнє значення результатів моделі, зазвичай після нормалізації softmax, для отримання остаточного прогнозу. Цей метод є простим і ефективним, коли моделі мають порівнянну продуктивність, зменшуючи дисперсію в прогнозах;

– зважене голосування призначає вагу моделям на основі їхньої точності, надаючи більший вплив сильнішим моделям. Потенційно більш точний, але вимагає налаштування ваг, що додає складності;

– стекінг використовує мета-навчальник для об'єднання прогнозів моделей, підвищуючи точність, але також збільшуючи обчислювальні витрати, що може бути недоцільним для веб-додатків.

Усереднення підходить для комбінації DenseNet201, EfficientNet-B7, InceptionV3 і ResNet50, оскільки їхні сильні сторони підвищують ефективність у складних випадках, таких як розрізнення видів зі схожим оперенням.

Ансамбль з DenseNet201, EfficientNet-B7, InceptionV3 та ResNet50 був обраний завдяки своїй вищій точності та надійності порівняно з окремими моделями. DenseNet201 чудово фіксує дрібні деталі, що є критично важливими для виявлення тонких відмінностей між видами. EfficientNet-B7 пропонує оптимізовану високу точність завдяки систематичному масштабуванню. InceptionV3 ефективно обробляє різномасштабні об'єкти, пристосовуючись до різноманітних візуальних патернів. ResNet50 забезпечує надійну базову лінію з ефективним виділенням ознак.

Окремі моделі стикаються з обмеженнями: ResNet50 може боротися з дуже схожими видами, EfficientNet-B7 вимагає значних ресурсів, DenseNet201 збільшує використання пам'яті, а InceptionV3 сповільнює висновок через складність. Ансамбль вирішує ці проблеми шляхом усереднення прогнозів, зменшуючи вплив слабких сторін кожної окремої моделі.

Було розглянуто альтернативні ансамблеві методи. Зважене голосування додає складності через налаштування ваги без суттєвого виграшу для збалансованого набору моделей. Стекування збільшує накладні витрати, що не підходить для веб-додатків з помірним трафіком.

Сумісність ансамблю з архітектурою веб-додатку додатково підтримує це рішення.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розробка веб-додатку для розпізнавання видів птахів передбачає поєднання технологій для управління наборами даних, навчання моделей, обробки зображень та надання веб-інтерфейсу. Додаток використовує Python як основну мову програмування, підтримувану різними бібліотеками та фреймворками для реалізації ансамблю глибинного навчання, обробки завантажених користувачем зображень, інтеграції зовнішніх даних та надання результатів через зручний інтерфейс.

3.1 Фреймворки для веб-додатків на Python

Створення веб-додатку для розпізнавання видів птахів потребує фреймворку, який забезпечує інтуїтивно зрозумілий інтерфейс для завантаження зображень і доступу до результатів, а також ефективно керує обробкою на стороні сервера прогнозів моделі глибинного навчання та інтеграцією зовнішніх наборів даних.

Python, як обрана мова програмування, надає ряд фреймворків для веб-додатків, кожен з яких має унікальні характеристики, пристосовані до різних розмірів і складності проектів. У цьому підрозділі оцінюються як легкі, так і повнофункціональні фреймворки Python, порівнюється їх відповідність потребам проекту та обґрунтовується вибір Flask як фреймворку для цього додатку [13].

Веб-фреймворки Python можна умовно поділити на легкі фреймворки та повностекові фреймворки. Легкі фреймворки, такі як Flask та FastAPI, надають перевагу простоті, гнучкості та мінімальним накладним витратам, дозволяючи розробникам створювати кастомні додатки, використовуючи лише необхідні компоненти. Ці фреймворки ідеально підходять для малих і середніх проектів зі специфічними вимогами, такими як обробка завантаження файлів і обслуговування динамічного контенту. Повностекові

фреймворки, такі як Django і Pyramid, надають комплексні інструменти, включаючи вбудовані функції для автентифікації користувачів, управління базами даних і шаблонування, що робить їх придатними для великомасштабних, складних додатків [14].

Для програми розпізнавання видів птахів основні вимоги включають зручний інтерфейс для завантаження зображень, обробку зображень на стороні сервера за допомогою ансамблю моделей глибокого навчання та інтеграцію зі статичним набором даних для перевірки присутності птахів в Україні. Додаток не потребує розширених функцій, таких як облікові записи користувачів, складна взаємодія з базами даних або розгалужені адміністративні панелі, що зменшує потребу у використанні повнофункціонального фреймворку. Тому оцінка зосереджується на фреймворках, які балансують між простотою, продуктивністю та легкістю інтеграції з робочими процесами глибокого навчання.

3.1.1 Легкі фреймворки

Flask – це мікрофреймворк, відомий своєю простотою використання та адаптивністю, що дозволяє розробникам створювати веб-додатки з мінімальною конфігурацією. Він полегшує використання RESTful API, завантаження файлів і рендеринг шаблонів, що робить його придатним для додатків, які потребують адаптованої внутрішньої логіки, наприклад, для обробки зображень з використанням моделей глибокого навчання. Легка конструкція Flask гарантує низьке використання ресурсів, що вигідно для розгортання додатків з помірним трафіком. Тим не менш, він не має інтегрованих функцій, таких як ORM або автентифікація, що вимагає подальшого налаштування для більш складних функцій.

FastAPI – це сучасний мікрофреймворк для асинхронного програмування, що вирізняється здатністю створювати високопродуктивні API. Він полегшує завантаження файлів і легко інтегрується з робочими

процесами машинного навчання завдяки своїм асинхронним функціям, які ефективно керують паралельними запитами.

Крім того, FastAPI автоматично генерує документацію API, що є перевагою для розробників. Тим не менш, асинхронна модель може створювати проблеми для тих, хто не дуже добре знайомий з асинхронним програмуванням, і вона порівняно новіша, ніж Flask, що призводить до меншої спільноти для вирішення спеціалізованих проблем.

3.1.2 Повностекові фреймворки

Django – це всеосяжний фреймворк з повним стеком, який пропонує комплексне рішення, включаючи такі функції, як ORM, адміністративний інтерфейс та систему автентифікації користувачів. Він особливо добре підходить для великомасштабних додатків, які мають складні внутрішні потреби, включаючи управління користувачами та динамічну генерацію контенту. Однак для простих додатків, таких як розпізнавання видів птахів, широкі можливості Django можуть спричинити зайві накладні витрати, що призведе до довшого часу встановлення та більшого споживання ресурсів. Крім того, синхронна архітектура Django може обмежувати продуктивність при виконанні операцій, що вимагають великих обчислень, таких як виведення моделі.

Pyramid – це універсальний повностековий фреймворк, який забезпечує баланс між простотою використання та масштабованістю. Він дозволяє розробникам обирати компоненти відповідно до своїх потреб, в результаті чого фреймворк є легшим за Django, але більш потужним, ніж Flask. Pyramid ідеально підходить для проектів, які передбачають зростання з часом, однак, його більш складна крива навчання і необхідність додаткового налаштування можуть зробити його менш привабливим для простих додатків з базовими вимогами.

3.1.3 Вибір фреймворку

Після порівняння фреймворків було обрано Flask як найбільш підходящий вибір для веб-додатку з розпізнавання видів птахів [15]. Його легка архітектура відповідає потребам проекту в простому інтерфейсі, який обробляє завантаження зображень і відображає результати, не вимагаючи складних функцій, таких як автентифікація користувачів або управління базами даних. Гнучкість Flask дозволила безперешкодно інтегруватися з ансамблем моделей глибинного навчання, що дало змогу користувацькій внутрішній логіці ефективно обробляти зображення та запитувати набір даних про птахів України. Мінімальні накладні витрати фреймворку забезпечують ефективне використання ресурсів, що є критично важливим для додатків з помірним очікуваним трафіком та обчислювально інтенсивними завданнями, такими як виведення моделей.

Хоча асинхронні можливості FastAPI можуть підвищити продуктивність для сценаріїв з високим рівнем паралелізму, очікуваний трафік від низького до помірного не виправдовує додаткової складності асинхронного програмування. Повностекові фреймворки, такі як Django та Pyramid, були визнані надто складними, оскільки їх вбудовані функції виходять за рамки проекту, що потенційно сповільнює розробку та розгортання. На користь Flask також свідчили сформована спільнота, обширна документація та знайомство з ним серед розробників, що забезпечило швидку розробку та надійну підтримку для усунення несправностей.

3.2 Стратегії розгортання моделей глибинного навчання

Інтеграція набору моделей глибинного навчання у веб-додаток на базі Python для розпізнавання видів птахів вимагає стратегії розгортання, яка

забезпечує ефективну класифікацію зображень, зберігаючи при цьому швидкість реагування користувача.

Значні вимоги ансамблю до обчислювальних потужностей та пам'яті, з приблизно 135 мільйонами параметрів та високими обчислювальними витратами вимагають детальної оцінки методів розгортання. Треба проаналізувати підходи до хостингу та обслуговування моделей, зосереджуючись на балансі простоти, продуктивності, масштабованості та ефективності використання ресурсів для додатка з низьким або помірним трафіком [19].

3.2.1 Пряме завантаження

Пряме завантаження вбудовує набір моделей у процес веб-додатку, дозволяючи одному серверу обробляти взаємодії користувачів, такі як завантаження зображень і відображення результатів, поряд з виведенням моделей. При запуску додатку всі чотири моделі завантажуються в пам'ять, де вони залишаються протягом усього часу роботи сервера. Коли користувач завантажує зображення птаха, сервер попередньо обробляє його, послідовно або паралельно пропускає його через кожну модель залежно від реалізації та усереднює їхні результати softmax, щоб отримати остаточний прогноз. Цей прогноз, разом з метаданими, такими як наявність птаха в Україні, потім відображається користувачеві.

Цей підхід є простим і вимагає мінімальної інфраструктури. Єдина веб-платформа на базі Python, така як Flask, управляє як користувацьким інтерфейсом, так і виведенням моделі в рамках одного процесу, усуваючи необхідність у додаткових серверах, протоколах зв'язку або складних механізмах синхронізації. Така простота прискорює розробку, оскільки розробники можуть інтегрувати моделі безпосередньо в логіку додатка без управління зовнішніми залежностями. Налаштування також стає простішим, оскільки всі компоненти – завантаження зображень, попередня

обробка, інференція та візуалізація результатів – працюють в одному середовищі, що спрощує відстеження та усунення помилок [20].

Низький та помірний трафік, який очікується для додатку, також підтримує пряме завантаження. З обмеженою базою користувачів, сервер обробляє запити послідовно без значного зниження продуктивності. Це усуває необхідність у складному управлінні паралелізмом, зменшуючи накладні витрати на реалізацію. Відсутність мережевого зв'язку забезпечує стабільний час відгуку, оскільки всі обчислення відбуваються локально в межах серверного процесу. Це особливо вигідно для сценаріїв з низьким трафіком, де накладні витрати на міжсерверний зв'язок були б непропорційні робочому навантаженню.

Пряме завантаження також сприяє швидкому створенню прототипів і розгортанню, що є критично важливим для академічних або невеликих проектів з обмеженими ресурсами. Під час розробки цей підхід дозволив зосередитися на основних функціональних можливостях, таких як інтеграція моделей і точність результатів, не відволікаючись на управління інфраструктурою. Простота обслуговування одного сервера зменшує операційні витрати та необхідний технічний досвід, що робить його ідеальним для даного проекту.

Однак, пряме завантаження має значні проблеми через ресурсоємність ансамблю. Завантаження всіх чотирьох моделей в пам'ять під час запуску споживає значний обсяг оперативної пам'яті, особливо через щільне підключення DenseNet201 та велику кількість параметрів EfficientNet-B7. На сервері з 16 ГБ оперативної пам'яті використання пам'яті може наблизитися до граничного, що може призвести до уповільнення або збоїв, якщо інші процеси конкурують за ресурси. Для вирішення цієї проблеми рекомендуються сервери з 32 ГБ оперативної пам'яті або вище, а також ретельний моніторинг розподілу пам'яті під час розгортання.

Послідовна обробка HTTP-запитів і виведення моделі в рамках одного процесу може призвести до затримок, особливо якщо кілька користувачів

надсилають зображення одночасно. Наприклад, при одночасному завантаженні двох зображень другий користувач може чекати на 4-10 секунд довше, залежно від навантаження на сервер і можливостей процесора. Хоча це малоімовірно, враховуючи низький трафік, це вказує на обмеження масштабованості. Асинхронна обробка запитів або робочі потоки могли б пом'якшити цю проблему, але вони вносять складність, зводячи нанівець простоту прямого завантаження.

Час виведення, що в середньому становить 2-5 секунд на одне зображення на ЦП, може вплинути на якість роботи користувача, якщо очікування змістяться в бік продуктивності в реальному часі. Обчислювальна вартість ансамблю, особливо для EfficientNet-B7, зумовлює цю затримку. Методи оптимізації, такі як квантування моделі, обрізання або пакетна обробка, можуть зменшити вимоги до пам'яті та обчислювальних ресурсів.

Іншим недоліком є відсутність модульності. Оскільки моделі тісно пов'язані з веб-додатком, їх оновлення або заміна вимагає перезапуску сервера, що призводить до короткого простою.

3.2.2 Окремий сервер

Стратегія окремого сервера розгортає ансамбль на виділеній машині або процесі, окремо від веб-сервера, який обробляє взаємодію з користувачем. Веб-сервер, відповідальний за такі завдання, як завантаження зображень і рендеринг результатів, взаємодіє з сервером моделей через API. Коли користувач завантажує зображення птаха, веб-сервер надсилає його на сервер моделей, який попередньо обробляє зображення, виконує висновок за допомогою ансамблю, усереднює прогнози і повертає результат для відображення. Така архітектура дає змогу самостійно оптимізувати роботу сервера моделей, потенційно використовуючи спеціалізоване обладнання, наприклад, графічні процесори.

Цей підхід пропонує значні переваги масштабованості, що робить його придатним для додатків з високим трафіком. Ізолюючи виведення моделі, веб-сервер залишається швидкодіючим, ефективно обробляючи завдання користувацького інтерфейсу, такі як заповнення форм і рендеринг сторінок.

Сервер моделей може обробляти декілька запитів одночасно, особливо якщо він оснащений графічним процесором, що прискорює виведення для обчислювально інтенсивних моделей, таких як EfficientNet-B7. Розподіл ресурсів оптимізовано, оскільки вимоги до пам'яті та обчислювальних ресурсів модельного сервера відокремлені від легших завдань веб-сервера. Така ізоляція запобігає боротьбі за ресурси, забезпечуючи стабільну роботу.

Балансування навантаження ще більше підвищує масштабованість, дозволяючи декільком моделям серверів розподіляти трафік у сценаріях з високим попитом, що є цінною функцією для додатків, які очікують на зростання.

Підхід з окремим сервером також дозволяє спеціалізувати апаратне забезпечення. Сервер моделей, оснащений GPU, може використовувати паралельну обробку, що значно скорочує час виведення 135 мільйонів параметрів ансамблю. Це особливо ефективно для EfficientNet-B7, де прискорення на графічних процесорах пом'якшує його 37 мільярдів FLOP. Навіть без графічних процесорів високопродуктивний процесорний сервер, виділений для виведення, може перевершити загальний процес, оскільки він уникає конкуренції із завданнями веб-сервера.

Основний недолік – підвищена складність. Впровадження окремого сервера вимагає розробки надійного API для зв'язку, визначення форматів даних та забезпечення обробки помилок у разі збоїв у мережі або таймаутів. Синхронізація між серверами додає накладних витрат, оскільки веб-сервер повинен чекати відповідей від серверів моделей, перш ніж виводити

результати. Це вимагає ретельного проектування для підтримки надійності, що збільшує час розробки та тестування.

Виникають міркування безпеки, оскільки міжсерверна комунікація через API вимагає таких засобів захисту, як автентифікація та шифрування, щоб запобігти несанкціонованому доступу або витоку даних. Реалізація цих заходів додає додаткової складності, відволікаючи зусилля від основної розробки програми. Для додатків з низьким трафіком ризики безпеки мінімальні, але додатковий тягар реалізації залишається.

3.2.3 Вибір стратегії розгортання моделей

Для інтеграції ансамблю моделей у веб-додаток було обрано пряме завантаження. Його простота зводить до мінімуму зусилля з розробки, налаштування та обслуговування, що ідеально підходить для проекту з низьким трафіком. Оптимізована архітектура підходу дозволила зосередитися на критично важливих функціях, таких як точні прогнози та зручне відображення результатів, не відволікаючись на інфраструктуру.

Стратегія окремого сервера, пропонуючи масштабованість і потенційний приріст продуктивності, створює надмірну складність для сфери застосування додатку. Мережеві затримки, розробка API та витрати на інфраструктуру переважають переваги, коли пряме завантаження відповідає вимогам продуктивності. Навіть при використанні графічного процесора незначне скорочення часу виведення не виправдовує додаткових зусиль для додатків з низьким трафіком. Якщо трафік збільшиться, можна розглянути такі оптимізації, як асинхронна обробка або гібридний підхід, але вони виходять за рамки потреб поточного проекту.

Вибір прямого завантаження відповідає пріоритетам простоти, практичності та ефективності, забезпечуючи безперешкодну інтеграцію ансамблю моделей з архітектурою веб-додатку. Він підтримує надійну обробку завантажених користувачем зображень і точне надання результатів

розпізнавання видів птахів, що відповідає цілям проекту, зберігаючи при цьому просту і зручну в обслуговуванні систему.

3.3 Підходи до обробки та інтеграції даних

Веб-додаток для розпізнавання видів птахів залежить від точної обробки завантажених користувачем зображень птахів та ефективної інтеграції зовнішніх даних для надання точних результатів класифікації та контекстної інформації, наприклад, чи присутній птах в Україні. Ансамбль моделей глибинного навчання вимагає стандартизованих вхідних зображень для забезпечення надійних прогнозів, тоді як невеликий набір даних про птахів України, що використовується для перевірки статусу виду, потребує швидкого доступу для запитів у режимі реального часу.

3.3.1 Підходи до обробки зображень

Підготовка завантажених користувачем фотографій птахів для ансамблю моделей глибинного навчання включає такі етапи попередньої обробки, як зміна розміру, нормалізація та додаткове доповнення даних, щоб відповідати умовам навчання набору даних. Ці кроки стандартизують зображення, які відрізняються за роздільною здатністю, форматом або якістю, забезпечуючи точну класифікацію.

Обробка на стороні сервера виконує всі завдання попередньої обробки на веб-сервері після того, як користувач завантажив зображення. Отримавши фотографію, сервер застосовує перетворення: змінює розмір, нормалізує значення пікселів до середнього значення і стандартного відхилення, відповідно до попередньої обробки ImageNet. Потім попередньо оброблене зображення подається на ансамбль моделей для виведення, а результати, включаючи назву виду та статус України, повертаються користувачеві [21].

Обробка на стороні сервера забезпечує повний контроль над попередньою обробкою, гарантуючи узгодженість з конвеєром навчання моделі. Відтворюючи точні перетворення, що використовуються під час навчання, сервер мінімізує розбіжності, які можуть призвести до помилок у прогнозуванні. Цей підхід усуває варіативність клієнтського середовища, таку як відмінності в реалізації браузерів, операційних системах або апаратному забезпеченні пристроїв, забезпечуючи однаковість вхідних даних для всіх користувачів.

Однак, обробка на стороні сервера збільшує навантаження на сервер, оскільки всі обчислення відбуваються на тій самій машині, яка обробляє висновки моделі та запити користувачів. Для моделей, що вимагають багато пам'яті, додавання завдань попередньої обробки, хоча і є легким, але сприяє використанню ресурсів. Це можна контролювати при низькому трафіку, але може призвести до перевантаження ресурсів при високому рівні паралелізму.

Обробка на стороні клієнта переносить попередню обробку на пристрій користувача, зазвичай у браузері за допомогою JavaScript або WebAssembly. Перед завантаженням клієнт змінює розмір зображення, нормалізує значення пікселів і за потреби застосовує аугментацію. Попередньо оброблене зображення, часто перетворене в компактний тензорний формат, надсилається на сервер для висновків, що зменшує обсяг обчислень на стороні сервера.

Обробка на стороні клієнта зменшує навантаження на сервер, вивантажуючи попередню обробку на пристрої користувача. Зміна розміру та нормалізація, що виконуються в браузерах або на мобільних пристроях, звільняють ресурси сервера для виведення ансамблю, потенційно покращуючи масштабованість в умовах високого трафіку. Для моделей, що потребують багато пам'яті, це дозволяє серверу зосередитися на завданнях прогнозування, оптимізуючи розподіл ресурсів на обладнанні.

Однак, обробка на стороні клієнта ставить під загрозу узгодженість, оскільки попередня обробка залежить від пристрою та браузера користувача. Відмінності в реалізації JavaScript, версіях браузерів або апаратних можливостях можуть призвести до відмінностей у зміні розміру зображень або нормалізації значень, що відхиляються від конвеєра навчання.

Продуктивність сильно відрізняється на різних пристроях. Недорогі смартфони або старі браузери можуть мати проблеми з попередньою обробкою, що призводить до затримок або помилок. Наприклад, зміна розміру зображення з високою роздільною здатністю на бюджетному пристрої може зайняти секунди, що розчаровує користувачів.

Обробка на стороні сервера була обрана через її здатність забезпечувати узгодженість даних, що є критично важливим для точності ансамблю. Централізуючи попередню обробку на сервері на базі Flask, програма гарантує, що завантажені користувачем зображення відповідають умовам навчання, мінімізуючи помилки прогнозування. Легка природа завдань препроцесорної обробки в поєднанні з низьким очікуваним трафіком забезпечує прийнятну продуктивність на сучасному обладнанні.

Обробка на стороні клієнта, зменшуючи навантаження на сервер і пропускну здатність, вносить варіативність, яка може вплинути на точність, особливо для дрібнозернистої класифікації. Складність забезпечення узгодженої попередньої обробки на різних пристроях і браузерах переважає її переваги для додатків з низьким трафіком. Простота, надійність та узгодженість з фокусом проекту на точному розпізнаванні птахів роблять обробку на стороні сервера оптимальним вибором.

3.3.2 Методи інтеграції зовнішніх наборів даних

База даних птахів України – невелика статична колекція – використовується для визначення наявності в Україні визнаних видів птахів

та їхнього природоохоронного статусу. З огляду на компактний розмір, ефективний доступ до бази даних є важливим для запитів у режимі реального часу під час відображення результатів.

Завантаження файлів передбачає зчитування набору даних про птахів України з файлу на диску кожного разу, коли робиться запит. Після того, як модельний ансамбль прогнозує вид, сервер відкриває файл, шукає загальну назву виду та отримує його статус і наукову назву для відображення.

Цей метод простий у реалізації і вимагає мінімальних налаштувань. Сервер на основі Python, такий як Flask, може використовувати стандартні операції файлового вводу/виводу для читання CSV-файлу, розбираючи його на словник або список для пошуку. Для невеликого набору даних читання файлів є простим і потребує лише базової обробки помилок, пов'язаних з доступом до файлів. Це відповідає меті проекту – мінімізувати складність, оскільки не потрібно ніякої додаткової інфраструктури або залежностей.

Зберігання бази даних передбачає завантаження набору даних про птахів України в реляційну або NoSQL базу даних. Після прогнозування за допомогою моделі сервер запитує в базі даних статус виду та його наукову назву, використовуючи ідентифікатор, наприклад, загальну назву.

Бази даних забезпечують ефективні запити, особливо для структурованих даних. Легка база даних, така як SQLite, підходить для невеликих наборів даних, підтримує швидкий пошук і паралельний доступ, що покращує масштабованість. Для набору даних по Україні проста таблиця зі стовпчиками для загальної назви, наукової назви та статусу дозволяє швидко здійснювати пошук, збільшуючи час відгуку порівняно із завантаженням файлу.

Цілісність даних покращується, оскільки бази даних забезпечують дотримання схем, перевіряють вхідні дані та коректно обробляють помилки. Наприклад, відсутні або неправильно оформлені записи відловлюються під час завантаження даних, що зменшує проблеми під час виконання. Це

гарантує надійні результати при відображенні статусу видів, що узгоджується з фокусом проекту на точності.

Масштабованість є високою, оскільки бази даних ефективно обробляють паралельні запити, підтримуючи потенційне зростання трафіку. SQLite, з мінімальним налаштуванням, підходить для додатків з низьким трафіком, тоді як PostgreSQL або MongoDB можуть масштабуватися до більших навантажень, якщо це необхідно. Така гнучкість робить сховище баз даних перспективним, дозволяючи зберігати більші набори даних або додаткові функції, наприклад, дані, надані користувачами.

Зберігання в пам'яті завантажує набір даних про птахів України в оперативну пам'ять сервера під час запуску програми, як правило, у вигляді словника або хеш-таблиці. Запити виконуються шляхом пошуку загальної назви виду в пам'яті, миттєвого отримання його статусу та наукової назви.

Обслуговування є простим, оскільки оновлення передбачають модифікацію вихідного файлу і перезапуск сервера. Для статичного набору даних оновлення відбуваються рідко, що робить цей підхід практичним.

Цей метод пропонує неперевершену швидкість, оскільки доступ до пам'яті на порядки швидший за дисковий ввід/вивід або запити до бази даних. Для невеликого набору даних пошук займає мікросекунди. Це забезпечує швидке надання результатів, покращуючи користувацький досвід при відображенні статусу виду поряд з прогнозами.

Зберігання в пам'яті було обрано для інтеграції набору даних про птахів України завдяки його швидкості та простоті. Невеликий розмір набору даних займає незначний обсяг пам'яті, комфортно розміщуючись в оперативній пам'яті сервера разом з ансамблем моделей.

Завантаження файлів, хоч і є простим, але призводить до затримки дискового вводу/виводу, дещо затримуючи відповіді та обмежуючи масштабованість. Зберігання в базі даних забезпечує надійність запитів, але додає зайвої складності для невеликого статичного набору даних. Продуктивність сховища в пам'яті та простота інтеграції роблять його

оптимальним вибором, оскільки воно підтримує ефективні запити і відповідає меті точного і зручного розпізнавання видів птахів.

3.4 Технології бекенду

Бекенд керує критично важливими функціями, такими як обробка наборів даних, навчання та виведення моделей, попередня обробка зображень та інтеграція зовнішніх даних, покладаючись на бібліотеки та фреймворки на основі Python, оптимізовані для глибинного навчання та веб-сервісів.

3.4.1 Фреймворк Flask

Flask, легкий веб-фреймворк Python, забезпечує роботу серверної частини програми, пропонуючи мінімалістичну, але гнучку платформу для обробки HTTP-запитів, обслуговування користувацького інтерфейсу та надання прогнозів моделі. Його мікрофреймворк забезпечує тонкий контроль над компонентами програми, уникаючи накладних витрат більш комплексних фреймворків. Flask обробляє маршрутизацію для домашньої сторінки та кінцевих точок прогнозування, обробляє завантажені користувачем зображення та повертає результати у вигляді JSON-відповідей, які включають ідентифікацію видів та контекстні дані, такі як регіональна присутність.

Його простота полегшує інтеграцію з моделями глибинного навчання, дозволяючи пряме завантаження в серверний процес для ефективного виведення.

Підтримка Flask швидкої розробки API та сумісність з екосистемою Python забезпечує безперешкодну взаємодію з бібліотеками обробки зображень та виведення моделей.

Фреймворк не вимагає багато ресурсів і відповідає очікуванням проекту з низьким трафіком, що робить його оптимальним вибором для простого та надійного бекенду.

3.4.2 Бібліотеки PyTorch та torchvision

PyTorch, потужний фреймворк для глибокого навчання, керує навчанням і розгортанням ансамблю моделей, що використовуються для класифікації видів птахів. Його динамічний обчислювальний граф та інтуїтивно зрозумілий інтерфейс дозволяють гнучко конфігурувати, налаштовувати та робити висновки, що є критично важливим для адаптації попередньо навчених моделей до спеціалізованих наборів даних. PyTorch підтримує прискорення на графічних процесорах для швидшого навчання, хоча для потреб розгортання програми достатньо виводу на базі процесора.

Супутня бібліотека torchvision надає попередньо навчені моделі, утиліти для попередньої обробки зображень і завантажувачі наборів даних, що спрощує процес розробки. Вона надає архітектури моделей і стандартні перетворення, такі як зміна розміру і нормалізація, забезпечуючи узгодженість між навчанням і висновками. Разом PyTorch і torchvision дозволяють ефективно керувати обчислювально інтенсивним ансамблем, підтримуючи високоточну класифікацію, зберігаючи при цьому ефективність розробки.

3.4.3 Бібліотека Pillow

Pillow, бібліотека зображень Python Imaging Library, обробляє завантаження та маніпуляції з фотографіями, завантаженими користувачем. Вона підтримує різні формати зображень, забезпечуючи сумісність з різноманітними вхідними даними, і виконує завдання початкової обробки, такі як перетворення зображень у формат RGB перед подальшою

попередньою обробкою. Легка конструкція Pillow і простота інтеграції з веб-фреймворками роблять його ідеальним інструментом для обробки зображень на стороні сервера, що сприяє підвищенню надійності та продуктивності програми.

Його роль у підготовці зображень для введення в модель відповідає потребі проекту в послідовній попередній обробці, гарантуючи, що завантажені фотографії будуть правильно відформатовані для точних прогнозів.

3.4.4 Бібліотека NumPy

NumPy, базова бібліотека для чисельних обчислень, підтримує ефективні операції з масивами, необхідні для обробки результатів моделювання. Вона агрегує прогнози з ансамблю моделей, обчислюючи усереднені ймовірності для визначення остаточної класифікації видів та оцінки достовірності. Продуктивність NumPy для чисельних задач забезпечує швидку обробку даних прогнозування, підтримуючи швидку реакцію API.

Його сумісність з фреймворками глибинного навчання та здатність обробляти великі масиви роблять його критично важливим для логіки прогнозування ансамблю. Широке застосування NumPy в наукових обчисленнях гарантує надійну обробку числових даних, підтримуючи вимоги проекту до точності та ефективності.

3.5 Технології фронтенду

Фронтенд забезпечує користувачам зручний і інтуїтивно зрозумілий користувацький інтерфейс (UI) для завантаження зображень і перегляду результатів класифікації, використовуючи стандартні веб-технології для сумісності з різними пристроями та простоти використання. Інтерфейс має

адаптивний дизайн, створений із Tailwind CSS, що гарантує комфортну роботу як на мобільних телефонах, так і на комп'ютерах, із швидким завантаженням сторінок навіть на повільних з'єднаннях.

HyperText Markup Language (HTML) формує структурну основу, організуючи поля для завантаження, кнопки та секції результатів у логічний макет із семантичними тегами для підвищення доступності. Tailwind CSS, підключений через CDN, забезпечує сучасну стилізацію елементів, таких як форми й результати, уникаючи складних процесів збірки.

JavaScript додає інтерактивність, обробляючи завантаження зображень, асинхронні запити до бекенду та динамічне оновлення результатів без перезавантаження сторінки, створюючи плавний користувацький досвід. Ця комбінація технологій підтримує мету проекту – легкий, ефективний і доступний інтерфейс для розпізнавання птахів.

4 ДЕМОНСТРАЦІЯ РОБОТИ ЗАСТОСУНКУ

Після запуску веб-додатку користувач потрапляє на головну сторінку, яка пропонує зручний та інтуїтивно зрозумілий інтерфейс для розпізнавання видів птахів. Центральним елементом сторінки є вікно завантаження зображень, що дозволяє легко додати фотографію птаха для подальшої обробки та класифікації (рисунок 4.1).

Вікно завантаження підтримує популярні формати зображень, такі як JPEG і PNG, що робить застосунок доступним для широкого кола користувачів.

Після вибору файлу зображення автоматично передається на сервер для обробки ансамблем моделей глибокого навчання, забезпечуючи швидкий перехід до етапу аналізу без додаткових дій із боку користувача.

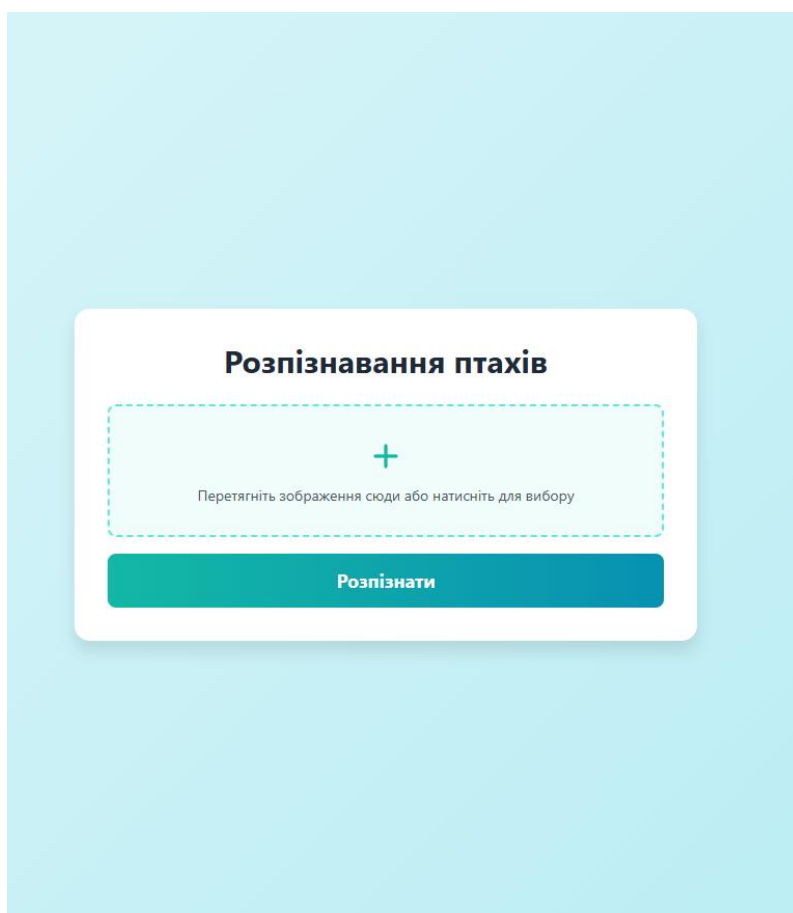


Рисунок 4.1 – Головна сторінка

Після вибору та завантаження фотографії птаха на головній сторінці веб-додаток оперативно обробляє зображення та презентує користувачу результати розпізнавання, які відображаються у зрозумілій і структурованій формі (рисунок 4.2). Результати представлені у компактному форматі, що включає ключову інформацію.

Відображення результатів реалізовано за допомогою JavaScript та Tailwind CSS, що забезпечує динамічне оновлення сторінки без перезавантаження. Інформація презентована у візуально привабливому форматі з чітким виділенням ключових даних, що сприяє зручності сприйняття. Повідомлення про присутність птаха в Україні та його статус акцентовані для швидкого ознайомлення.

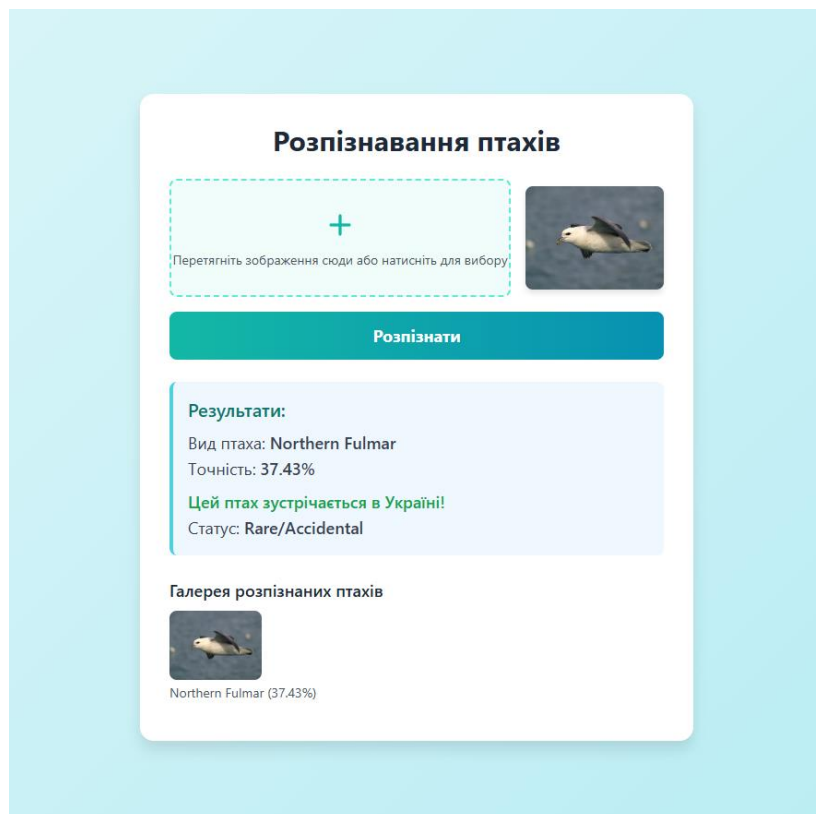


Рисунок 4.2 – Результати розпізнавання

Веб-додаток пропонує користувачам функцію перегляду історії їхніх розпізнавань, реалізовану через галерею розпізнаних птахів. Ця можливість

дозволяє відкрити спеціальний розділ, де відображаються всі зображення, оброблені під час поточної сесії, разом із результатами їхньої класифікації (рисунок 4.3).

Завдяки JavaScript галерея підтримує динамічне відображення даних, дозволяючи користувачу легко переглядати історію сесії через прокручування або кліки по елементах.

Ця функція зберігає результати розпізнавань у тимчасовій пам'яті браузера протягом сесії, що оптимізує ресурси сервера та забезпечує швидкий доступ до даних без додаткових запитів до бекенду. Для користувачів це означає зручний спосіб аналізу кількох зображень або повернення до попередніх прогнозів для подальшого вивчення.

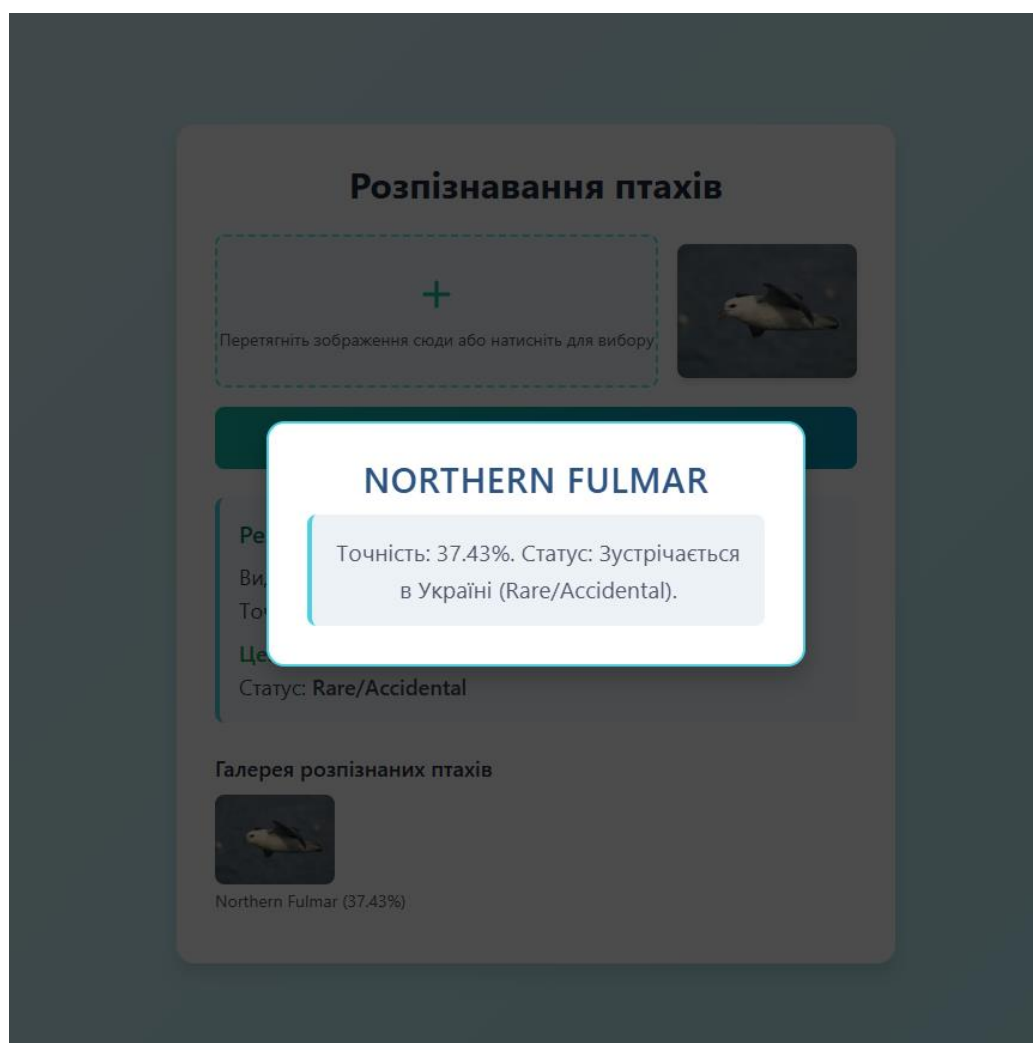


Рисунок 4.3 – Історія розпізнавань

Ця функція робить веб-додаток цінним інструментом для тих, хто прагне систематизувати свої спостереження за птахами, зберігаючи при цьому простоту взаємодії та високу функціональність.

У випадках, коли розпізнаний птах не зареєстрований у базі даних птахів України, застосунок інформує користувача про це. Це дозволяє краще зрозуміти, чи є спостереження аномальним або ймовірно зафіксовано мігранта чи рідкісний вид.

Така функціональність підвищує інформативність результатів та актуальність для орнітологічних досліджень (рисунок 4.4).



Рисунок 4.4 – Розпізнавання птаха

У деяких випадках, особливо коли птахи мають дуже схожі візуальні ознаки, модель може демонструвати нижчу точність. В такому випадку розпізнавання супроводжується нижчим показником впевненості (рисунок 4.5). Це підкреслює важливість подальшого вдосконалення моделі, зокрема шляхом додавання контекстної інформації або мультимодальних даних.

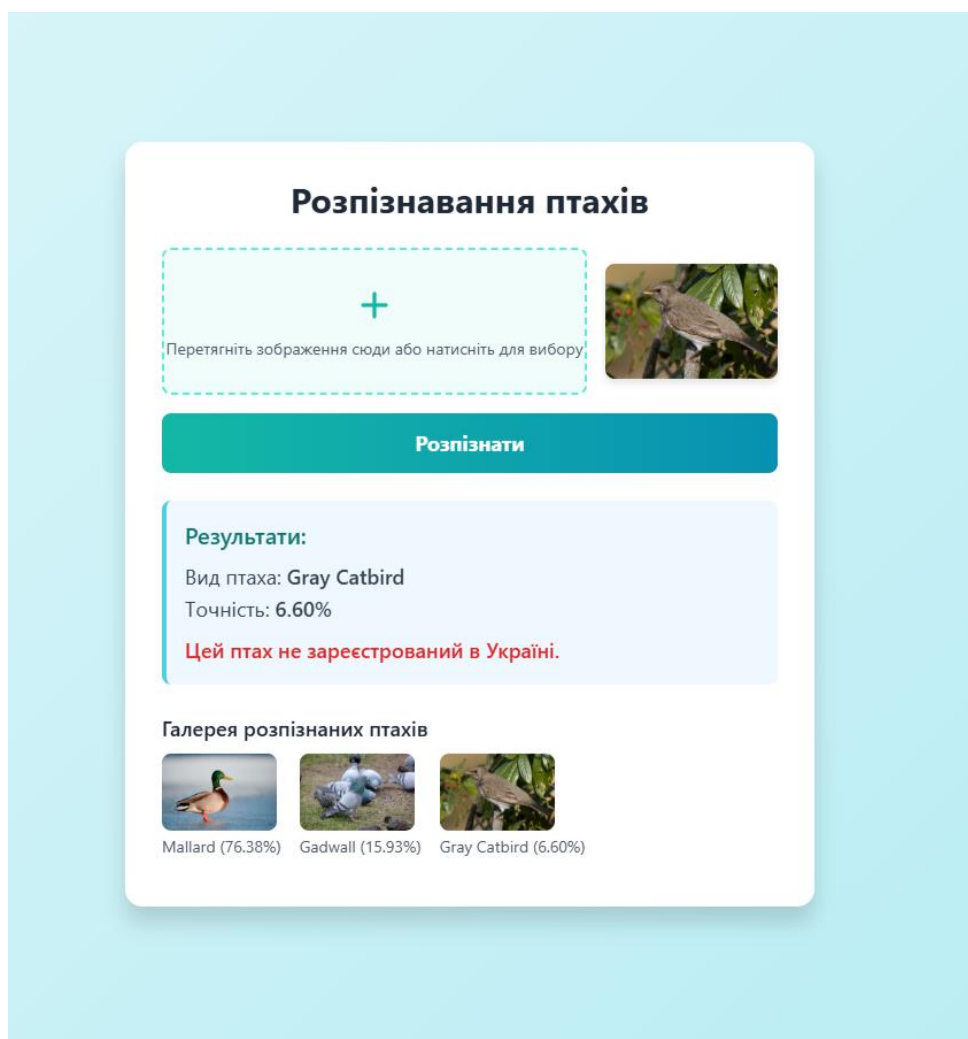


Рисунок 4.5 – Розпізнавання птаха

Таким чином, результати роботи веб-додатку демонструють як його ефективність, так і наявні виклики, пов'язані з розпізнаванням подібних видів. Надалі це відкриває перспективи для вдосконалення системи та підвищення точності класифікації.

ВИСНОВКИ

Основною метою роботи було створення веб-додатку для автоматичного розпізнавання видів птахів за зображеннями з використанням глибинного навчання, а також попереднє дослідження технологій і методів, найбільш придатних для реалізації цього завдання.

У процесі розробки сформовано структуру для зберігання та обробки даних, включаючи набір даних CUB-200-2011 і базу даних птахів України, яка надає додаткову контекстну інформацію про поширення видів у регіоні. Під час підготовки даних було встановлено, що нормалізація зображень із параметрами ImageNet є критично важливою для узгодженості між етапами навчання та інференсу, що суттєво впливає на якість прогнозів.

На основі обраних технологій реалізовано бекенд застосунку з використанням мови програмування Python, фреймворку Flask, бібліотек PyTorch, torchvision, Pillow і NumPy. Для створення користувацького інтерфейсу застосовано HTML, Tailwind CSS і JavaScript, що забезпечило інтуїтивну та адаптивну взаємодію з користувачами. Застосунок демонструє високу функціональність, дозволяючи завантажувати зображення, отримувати результати розпізнавання з точністю прогнозу та переглядати історію класифікацій у галереї.

Перспективи розвитку веб-додатку для розпізнавання видів птахів є значними, особливо в контексті екологічного моніторингу та популяризації орнітології. Застосунок може стати цінним інструментом для аматорів і професійних дослідників, сприяючи збереженню біорізноманіття шляхом швидкої ідентифікації видів. Потенційні напрями вдосконалення включають підвищення точності розпізнавання для видів зі схожими візуальними ознаками шляхом інтеграції додаткових даних, таких як геолокація чи поведінкові характеристики.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What we do. *BirdLife International*. URL: <https://www.birdlife.org/what-we-do/species/> (дата звернення 10.04.2025)
2. Ebird. *eBird*. URL: <https://ebird.org/home> (дата звернення 10.05.2025)
3. All About Birds. *Cornell Lab of Ornithology*. URL: <https://www.allaboutbirds.org/news/> (дата звернення 11.04.2025)
4. Automated Bird Identification Systems. *National Geographic*. URL: <https://www.nationalgeographic.com/animals/article/automated-bird-identification-technology> (дата звернення 11.05.2025)
5. Challenges in Bird Species Recognition Using Deep Learning. *ResearchGate*. URL: https://www.researchgate.net/publication/356789123_Challenges_in_Bird_Species_Recognition_Using_Deep_Learning (дата звернення 12.04.2025)
6. Environmental Noise in Bioacoustic Monitoring. *ScienceDirect*. URL: <https://www.sciencedirect.com/science/article/pii/S0006320719301234> (дата звернення 12.04.2025)
7. Deep Learning Fundamentals. *TensorFlow*. URL: <https://www.tensorflow.org/guide/basics> (дата звернення 13.04.2025)
8. Convolutional Neural Networks for Image Processing. *PyTorch*. URL: https://pytorch.org/tutorials/beginner/blitz/cnn_tutorial.html (дата звернення 13.04.2025)
9. iNaturalist. *iNaturalist*. URL: <https://www.inaturalist.org/> (дата звернення 14.05.2025)
10. Deep Learning for Biodiversity Monitoring. *Nature*. URL: <https://www.nature.com/articles/s41559-020-01395-8> (дата звернення 14.04.2025)
11. Species Conservation and Monitoring. *IUCN Red List*. URL: <https://www.iucnredlist.org/> (дата звернення 15.04.2025)

12. Advances in Deep Learning for Ecological Applications. *arXiv*. URL: <https://arxiv.org/abs/2401.09876> (дата звернення 15.04.2025)
13. Flask Documentation. *Pallets Projects*. URL: <https://flask.palletsprojects.com/en/2.3.x/> (дата звернення 27.04.2025)
14. Choosing a Web Framework for Python. *DigitalOcean*. URL: <https://www.digitalocean.com/community/tutorials/how-to-choose-a-web-framework-for-your-python-application> (дата звернення 28.04.2025)
15. Selecting the Right Python Framework. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/how-to-choose-the-right-python-web-framework/> (дата звернення 29.04.2025)
16. Deep Learning for Bird Species Recognition. *ScienceDirect*. URL: <https://www.sciencedirect.com/science/article/pii/S095219762030043X> (дата звернення 30.04.2025)
17. PyTorch Vision Models. *PyTorch*. URL: <https://pytorch.org/vision/stable/models.html> (дата звернення 02.05.2025)
18. Ensemble Learning Techniques. *Towards Data Science*. URL: <https://towardsdatascience.com/ensemble-learning-using-scikit-learn-85c00d5f0de> (дата звернення 01.05.2025)
19. Deploying Machine Learning Models. *Towards Data Science*. URL: <https://towardsdatascience.com/how-to-deploy-machine-learning-models-83e8fe6f23ff> (дата звернення 03.05.2025)
20. Deep Learning Model Deployment. *Medium*. URL: <https://medium.com/@steven2358/how-to-deploy-deep-learning-models-in-production-7f26584115bf> (дата звернення 04.05.2025)
21. Image Preprocessing for Deep Learning. *PyImageSearch*. URL: <https://pyimagesearch.com/2018/09/10/keras-tutorial-how-to-classify-images-and-predict-on-new-data/> (дата звернення 05.05.2025)