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)

Технології міркування здорового глузду для обробки зображень та виявлення дипфейків
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІ-21-2

Олена Почерніна
(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 _____ Олег ЗОЛОТУХІН
(підпис) (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 Почерніній Олені Михайлівні
(прізвище, ім'я, по батькові)

1. Тема роботи Технології міркування здорового глузду для обробки зображень та виявлення дипфейків

затверджена наказом університету від 19 травня 20 25 р. № 378Ст
2. Термін подання студентом роботи до екзаменаційної комісії 25 червня 20 25 р.
3. Вихідні дані до роботи Науково технічні публікації, дані Інтернет-джерел, набори даних для класифікації дипфейків, набір даних для сегментації, сервіс Google Cloud, документація Python, документація Pytorch

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі та постановка задачі
2) Теоретичні дослідження
3) Проектування моделі та користувацького інтерфейсу
4) Програмна реалізація

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	19.05.2025	виконано
2	Аналіз предметної галузі	25.05.2025	виконано
3	Огляд аналогів системи	27.05.2025	виконано
4	Постановка задачі	28.05.2025	виконано
5	Теоретичні дослідження	02.06.2025	виконано
6	Проектування моделі та користувацького	08.06.2025	виконано
	інтерфейсу		
7	Програмна реалізація	14.06.2025	виконано
8	Написання пояснювальної записки	16.06.2025	виконано
9	Перевірка на академічний плагіат	17.06.2025	виконано
10	Нормоконтроль	18.06.2025	виконано
11	Підготовка презентації та доповіді	19.06.2025	виконано
12	Попередній захист	21.06.2025	виконано
13	Рецензування	23.06.2025	виконано
14	Захист перед ЕК	25.06.2025	

Дата видачі завдання 19 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Марія Головянко
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 118 с., 28 рис., 3 табл., 4 дод., 77 джерел.

АРТЕФАКТ, ВЕЛИКА МОВНА МОДЕЛЬ, ГЕНЕРАЦІЯ, ДИПФЕЙК, КЛАСИФІКАЦІЯ, МІРКУВАННЯ ЗДОРОВОГО ГЛУЗДУ, НЕЙРОННА МЕРЕЖА, СЕГМЕНТАЦІЯ, GEMINI, MASK R-CNN, RESNET-50.

Об’єкт дослідження – процес визначення реального зображення та фейкового за допомогою технології міркувань здорового глузду.

Предмет дослідження – реальні зображення та фальшиві зображення, що створенні за допомогою інструментів штучного інтелекту. До фальшивих зображень відносяться повністю згенеровані й частково змінені.

Мета роботи – розробка системи обробки зображення та виявлення дипфейків за допомогою технології міркування здорового глузду, що дозволить підвищити точність класифікації, вирішити проблему з узагальненням та покращити інтерпретацію результатів.

Методи дослідження – аналіз наукових публікацій, що освітлюють процес створення підробок та етичні сторони проблеми, бази даних інцидентів, пов’язаних зі штучним інтелектом, датасетів дипфейк контенту, а також порівняння існуючих рішень для вирішення проблеми.

ABSTRACT

Bachelor’s thesis contains: 118 pp., 28 fig., 3 tabl., 4 ann., 77 references.

ARTIFACT, CLASSIFICATION, COMMON SENSE REASONING, DEEPFAKE, GEMINI, GENERATION, LARGE LANGUAGE MODEL, MASK R-CNN, NEURAL NETWORK, RESNET-50, SEGMENTATION.

The object of the research is the process of detection a real image and a fake image using common sense reasoning technology.

The subject of the research is real images and fake images created using artificial intelligence tools. Fake images include fully generated and partially modified images.

The purpose of the work is to develop an image processing system and detect deepfake using common sense reasoning technology, which will increase the classification accuracy, solve the problem of generalization, and improve the interpretation of the results.

The research methods are analysis of scientific papers covering the process of creating deepfakes and the ethical aspects of the problem, databases of incidents related to artificial intelligence, content deepfake datasets, and comparison of existing solutions to solve the problem.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional neural network – згорткова нейронна мережа;

CSR – Common Sense Reasoning – міркування здорового глузду;

GAN – Generative Adversarial Network – генеративна змагальна мережа;

LLM – Large Language Model – велика мовна модель;

VAE – Variational Autoencoders – варіаційний автоенкодер;

VCR – Visual Commonsense Reasoning – візуальне міркування здорового глузду;

ViT – Vision Transformer – візуальний трансформер;

VQA – Visual Question Answer – візуальне запитання-відповідь.

ВСТУП

Із популяризацією глибинного навчання та розповсюдженням відкритих інструментів штучного інтелекту, з’являються нові способи та галузі їх використання. Однією з таких стала генерація синтетичних медіа – дипфейків, що можуть бути небезпечними для інформаційного суспільства.

Завдяки доступності інструментів створення фальсифікованого контенту, виготовлення стало можливим навіть для непрофесіоналів, які не розбираються в галузі штучного інтелекту. Навмисно шкідливе використання таких матеріалів веде до дезінформації окремих особистостей або великої групи людей, дискредитації публічних осіб, формування хибної громадянської думки, маніпуляції в політичних процесах, підриву довіри до медіа. Попри того, що здебільшого люди можуть відрізняти підробки від справжнього контенту, реалістичність дипфейків зростає кожний день з великою швидкістю.

Для розв’язання цієї задачі використовуються різні моделі машинного та глибинного навчання, а також їхні комбінацій. Проте, з урахуванням того, що алгоритми генерації постійно вдосконалюються, завдання залишається складним і потребує гнучких та адаптивних рішень. Існуючі моделі часто фокусуються на аналізі низькорівневих ознак, артефактах стиснення або спотворення обличчя, однак вони не завжди здатні виявити підробку, яка виглядає технічно бездоганно. Зокрема, все більшої уваги потребує аналіз не лише візуальних характеристик, але й змістової узгодженості, чи відповідає зображення звичайному здоровому глузду.

Здоровий глузд є ключовим фактором у виявленні дипфейків. Він дозволяє інтерпретувати контекст, помічати нелогічні речі та невідповідності в підробці. Завдяки йому користувачі можуть відрізнити навіть якісний дипфейк. Технології міркувань здорового глузду – це спроба відтворити людську здатність в інформаційних системах, вкласти в звичайні моделі штучного інтелекту знання про повсякденні факти. Такі системи

можуть розпізнавати абсурдні або малоймовірні сцени, які технічно виглядають правдоподібно, але суперечать базовим знанням про світ. Інтеграція таких механізмів у процес виявлення дипфейків відкриває нові можливості для підвищення надійності та точності автоматизованого аналізу медіа.

У кваліфікаційній роботі ставиться завдання розробити систему для обробки зображень та виявлення дипфейків, використовуючи технології міркування здорового глузду. Такий підхід дозволить підвищити рівень інтерпретації результатів та забезпечити стійкість до нових способів фальсифікацій, які стають усе складнішими для виявлення традиційними методами.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Дипфейк (від англ. «deep learning» – «глибинне навчання» та «fake» – «фальшивий», «фейковий») – форма синтетичних медіа, зображення, відео або аудіо, які зображують реальних або вигаданих людей, які роблять того, чого вони не робили. Як і всі синтетичні медіа, дипфейки використовують інструменти на основі штучного інтелекту для генерації або редагування контенту. Створення дипфейк-моделей вимагає великих обсягів даних і використання нейронних мереж для генерації фальшивих зображень, відео або аудіо. На рисунку 1.1 приведений приклад такого контенту, де вираз обличчя було змінено за допомогою програмного застосунку.

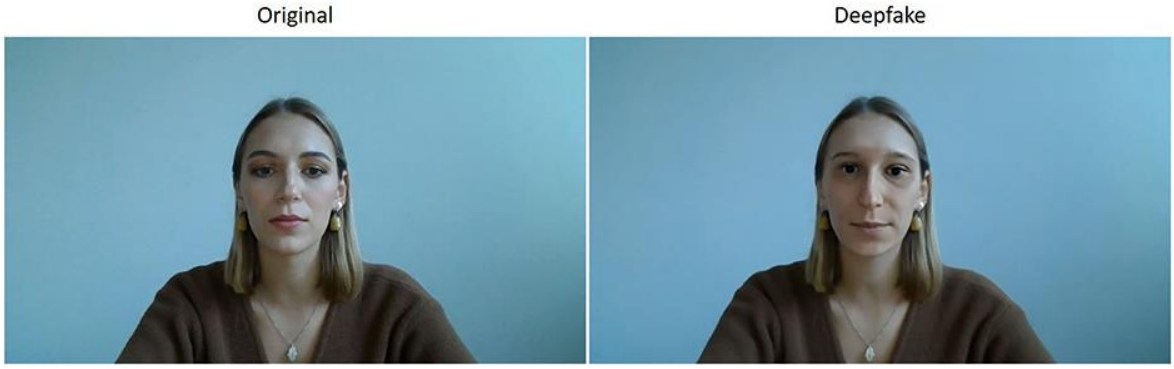

Рисунок 1.1 – Оригінальне (справа) та змінений дипфейк (зліва) [1]

Подібний медіаконтент почав своє існування в 2010-х роках, що пов’язується з проривом у сфері глибинного навчання. Одно з найперших згадувань в ЗМІ власне такого терміну з’явилось у кінці 2017 року [2]. Користувач соціальної платформи Reddit з нікнеймом «deepfake» опублікував реалістичні відеоролики компрометуючого характеру з голлівудськими актрисами, які насправді не були причетні до цього. На той момент дипфейки використовували інструменти машинного навчання

бібліотеки TensorFlow, розробленої компанією Google [3]. На сьогодні доступність відкритих інструментів на базі бібліотек TensorFlow та PyTorch від компанії MetaAI [4] дозволяють ентузіастам, які навіть не мають глибокі знання у сфері ШІ, створювати власні медіа за лічені години.

1.1.1 Типи дипфейків

Є декілька типів дипфейків, які відрізняються методами створення: текст, фото, аудіо, відео та дипфейки в реальному часі. Галузь дуже швидко розвивається та постійно з'являються нові технології для формування контенту, тому далі приведені лише найпопулярніші з них.

Текстові дипфейкі є сфабрикованими текстами, що створені за допомогою генеративних текстових моделей. Вони імітують стиль, думки та мовні особливості конкретної особи шляхом навчанні на вже наявних текстових повідомлень людини.

Фотореалістичні дипфейки базуються на методах «Face-swapping» чи «Face-morphing». У першому випадку обличчя однієї людини замінюється на обличчя іншої, у другому – відбувається плавне поєднання рис двох (або більше) осіб, у результаті чого виникає гібридний образ, «середній» між ними. Окремою загрозою становлять генерації «з нуля», які будуть розглянуті в частині 2.

Аудіо може бути згенерованим трьома шляхами: заміна голосу іншим («Voice-swapping»), конвертацією тексту в аудіо, частіше в універсальний голос, не прив'язаний до жодної людини («Text-to-Speech») або навчанням на записах певної людини для подальшої генерації голосу («Voice-cloning»).

Відеодипфейки створюють за допомогою комбінацій вже згаданих вище технологій, а також «Lip-syncing», при якому заміняється не лише обличчя та тіло, але також рух губ. Він повинен збігатися з вимовою, тому зазвичай необхідно окремо їх синхронізувати. Ще однією технологією для

створення аудіо-візуальних фейків є «Puppet-mastering», при цьому методі відбувається перенесення міміки й рухів голови з однієї на іншу та зберігається природні вирази.

Одними з найцікавіших медіа є дипфеки «в реальному часі». Вони створюється миттєво, під час трансляцій або відеодзвінків. До методів створення відносяться фільтри, що автоматично замінюють обличчя, та більш складний режим керування аватаром, який додатково дозволяє керувати рухами відтвореного об’єкту дипфейку.

1.1.2 Загроза

Використання дипфейків може бути шкідливим як для жертви дипфейку, так і для суспільства. Людина може навіть не знати, що його зображення або голос використали, що підсилює етичні та правові суперечності навколо дипфейків. Однією з ранніх проявів такого контенту є створення компрометуючого контенту, який зображує осіб, що без згоди вчиняють статевий акт, який ніколи не відбувався, зазвичай шляхом розміщення обличчя на тіло іншої людини. Такі медіа можуть бути руйнівними та мати негативний вплив на життя жертви.

Деякі дипфейки привертають багато уваги, коли вони стосуються публічних осіб. Лідери думок мають авторитет та довіру аудиторії, а також багато відео- та аудіоматеріалу для генерації дипфейків, тому їхні обличчя та голоси часто залучаються в шахрайські схеми або фейкові новини. Синтетичні медіа використовуються для поширення неправдивої інформації та маніпулювання громадською думкою, що становить значну загрозу для політичного дискурсу та суспільної довіри. Також дипфейки можуть використовуватися у фішингових атаках, коли зловмисники використовують переконливі голосові або відеоматеріали довірених осіб, змушуючи людей розкривати конфіденційну інформацію.

Оскільки такі матеріали стають все більш витонченими, довіряти відео- та аудіоконтенту стає дедалі важче, що призводить до широкого скептицизму щодо достовірності онлайн-медіа, знижуючи довіру до новинних агентств, публічних осіб і навіть до платформ соціальних мереж. Останні вводять обмеження на синтетичний контент, який дезінформує, використовується для шахрайства та порушує приватність людини. Наприклад, відеоплатформа YouTube розробляє нові технології для детекції ШІ-контенту, які допомагають творцям і видавцям боротися з несанкціонованим використанням їхніх особистостей [5].

У листопаді 2020 року некомерційна коаліція Partnership on AI [6], яка сприяє відповідальному розвитку й використанню штучного інтелекту, представила базу даних AI Incident [7]. Платформа є каталогом випадків з реального світу, коли інтелектуальні системи спричинили або майже спричинили шкоду. Головною метою проекту є сприяння прозорості, відповідальності та безпеці в галузі ШІ, дозволяючи вивчати минулі інциденти та запобігати подібних випадків у майбутньому. Наразі база даних налічує понад чотирьох тисяч повідомлень про одну тисячу інцидентів, близько двохсот з них належать до дипфейків.

1.1.3 Відомі випадки

Усі випадки відрізняються одне від одного типом створеного контенту та ступенем завданої шкоди. У 2018 році американський кінорежисер Джордан Піл сумісно з американським інтернет-виданням BuzzFeed створили відеоролик, де Барак Обама, колишній президент Сполучених Штатів, виступає з соціальною рекламою проти фейкових новин [8]. На рисунку 1.2 зображений політик та власне режисер.

Рисунок 1.2 – Кадр, де Барак Обама копіює промову Джордана Піла [9]

У відео кінорежисер голосом президента попереджає про те, як штучний інтелект може бути використаний для маніпуляціями людьми. У висновку Джордан Піл підкреслює, що треба бути більш пильними з тим, чому ми довіряємо в інтернеті. Відео було створено продюсерською компанією Піла, використовуючи поєднання старих та нових технологій: Adobe After Effects та інструменту FakeApp для заміни облич на основі штучного інтелекту, який застосовує TensorFlow. Створений ролик не є ідеальним: місцями він затримується, і часом досить очевидно, що щось не так з ротом фальшивого «Обами», деякі рухи руками скуті та повторюються.

16 березня 2022 року хакери зламали національну новинну програму на телеканалі «Україна 24» [10]. Новинна стрічка відображала повідомлення, ніби вони надходять від президента України Володимира Зеленського, в яких закликали українців припинити війну та здати зброю, водночас стверджувалося, що Зеленський «хотів захопити Донбас», але не досяг успіху, тому він утік з Києва. Хані Фарід, професор Каліфорнійського університету в Берклі, вказав на кілька очевидних ознак того, що відео є дипфейком: запис був низької якості та з низькою роздільною здатністю, що приховує артефакти генерації, Зеленський дивиться прямо перед собою, не рухаючи руками, а візуальний ряд має невеликі візуальні невідповідності.

Професор також зазначив, що, хоча він і не спілкується українською, голос президента звучить доволі дивно [11].

У 2024 році в соцмережі Facebook з'явилася реклама з участю співачки Тейлор Свіфт [12]. Саме відео розповідало про розіграш посуду компанії Le Creuset. Наївні користувачі переходили за посиланням, щоб отримати безкоштовні набори, але попадалися на фішинговий сайт, призначення для крадіжки особистих даних та стягування несанкціонованих платежів. Шахраї використовували технологію штучного інтелекту для створення синтетичної версії голосу співачки, аудіо відтворювалося поверх різних зображень продуктів Le Creuset та реально існуючих відео з Свіфт.

Також відомі випадки, коли звичайні люди зіткнулися з серйозними фінансовими маніпуляціями. У 2024 році фінансовий працівник компанії Arup, британської багатонаціональної фірми професійних послуг, виплатив 25 мільйонів доларів шахраям [13]. За даними поліції, працівник спочатку підозрював, що отримав фішинговий електронний лист з британського офісу компанії, оскільки в ньому зазначалася необхідність проведення секретної транзакції. Однак, після проведення відеодзвінка, працівник відкинув свої сумніви, оскільки фейковий фінансовий директор та інші присутні люди виглядали та говорили так само, як і в реальному житті.

Фотографічні дипфейки також можуть дезінформувати людей. У 2023 році в соціальній мережі X (колишній Twitter) поширилась фотографія Папи Римського Франциска в пуховику [14]. Користувачі інтернету не могли відрізнити, що матеріал є фейком, а лише змогли здогадатися про це з контексту. Через декілька днів після того Папа виступив з промовою, де зазначив, що штучний інтелект ставить серйозні питання та має використовуватися етично та відповідально для просування людської гідності та спільного блага [15]. Голова церкви не згадував дипфейки з його участю, тож нема доказів, що ці дві події пов'язані. Власне інцидент не завдав ніякої шкоди, але попереджає про те, як швидко еволюціонують

технології зі створення дипфейків. На рисунку 1.3 представлене зображення дипфейку.

Рисунок 1.3 – Дипфейк-зображення Папи Римського Франциска [16]

Того ж року в мережі з’явились підроблені фотографії арешту американського політика Дональда Трампа [17]. Деякі люди змогли одразу розпізнати фейк, а деяким додатково довелось проводити факт-чекінг в офіційних джерелах. У обох випадках фотографії були створені за допомогою програмного забезпечення Midjourney на базі дифузійних моделей [18], який генерує зображення на основі текстового опису.

1.1.4 Позитивний вплив та етичні проблеми

Необов’язково дипфейки можуть бути створені заради шкоди, деякі з них мають розважальну функцію. Доволі часто такий контент за участю відомих людей набирає популярності у випадках, коли доволі чітко видно, що це медіа є згенерованим. У 2025 році в соціальній мережі Instagram набуло поширення відео, на якому новообраний Папа Римський Лев XIV вимовляє фрази з популярного інтернет-мему [19]. Для користувачів є очевидним, що цей відеоматеріал є дипфейком, тому його сприйняття

відбувається в контексті іронії, де аудиторія усвідомлено інтерпретує синтетичний контент як форму пародії, а не спробу маніпуляції. Попри того, що контент не має на меті нашкодити, він все одно може бути образливим для реципієнта. З іншого боку, сприйняття неправдивих медіа в позитивному ключі сприяє поблажливому ставленню й зменшує етичні засудження.

Дипфейк-технології також можуть мати творчий потенціал. Існують інструменти, що дозволяють «оживляти» померлих акторів для нових ролей. Так, у 2016 році у фільмі «Rogue One: A Star Wars Story» з'явився персонаж Ґранд Мофф Таркін у виконанні актора Пітера Кушинга, який помер ще у 1994 році [20]. На рисунку 1.4 приведені два кадри з різних фільмів, де зображений актор.

Рисунок 1.4 – Зображення Пітера Кушинга в кіно: реального актора (зліва) та згенерованої моделі (справа) [21]

При першому аналізі не зрозуміло, на одному з фрагментів використана цифрова копія, але в порівнянні з іншим, реальним фото, різниця стає очевидною. Такий прояв дипфейків називається «Computer-

generated imagery», скорочено CGI, коли комп'ютерна графіка застосовується в кінематографі та телебаченні. Суспільство неоднозначно відреагувало на CGI-модель актора, виникла дискусія щодо етичності практики та міркувань щодо порушення прав та спадщини померлих акторів. Деякі відзначали, що цифрова копіє викликає в них ефект «моторошної долини», згідно з яким людиноподобні об'єкти викликають занепокоєння.

Ще один прояв використання дипфейк-технологій в кіно відбувся у фільмі «Ірландець», де застосували передову технологію цифрового омолодження, що дозволила акторам грати своїх персонажів у різні періоди життя без використання гриму [22]. На рисунку 1.5 представлене оригінальне зображення та відредаговане.

Рисунок 1.5 – Омолодження актора за допомогою дипфейк-технологій [23]

Також панує думка, що дипфейки сприяють розвитку цифрової свідомості суспільства. Як зазначає філософ Юбер Етьєн у своєму дослідженні, ці технології здатні підвищити рівень критичного мислення, стимулюючи перевірку інформаційних джерел [24]. Навпроти, в статті 2020 року Крістіан Ваккарі та Ендрю Чедвік висловлюють думку, що дипфейки

можуть посіяти невпевненість, яка, в свою чергу, знижує довіру до новин у соціальних мережах. Якщо користувачі ще менше довірятимуть новинам в інтернеті, вони можуть стати безвідповідальними при поширенні інформації, що сприятиме подальшому зниженню довіри та зростанню байдужості до правдивості в мережі [25].

1.2 Аналоги

Задача виявлення дипфейків є доволі поширеною, тому існує багато досліджень, присвячених аналізу різних підходів, які відрізняються архітектурою. Далі будуть розглянуто декілька різних систем, а також моделі, які виявляють фейкові зображення та відео.

Сайт Decopy AI [26] є безкоштовним онлайн-сервісом для виявлення контенту, створеного штучним інтелектом. Він пропонує низку сервісів, зокрема для розпізнавання ШІ-тексту, перефразування речень та інші корисні функції. Один із цих інструментів, AI Image Detector, дозволяє визначати, чи було зображення штучно згенероване, розпізнаючи такі популярні моделі, як Midjourney та DALL-E. Сервіс надає оцінку у вигляді відсотка ймовірності того, що зображення є фейковим. Водночас розробники не розголошують, яку саме модель використовують для аналізу зображень.

Ще одним інструментом є ресурс Fake Image Detector [27], що дозволяє виявляти змінені або згенеровані штучним інтелектом зображення. Сервіс приймає на вхід зображення та після обробки видає текстовий результат. Також, користувач отримує візуалізацію потенційних слідів маніпуляції в структурі пікселів (ELA аналіз). Як і попередня система, нема відкритої інформації про те, які засоби застосовуються для обробки.

Останнім розглянутим сервісом є FaceOnLive [28]. При завантаженні зображення сайт виводить відсоток належності до дипфейків, найбільш вірогідні генератори, використані для його створення, а також рівень

маніпуляції з обличчям. Сервіс, за словами розробників, аналізує патерни шуму, розподіл кольорів і невідповідності в структурі зображення.

Більшість моделей, які класифікують зображення на реальне та дипфейк, беруть за основу попередні створенні моделі (наприклад, моделі для задач загальної класифікації) та навчають їх на нових датасетах. Часто для таких цілей використовують моделі з вже натренованими вагами.

Спочатку розглянемо моделі, що базуються на згорткових нейронних мережах (англ. Convolutional Neural Network, CNN) [29], які досі є стандартом у галузі комп'ютерного зору та задач класифікації зображень.

Модель ResNet-50 вперше була представлена в 2015 році в роботі «Deep Residual Learning for Image Recognition» [30]. Головна ідея моделей з сімейства ResNet полягала у будуванні залишкових блоків (англ. residual blocks) та створенні зв'язків між ними (англ. skip-connections), дозволяючи ефективно навчати більш глибокі нейронні мережі, уникаючи проблем затухання градієнта й деградації моделі при додаванні нових шарів. Число в назві моделей співвідноситься з кількістю шарів. При навчанні ваги тренуються приглушувати шляхи, що не додають корисної інформації, та збільшувати більш важливі – таким чином відбувається коригування градієнту. У роботі 2024 року [31] модель, натренована на датасеті CIFAKE [32] отримала пікову точність в 98,87%.

Модель Xception початково була представлена в роботі Франсуа Шолле в 2017 році [33]. Нейронна мережа стала новою інтерпретацією згорткової мережі Inception [34], де складні блоки замінювались на більш прості шари глибинно-сепарабельної згортки (англ. depthwise separable convolution). Спочатку застосовується шар глибинної згортки (англ. depthwise convolution), яка оброблює ознаки в кожному каналі незалежно, потім використовується точкова згортка, щоб об'єднати інформацію в різних каналах. В 2019 модель була адаптована для використання виявлення дипфейк-контету, будучи натренованою на датасеті FaceForensics++ [35], точність склала 96,3%.

Окрім згорткових нейронних мереж, для виявлення дипфейків використовуються трансформери (англ. Vision Transformer, ViT) [36] та, на відміну від аналогів з CNN, потребують менше ресурсів. Такі моделі розбивають зображення на маленькі патчі та перетворює їх в уявне представлення у вигляді векторів, до яких додається позиційне кодування з інформацією про розташування патча на зображенні. Вектори подаються на вхід до трансформерної архітектури, що використовує механізм самоуваги для подальшої обробки локальних та глобальних зв'язків на зображенні. У роботі «Deepfake Image Detection using Vision Transformer Models» [37] натренована на датасеті від платформи Kaggle [38] модель досягла 89,9% точності.

У дослідженні «When Handcrafted Features and Deep Features Meet Mismatched Training and Test Sets for Deepfake Detection» [39] порівнюють точності двох типів моделей: глибинного навчання та ручних з використанням методів машинного навчання (SVM-класифікатор). Результати показали, що методи глибинного навчання все ще панують за ефективністю, проте ручні ознаки демонструють кращу роботу з новими даними, які суттєво відрізняються від навчальних. Таким чином розкривається потенціал комбінованих підходах у задачах класифікації дипфейків.

Більшість сучасних моделей та систем, розроблених для виявлення фейкового контенту, працюють за принципом бінарної класифікації або оцінюють ступінь реалістичності за шкалою ймовірностей. Хоча такі підходи дозволяють отримати швидку відповідь щодо достовірності контенту, вони не дають пояснення, чому зображений контент вважається підробленим.

Крім того, велика частина моделей тренується на вузьких, часто штучно створених датасетах, що добре репрезентують лише обмежене коло типових маніпуляцій. У результаті такі системи демонструють високу точність на знайомому датасеті, але значно втрачають ефективність при

перенесенні на нові дані, де зустрічаються незнайомі методи відтворення контенту. При створенні нових технологій підробок, попередні моделі швидко втрачають результативність.

Прямий аналог запропонованого розв'язку описаний в роботі «Common Sense Reasoning for Deepfake Detection» [40]. Автори пропонують новий підхід до виявлення дипфейків на основі міркувань здорового глузду. Нейронні мережі не можуть ефективно розпізнавати неприродні семантичні атрибути обличчя, наприклад, розмиті лінії волосся, подвійні брови, спотворені очі тощо, але подібні аномалії легко сприймаються людиною. Щоб усунути обмеження, автори пропонують розглядати виявлення дипфейків як задачу з візуальною відповіддю за запитання (DD-VQA), імітуючи людську інтуїцію за допомогою текстових пояснень для маркування зображення як справжнього або фейкового.

На рисунку 1.6 представлена приблизна схема моделі.

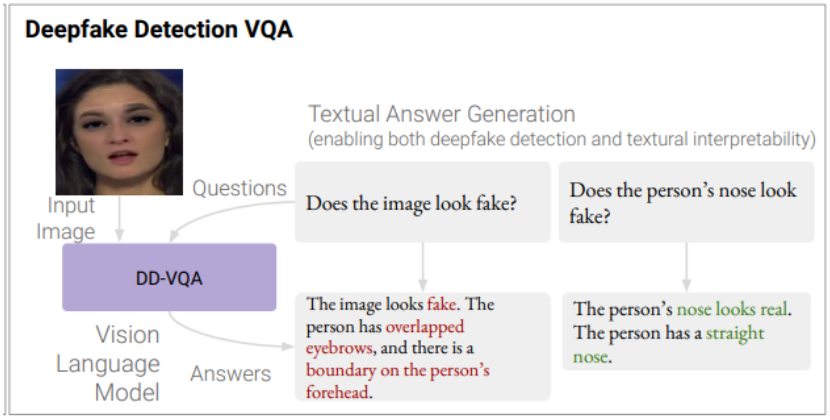

Рисунок 1.6 – Ілюстрація роботи алгоритму [40]

У відкритому доступі представлений набір даних, що містить зображення, взяті з датасету FaceForensics++, питання та відповіді. Окрім стандартного запиту («Чи виглядає зображення реальним/фейковим?»), питання ставляться до інших частин тіла (наприклад, «Чи виглядає ніс людини реальним/фейковим?»).

Результати експериментів показали, що такий підхід покращує ефективність виявлення дипфейків, допомагає адаптуватися до нових даних та дозволяє людині краще зрозуміти, чому модель прийняла певне рішення.

1.3 Постановка задачі

Метою цієї роботи є розробка системи, яка застосовує технології міркування здорового глузду для обробки зображення та виявлення дипфейків. Для досягнення мети необхідно вирішити конкретні задачі та виконати наступні дії:

– визначити основні моделі для створення дипфейків, виявити візуальні недосконалості та артефакти генерації;

– проаналізувати людські стратегії в задачі виявлення дипфейків;

– дослідити міркування здорового глузду та підходи, а також проаналізувати наявні моделі;

– спроектувати модель для обробки зображень та визначення дипфейків;

– визначити вимоги користувача та інтерфейсу системи;

– реалізувати основні компоненти моделі;

– навчити модель на наявних відкритих даних з доступних джерел;

– представити систему у вигляді робочого застосунку, реалізувати користувацький інтерфейс;

– оцінити результати;

– зробити висновки та запропонувати подальші шляхи розвитку створеної системи.

Система отримує на вхід зображення, обране користувачем, після чого виконує його аналіз. У результаті обробки користувач отримує текстове повідомлення: якщо зображення є справжнім – система підтверджує його автентичність; якщо ж воно виявляється дипфейком, користувач отримає відповідне пояснення про виявлені ознаки підробки.

2 ТЕОРЕТИЧНІ ДОСЛІДЖЕННЯ

2.1 Моделі генерації

Контент створюється за допомогою різних моделей глибинного навчання. Вони навчаються на великих обсягах даних, мають свої закономірності та галузі застосування, підходять під різни типи генерацій. Окрім наявних методів маніпуляції даними для створення дипфейків, описаних в 1.2, моделі можуть відтворювати контент за текстовим описом, попередньо навчившись на великих датасетах або на даних з інтернету.

Перша з моделей є генеративна змагальна мережа (англ. Generative Adversarial Network) [41], які складаються з двох типів нейронних мереж: генератор, який тренується генерувати контент, та дискримінатор, який тренується визначати фейкове згенероване зображення. Мережі змагаються між собою в процесі навчання: генератор намагається створити настільки правдиве медіа, щоб «обдурити» дискримінатор, а той, у свою чергу, удосконалюється у виявленні підробок. На рисунку 2.1 представлений приклад генерації GAN.

Рисунок 2.1 – Зображення, згенероване за допомогою GAN [42]

Другою генеративною моделлю є варіаційний автоенкодер (англ. Variational Autoencoders) [43], яка працює на базі автоенкодерів. Перша частина, енкодер, перетворює дані у латентне представлення, а друга частина, декодер, відновлює контент з нього. На відміну від звичайних автоенкодерів, де представлення є одним вектором, VAE зберігає представлення у двох векторах: вектор середніх значень параметрів та вектор стандартних відхилень. Завдяки такому представленню модель більше підходить для генерації різноманітного контенту. Порівнюючи з GANs, VAEs дозволяють керувати параметрами, наприклад, позами, емоціями або освітленням, хоч зазвичай такі медіа можуть бути менш фотореалістичними. На рисунку 2.2 представлен приклад генерації автоенкодеру, де змінюється вектор «smile».

Рисунок 2.2 – Зображення, згенероване за допомогою VAE [44]

Останньою розглянутою моделлю є дифузійна модель (англ. Diffusion Model) [45], що базується на зворотному процесі шумування: спочатку до вхідних даних поступово додається шум, потім модель вчиться відновлювати дані назад. Такі моделі демонструють високу ефективність в генерації фото- та аудіореалістичного контенту. Дифузійні моделі широко використовуються в популярних генераторах зображень, наприклад, сервісу DALL-E [46] від компанії OpenAI. На рисунку 2.3 представлений приклад подібної генерації.

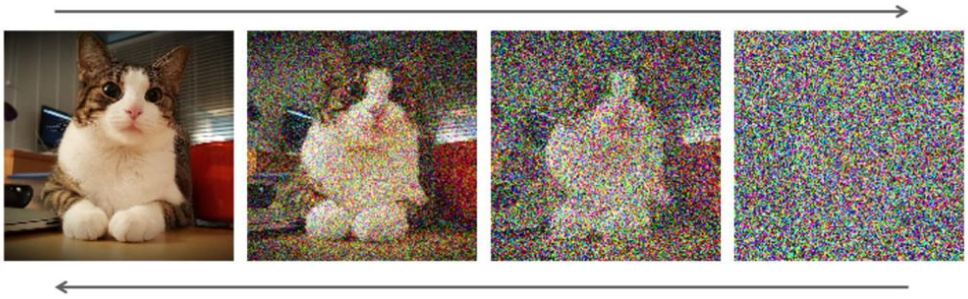

Рисунок 2.3 – Генерація зображення за допомогою дифузійної моделі [47]

2.2 Візуальні артефакти

Дипфейк-зображення часто містять візуально-помітні помилки, за якими можна відстежити підробку.

У роботі «Characterizing Photorealism and Artifacts in Diffusion Model-Generated Images» [48] представлена окрема таксономія візуальних характеристик, які є підсказками до того, що зображення було штучно згенеровано за допомогою дифузійних моделей.

Одним з найбільш поширених типів є анатомічні аномалії. Вони охоплюють артефакти окремих частин тіла, такі як руки з додатковими пальцями, зайві кінцівки, непропорційні шиї тощо. Також до подібних аномалій входять артефакти обличчя, «порожній» погляд, надмірно блискучі очі, перекриття зубів. У зображенні з кількома людьми присутні злиття частин тіла окремих осіб та нереалістичні пропорції між ними.

Стилістичні артефакти включають загальну глянцевість зображення та неприродну однорідність текстур. До них належать згладжені шкіра та волосся, ідеалізоване освітлення тощо. Аномалії стилю можуть поширюватися на фон та аксесуари, наприклад, візерункові текстури з повторюваними сітками, які більшість людей сприймають «занадто ідеальними». Такі артефакти також належать генеративним моделям [49]. Також у GAN моделях виділяється надзвичайна симетрія [50]. У реальних

обличчях так чи інакше присутня легка асиметрія, але моделям важко її відтворити.

Функціональні аномалії пов’язані з порушенням логіки реального світу, об’єктами, які не функціонують належно або взаємодія з ними виглядить нереалістично. Часто спостерігаються спотворення в деталях одягу, неправильні блискавки, ґудзики, надписи тощо.

Неправильна фізики проявляються в невідповідності віддзеркалень, помилках перспективи, порушенні логіки тіней. Як для дифузійних моделей, так і для генеративних освітлення має вагомий вплив [49]. Неузгодженість світла швидко відмічається людським оком, помилки особливо помітні навколо обличчя.

Останнім відокремленим типом є соціокультурні аномалії, що стосуються ситуацій, які малоймовірні з точки зору прийнятих норм. Наприклад, неузгодженість одягу та контексту (наприклад, якщо людина вдягнута в офіційний одяг під водою) або зображення відомих осіб у хибних історично-культурних умовах (наприклад, зображення Альберта Ейнштейна в 21-му столітті). Також мають місце помилки у відтворенні національних елементів, що нехарактерні для зображеної культури.

2.3 Людські стратегії виявлення дипфейків

Більшість досліджень висловлюють думку, що люди проявляють низьку ефективність у виявленні дипфейків [51]. Водночас деякі роботи відмічають, що люди здатні визначати ті підробки, які машини не здатні помітити [52], що підкреслює потенціал людського міркування в якості додаткового помічника.

У процесі розпізнавання дипфейків люди, як правило, схильні зосереджувати увагу на найбільш помітних рисах обличчя (такі як очі, рот) та ігнорують інші області зображення, де можуть знаходитись помилки генерації [53]. Проте нові дослідження показують: суттєвий вплив на

виявлення дипфейків мають фони, що свідчить про їхню недосконалу генерацію, в той самий час як колір шкіри та аксесуари майже не впливають на рішення [54]. Іншими факторами є риси обличчя, контури, одяг та волосся.

На рисунку 2.4 зображений розподіл, на які фрагменти зображення частіше всього привертали увагу учасники експерименту, описаного в роботі «Analysis of human perception in distinguishing real and ai-generated faces: an eye-tracking based study» [54].

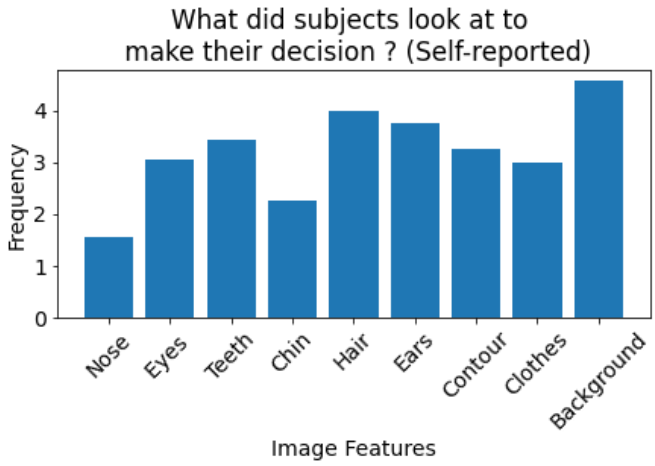

Рисунок 2.4 – Ознаки, на які частіше всього звертають увагу люди під час виявлення дипфейків [54]

2.4. Візуальна відповідь на запитання

Visual Question Answering (VQA) – це задача, яка поєднує методи комп'ютерного зору та обробки природної мови: моделі подається зображення разом із запитанням, сформульованим природною мовою, а її завданням є надати текстову відповідь.

Для вирішення задачі модель повинна:

– проаналізувати зображення;

– обробити текстове запитання;

– пов’язати обидва джерела;

– згенерувати або обрати відповідь.

Частіше в таких моделях використовуються CNN або ViT задля обробки зображення. Для обробки тексту використовують велику мовну модель (англ. Large Language Model, LLM), яка спеціалізується на обробці, розумінні та генерації людської мови. Такі моделі здатні не лише генерувати граматично правильний текст, а й імітувати логіку, стиль, контекст та певний рівень міркування.

LLM побудована на основі трансформеру й механізму самоуваги [55], що дозволяє моделі зважувати важливість різних слів на різних частинах вхідної інформації. Вхідний текст розбивається на токени, наприклад, слова або частини слів, які перетворюються на вектори. Під час тренування модель поглинає великі масиви текстових даних, навчаючись передбачати наступні токени, заповнювати пропущені та розуміти зв'язки між ними. Після навчання LLM приймає вхідний запит (промпт), який використовується для генерації наступних токенів, що формують осмислену текстову відповідь. Відповідь створюється поетапно залежно від найбільш ймовірних наступних елементів у контексті.

Мовні моделі покладаються на статистичні закономірності, а не на людські міркування, можуть генерувати правдоподібну, але неправдиву інформацію, що може бути критичним в сферах, де очікується фактологічна достовірність. У зв’язку з цим важливо формулювати запити максимально точно та контролювати результат за допомогою верифікації.

2.5 Міркування здорового глузду

Незважаючи на успіхи в галузі штучного інтелекту, машини досі зазнають труднощів із завданнями, які вимагають елементарного здорового глузду, що знижує їхню ефективність в сферах обробки природньої мови та комп’ютерного зору. Наприклад, сучасні моделі не можуть зрозуміти, що

вода тече з пляшки, тому що вона лежить горизонтально, або підкинутий м’яч у повітрі полетить вниз. Подібні розсуди базуються не лише на конкретних знаннях, але й неочевидних, «інтуїтивних» міркуваннях. Наділити системи ШІ «здоровим глуздом» є складною задачею, що потребує розуміння базових уявлень про світ, причині-наслідкові зв’язки та людську поведінку.

У роботі «Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence» [56] виділяються декілька складнощів такої задачі. По-перше, здоровий глузд охоплює величезний спектр фактів та уявлень, які людина набуває підсвідомо з досвідом. По-друге, міркування на основі здорового глузду дуже залежать від контексту. По-третє, значною перешкодою є охоплення всіх тонкощів людських знань у вигляді жорстких правил або наборів даних, які машини здатні обробити.

Також автори аналізують різні підходи до моделювання здорового глузду: формальну логіку, ручне створення бази знань та наборів даних, машинне збирання фактів з текстових ресурсів, а також краудсорсинг. Проте жоден з методів не є достатнім, вони стикаються з труднощами масштабування, повноти та точності. На думку дослідників, майбутнє міркування здорового глузду полягає в поєднанні різних підходів, а також у більш глибокому вивченні людської інтуїції та когнітивних механізмів. Без залучення здорового глузду ШІ не зможе повноцінно розуміти навколишній світ, будувати логічні міркування та надійно взаємодіяти з людьми в реальному середовищі.

2.5 Міркування здорового глузду в обробці зображення

Прості моделі навчилися виконувати базові завдання (наприклад, розпізнавання об’єктів, відповідь на запитання типу «Який колір машини?»), але не вміють робити висновки на рівні здорового глузду. Візуальне міркування здорового глузду (Visual Commonsense Reasoning,

VCR) вимагає від моделей не тільки відповідати на запитання про зображення, а й обґрунтовувати свої відповіді, що вимагає високорівневого когнітивного мислення та знань, які виходять за рамки звичайного розпізнавання об'єктів.

Для створення моделей, які здатні проводити міркування на основі здорового глузду, пропонуються декілька різних підходів.

2.5.1 Мультимодальне попереднє навчання та злиття ознак

Одним з популярних підходів є мультимодальне попереднє навчання. До архітектури таких моделей входить як кодер для візуальної інформації, так і мовна модель, частіше LLM.

BLIP-2 (Bootstrapping Language-Image Pre-training) – попередньо-навчена візуально-мовна система, яка складається з двох заморожених модулів для зображення та тексту, які не підлягають тренуванню. Замістить них навчається трансформер запитів (Q-Former), який витягує ознаки з замороженої моделі тексту та перетворює у форму, яку може обробити мовна модель. На рисунку 2.5 зображена архітектура моделі.

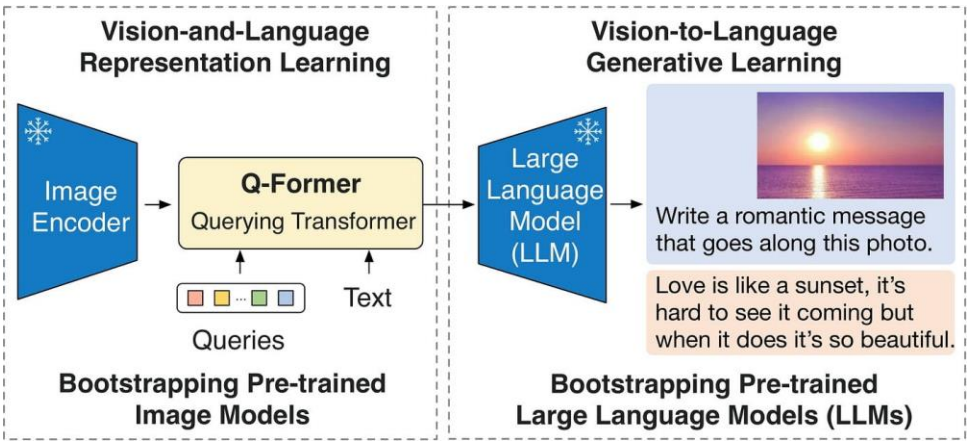

Рисунок 2.5 – Архітектура BLIP-2 [57]

LLaVA (Large Language and Vision Assistant) [58] – ще один приклад мультимодальної моделі. На відміну від BLIP-2, де використовується окремий трансформер запитів, у LLaVA вхідне зображення обробляється замороженим візуальним кодером, після чого ознаки передаються у проекційний шар. Проекційна матриця перетворює вихідні ознаки з візуального кодера у простір ознак, які об'єднуються з текстовою інструкцією, поданою користувачем, та подаються на вхід LLM. Архітектура моделі подана на рисунку 2.6.

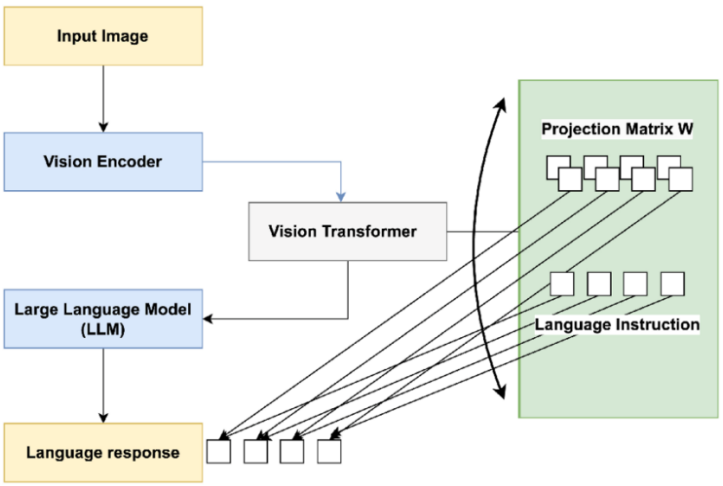

Рисунок 2.6 – Архітектура LLaVA [59]

Останнім розглянутим прикладом є Flamingo [60] – потужна мультимодальна модель, що інтегрує візуальну інформацію безпосередньо в процес генерації тексту за допомогою механізму крос-уваги. Модель отримує на вхід чергування тексту та зображень, що представлені у вигляді потоків.

Вхідне зображення обробляється замороженим кодером, після чого ознаки передаються до спеціального модуля, який був натренований з нуля та стискає високорозмірні візуальні представлення у компактну послідовність токенів фіксованого розміру. Ці токени передаються до замороженої великої мовної моделі, в яку були інтегровані міжмодальні

шари, які змінюються під час навчання, що дозволяє ефективно передавати візуальну інформацію, не перенавчаючи LLM з нуля. На рисунку 2.7 зображено архітектуру системи.

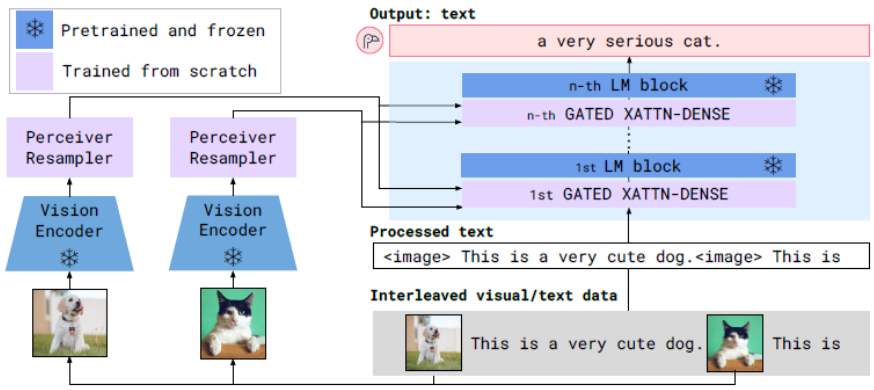

Рисунок 2.7 – Архітектура Flamingo [61]

2.5.2 Застосування зовнішніх знань

Другим підходом, що підвищує ефективність VCR-моделей є населення моделей знаннями, взятими з зовнішніх джерел. Зазвичай в ролі таких джерел виступають графи, бази даних та датасети.

Одним з найтиповіших представників є ConceptNet [62], що є доступною семантичною мережою, призначеною для зберігання загальних знань про світ, які люди сприймають як належне. Ресурс бере свій початок з проекту «Open Mind Common Sense» 1999 року медіа-лабораторії Массачусетського технологічного інституту, який зібрав знання про здоровий глузд від тисяч людей. ConceptNet кодує факти у вигляді відношень між об'єктами за допомогою таких зв’язків, як «IsA», «UsedFor», «CapableOf» тощо, дозволяючи машинам семантично пов’язувати поняття та розуміти мову за межами словникових значень. ConceptNet використовується для покращення розуміння природної мови, допомоги у відповідях на запитання та впровадженні міркування здорового глузду.

Ще одним графом знань є Atomic [63], який фіксує знання за допомогою зв’язків «If-then». Факти складаються з головної події та кінцевої події, що мають на меті забезпечити системи ШІ знаннями про причини та наслідки, пов’язаними з повсякденними ситуаціями. Автори зазначають, що багатозадачні моделі, що включають подібну ієрархічну структуру зв’язків, мають більш точний висновок, аніж інші ізольовані моделі.

Серед прикладів моделей є VisualComet [64] – фреймворк для міркувань здорового глузду на основі візуального аналізу, що фокусується на прогнозуванні подій, які сталися раніше, та подій, які стануться в майбутньому, а також подій поза кадром. Графова структура моделі нагадує Atomic та містить великий репозиторій з 60 тисяч зображень та понад 1,4 мільйонів текстових анотацій. Фреймворк дозволяє генерувати висновки на основі здорового глузду про динамічний контекст, що оточує нерухоме зображення. VisualComet розширює попередньо навчений GPT-2, щоб приймати характеристики зображення як додаткові вхідні дані та генерувати описи подій.

Окремо можна виділити інструмент Physion [65], бенчмарк, що оцінює моделі штучного інтелекту на основі людської інтуїції щодо фізики об’єктів, як вони рухаються та взаємодіють одне з одним. До датасету входить понад тисячі прикладів фізичних явищ, падіння, ковзання, зіткнення, деформації тощо. Дослідження показують, що моделі, які оснащені подібними знаннями, демонструють більшу ефективність в розумінні фізичної сцени.

2.5.3 Механізм підказок

Підказки, які закладені в промпті до зображення, слугують своєрідною інструкцією для моделі. Вони задають контекст, формулюють очікуване завдання та спрямовують увагу на конкретні аспекти вхідних

даних. Наприклад, запити «Що незвичного на цьому зображенні?», «Опиши взаємодію між об'єктами» або «Чому ця сцена здається нелогічною?» активізують у моделі механізми міркування, змушуючи її не лише інтерпретувати візуальну інформацію, а й робити висновки на основі знань та контексту.

Подібні підказки поєднуються з покроковими міркуваннями. У роботі «Multimodal Chain-of-Thought Reasoning in Language Models» [66] автори пропонують генерувати послідовність міркувань, а вже потім на основі цієї послідовності формулювати фінальну відповідь. На першому етапі модель генерує пояснення або міркування щодо ситуації на зображенні, а на другому використовує це міркування як доповнений контекст для формування точної відповіді.

3 ПРОЕКТУВАННЯ МОДЕЛІ ТА КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ

3.1 Побудова моделі

У зв’язку з обмеженими ресурсами для тренування великих моделей, що зможуть відповісти на запитання та обробити зображення одночасно, пропонується використати наявну модель та наділити її контектом у вигляді текстового запиту, а також дати підказки про те, які фактори слід врахувати під час виявлення дипфейку.

Для виконання поставленої задачі була запропонована мультимодальна модель (рисунок 3.1), яка поєднує декілька модулей.

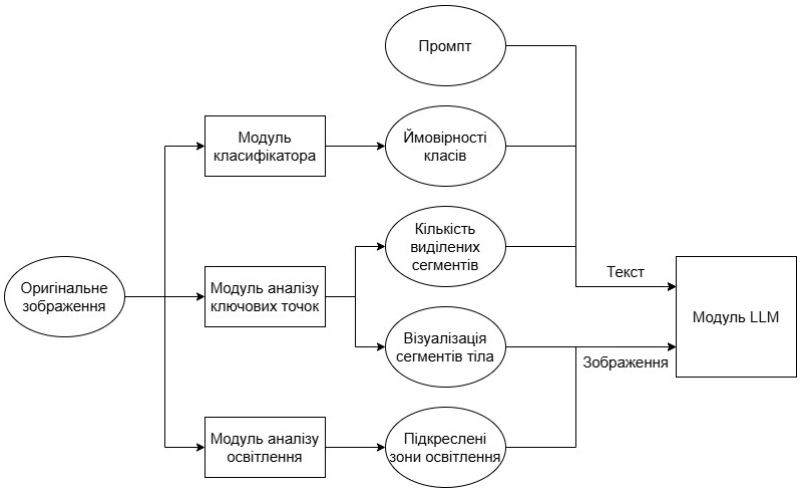

Рисунок 3.1 – Схема роботи моделі

Повна структура моделі містить:

– модуль класифікатора, що класифікує зображення за певним класом;

– модуль аналізу ключових точок, що визначає сегменти на зображені;

– модуль аналізу освітлення, що виділяє темні та світлі зони;

– модуль LLM, на вхід до якого подаються текстові та візуальні дані.

3.1.2 Модуль класифікатора

Модуль має на меті забезпечити додатковим контекстом про штучне походження зображення, але не спиратись на нього. Його завдання полягає в визначенні ймовірності, що зображення належить до певного класу. Модуль приймає на вході зображення та пропускає його через модель класифікатора з попередньо натренованими вагами.

Для класифікатора була обрана нейронна мережа ResNet-50. Архітектура широко використовується для задач класифікації зображень, виявлення об’єктів та визначення ознак. Вона була розроблена підрозділом Microsoft Research Asia у 2015 році та представлена у роботі «Deep Residual Learning for Image Recognition» [30], наряду з іншими моделями сімейства ResNet. Головна ідея моделей з сімейства ResNet полягає у впровадженні залишкових блоків (англ. residual blocks) та створення спеціальних з’єднань між шарами (англ. skip-connections). Замість того, щоб кожен шар використовував вихід попереднього, залишкові з’єднання дозволяють пропускати один або більше шарів.

Подібні моделі уникають проблему затухання та вибуху градієнта. Під час тренування нейронних мереж за допомогою зворотного поширення помилки (англ. backpropagation) градієнти, проходячи по всім шарам, можуть становити надзвичайно малими (або надзвичайно великими). Ваги в початкових шарах майже не змінюються (або змінюються дуже сильно), що призводить до неефективному навчанні нейронної мережи. Залишкові з’єднання в ResNet дозволяють градієнтам напряму проходити через мережу, минаючи шари, що забезпечує стабільний потік градієнтів у глибоких мережах.

На рисунку 3.2 подана архітектура моделі ResNet-50.

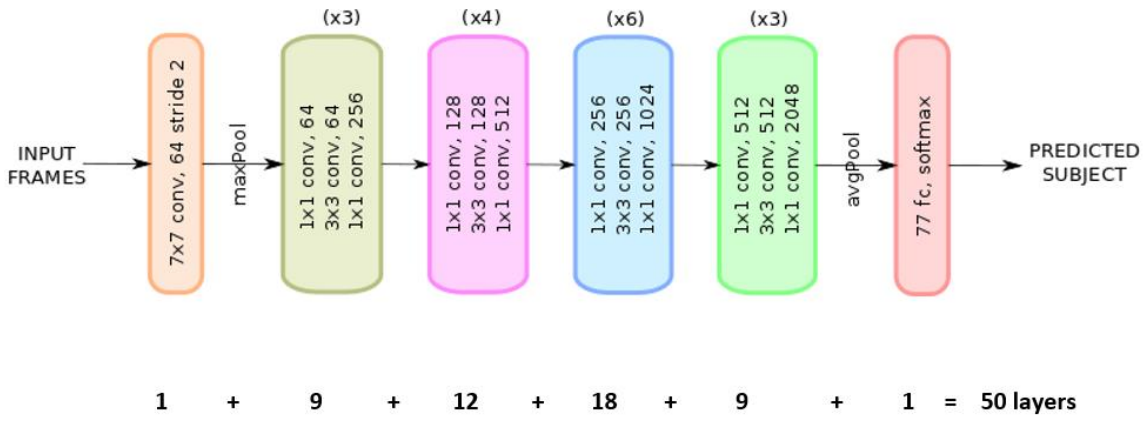

Рисунок 3.2 – Архітектура ResNet-50 [67]

Число в назві моделей ResNet відповідає кількості шарів, ResNet-50 використовує 50 шарів відповідно. Початковий шар є згортковим, після якого йдуть залишкові блоки. Архітектура починається з вхідного шару, що приймає зображення, за ним йде початковий згортковий шар з використанням фільтрів 7x7 з 64 каналами та кроком 2, після нього відбувається операція максимального пулінгу (англ. max pooling).

Ядро мережи складається з чотирьох залишкових блоків, які мають однакову структуру згорток (1x1, 3x3 та 1x1):

– перший блок має 64, 64 та 256 вихідні канали відповідно, згортки повторюються тричі;

– другий блок має 128, 128 та 512 канали, повторюються чотири рази;

– третій блок має 512, 512 та 1024 канали, повторюються шість разів;

– четвертий блок має 1024, 1024 та 2048 каналів, повторюються тричі.

Після блоків застосовується середній пулінг (англ. average pooling). Останнім є повноз’єднаний шар з 77 єдиниць та софтмакс фукнцією активації для отримання прогнозованого класу.

Для використання класифікатора для задачі виявлення дипфейків необхідно натренувати мережу на наборі даних з відповідними класами. Для цього був обраний набір даних «AI vs Deepfake vs Real», доступний на

HuggingFace [68]. Датасет призначений для задачі класифікації та містить 10 тисяч зображень, рівномірно розподілених між трьома класами (штучні, дипфейк та справжні). Він охоплює різноманітні приклади, що сприяє підвищенню точності та загальної ефективності моделі.

Для покращення навчання необхідно використати попередньо натреновані ваги на стандартному датасеті з класифікації зображень. За основу була взяла модель ResNet-50, натренована на датасеті ImageNet [69].

3.1.2 Модуль аналізу ключових точок

Модуль запропонований для аналізу частин тіла, які мають знадобитися для контексту зображення. Згенеровані зображення часто мають зайві кінцівки, що системи часто не помічають, тому запропонована модель визначає спільну кількість кінцівок та їхні контури.

Модуль приймає на вході зображення та пропускає його через модель сегментації. Моделі сегментації призначені для поділу зображення на окремі області або об’єкти, наприклад, для виділення контурів людей, їхніх частин тіла чи інших важливих елементів, що дозволяє проводити подальший аналіз аномалій.

Моделлю стала Mask R-CNN [70], що є глибокою нейронною мережою, яка використовується для виявлення об’єктів на зображенні та побудови сегментаційних масок для кожного об’єкта. Вона є розширенням моделі Faster R-CNN [71], яка виконує класифікацію об'єктів та детекцію їх на певній області зображення. Mask R-CNN додає гілку для піксельної сегментації кожного об’єкта, що дозволяє визначити точні межі об’єктів на зображенні.

На рисунку 3.3 зображений приклад архітектури моделі.

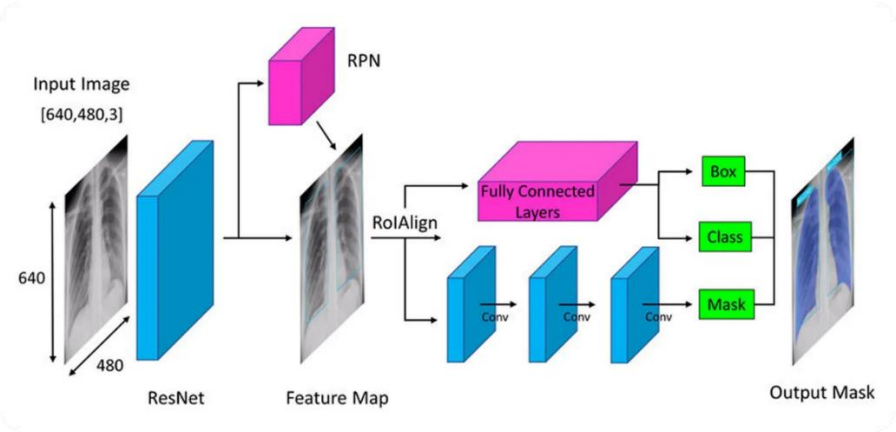

Рисунок 3.3 – Структура Mask R-CNN для виявлення COVID-19 [72]

Модель Mask R-CNN складається з трьох основних компонентів:

– першим компонентом є глибока згорткова мережа, яка витягує ознаки з вхідного зображення (англ. backbone), частіше всього в цій ролі виступає моделі сімейства ResNet;

– другим компонентом є модуль, який виявляє регіони з потенційною наявністю об'єктів (Region Proposal Network);

– останнім компонентом є операція, що дозволяє витягнути ознаки для кожного виявленого регіону (RoIAlign).

На зображеному прикладі витягнуті ознаки з кожної області подаються на повнозв’язні шари для класифікації або регресії, які потім формуються у координати обмежувальної рамки та клас об’єкта. Інша гілка мережі пропускає витягнуті ознаки через згорткові шарі, що на виході створюють маску, яка показує пікселі об’єкта.

У якості бекбону використовується ResNet-50, під’єднаний до Feature Pyramid Network [73], що дозволяє працювати з об'єктами різного масштабу.

Для тренування був обраний набір даних COCO Keypoints 2017, який є частиною масштабного датасету COCO 2017 [74] для виявлення, сегментації та підпису об'єктів. COCO Keypoints спеціалізується на оцінці пози людини та містить анотації ключових точок для кожної людини на зображенні. Датасет містить 150 тисяч прикладів людей, для кожної з яких

вказано 17 ключових точок тіла з координатами, Зображення представлені в широкому спектрі варіацій: різні пози, кути огляду, кількість людей на сцені, взаємодії між людьми та часткове перекриття тіл.

3.1.3 Модуль аналізу освітлення

Модуль призначений для яскравої візуалізації освітлення, яке на оригінальному зображені не таке помітне, тому штучність походження не зможе зчитатись моделлю, яка приймає рішення. Модуль приймає на вході зображення, виділяє його темні та світлі зони та візуально «підсилює» їх. Спочатку зображення перетворюється в сіру гаму та розмивається, після цього інвертується і створюється маска тіней, яка накладається на оригінал. Тіни підсилюються шляхом зменшення яскравості. У результаті повертається модифіковане зображення, яке додається до контексту.

3.1.4 Модуль великої мовної моделі

Останнім модулем, який відповідає за прийняте рішення, є велика мовна модель.

Замість звичайного промпту («Чи є зображення дипфейком?»), необхідно використати спеціально написаний текст, який буде вказувати на те, на що варто звернути увагу при аналізі зображення, наприклад, варто звертати увагу на природність освітлення, симетрію обличчя, артефакти навколо очей, рота та волосся тощо. Для отримання розгорнутої відповіді, в промпті необхідно вказати, щоб мовна модель дала пояснення своїм розсудам, описала, які ознаки були враховані, та чому саме ці ознаки свідчать про реальність або штучність зображення. Також у промпті буде використаний контекст, взятий з попередніх модулів системи.

Обраною моделлю стала Gemini 2.5 Pro [75]. Gemini є сімейством мультимодальних великих мовних моделей, розроблених Google AI. Вони

здатні обробляти текстову та візуальну інформацію, що робить їх придатними для завдань, пов’язаних з аналізом зображень у поєднанні з природною мовою. Моделі Gemini демонструють високі результати в задачах генерації, перекладу, розпізнавання образів, кодування та міркування, що підтверджується тестами на бенчмарках.

Gemini 2.5 Pro є однією з найпотужніших версій цього сімейства, оптимізованою для складних запитів, що потребують багатокрокового міркування, роботи з зображеннями, текстом та їхньою взаємодією. Ключовими факторами, які стали вирішальними в обрання цієї версії моделі, полягають в покращеному мисленню, яке є основою для міркування здорового глузду. Також модель здатна ефективно обробляти великі обсяги тексту (до мільйона токенів) та наряду з текстом приймає зображення, аудіо та відео.

3.2 Проектування користувацького інтерфейсу

Основною метою користувача є отримання інформації про зображення, перевірити, чи є воно дипфейком. Користувачі системі – це люди, що схильні перевіряти факти, найголовнішим чинником є здатність до критичного мислення. Функціонал системи може знадобитись як звичайним користувачам в інтернеті, які схильні перевіряти факти, так і журналістам, що пишуть статті, та факт-чекерам.

Єдиним обмеження системи є знання англійською мови, яка є мовою інтерфейсу застосунку.

Для використання системи необхідно розташовуватись вдома, в офісах, робочих просторах або інших містах за наявністю стаціонарного комп’ютеру або ноутбуку з браузером та доступному інтернет-підключенні.

Вимоги користувачів:

− лаконічний текстовий аналіз картинки;

− доступність до сайту в будь-який час доби;

− зручний та простий інтерфейс;

− наявна сторінка, де можна дізнатись як спроектована система;

− підтримка обрання зображення з файлової системи комп’ютеру.

Для проектування інтерфейсу системи аналізу зображень з використанням штучного інтелекту знадобилося врахувати функціональні вимоги та зручність взаємодії для користувача. Компонентно-функціональна, функціонально-об'єктна та функціонально-часова структури подані в Додатку А.

На рисунку 3.4 зображений прототип інтерфейсу сторінки.

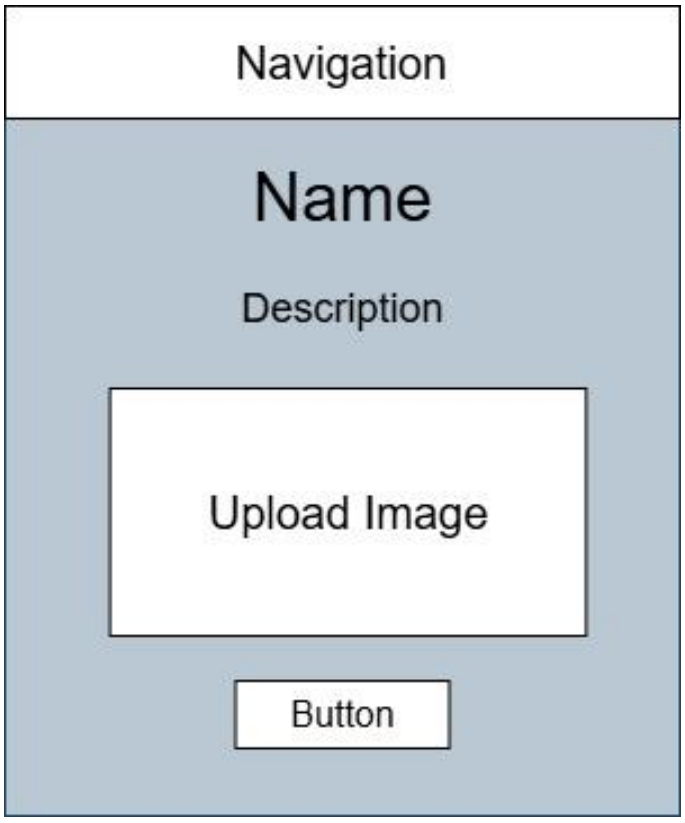

Рисунок 3.4 – Прототип інтерфейсу сторінки

На сторінці зображені основні елементи інтерфейсу, що забезпечують зручну взаємодію користувача з системою. Центральне місце займає назва системи, нижче розміщено короткий опис, який пояснює основну мету та функціональність системи. Під описом розташоване поле для завантаження

зображення, після якого йде кнопка, що запускає процес аналізу зображення. Також сторінка містить навігаційну панель, яка дозволяє легко переходити на інші сторінки сайту.

На рисунку 3.5 зображений інтерфейс сторінки після завантаження зображення та отримання аналізу.

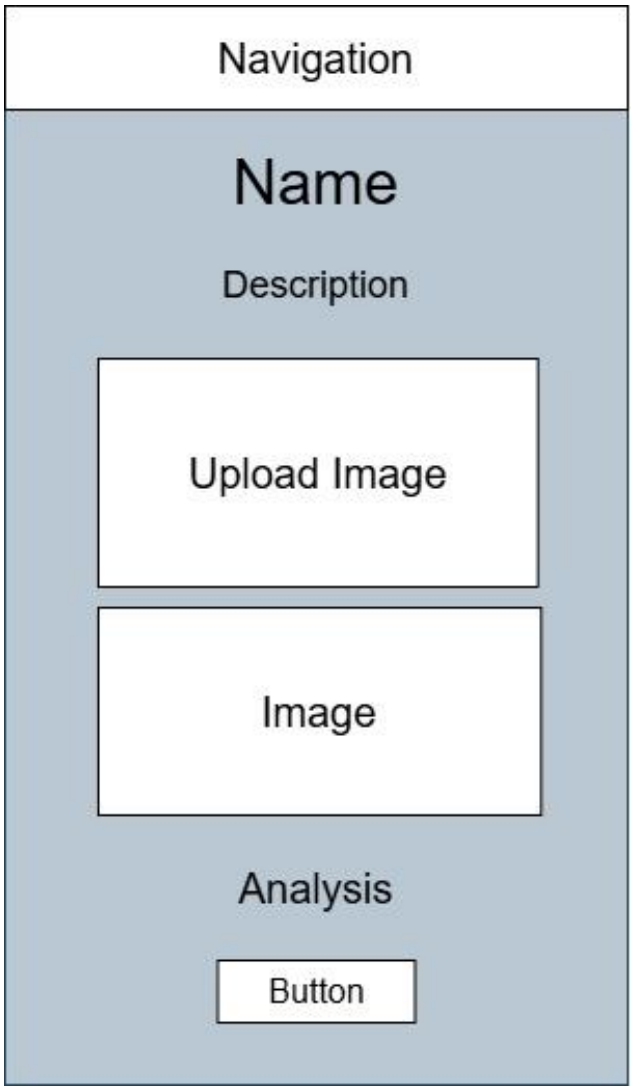

Рисунок 3.5 – Прототип інтерфейсу сторінки з отриманим аналізу

На сторінці відображаються всі попередні елементи. Крім того, при завантаженні зображення воно відображається на сторінці. Після отримання

текстового аналізу, воно розміщується між представленим зображенням та кнопкою.

На рисунку 3.6 зображений інтерфейс інформаційної сторінки, де знаходиться основна інформація про систему.

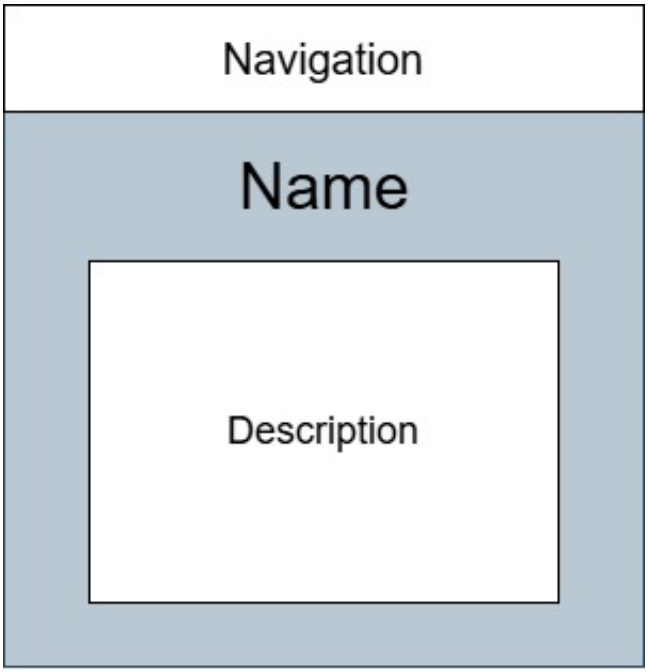

Рисунок 3.6 – Прототип інтерфейсу інформаційної сторінки

На інформаційній сторінці представлено текстовий блок, який детально описує модель, що лежить в основі роботи сервісу, а також модулі, що входять до складу моделі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Програмне та апаратне забезпечення

Для програмної реалізації системи був використаний ASUS TUF Gaming F15 FX506HC з процесором Intel® Core™ i5-11400H Processor 2.7 GHz та відеокартою NVIDIA® GeForce RTX™ 3050 Laptop GPU. Для реалізації тренування моделей була використана платформа Kaggle, яка надає середовище для розробки та безкоштовний GPU P100.

Для реалізації моделі була обрана мова програмування Python версії 3.12, яка завдяки широкій екосистемі бібліотек машинного навчання забезпечує зручну розробку нейронних мереж та допоміжних алгоритмів. Створення користувацького інтерфейсу здійснюється за допомогою мови розмітки HTML та каскадних таблиць стилів CSS. Взаємодія з серверною частиною реалізована за допомогою мови JavaScript.

4.2 Модуль класифікатора

Першим кроком в створенні модулю класифікатора є попереднє тренування нейроної мережі, ваги якої зберігаються для подальшого користування. Другим кроком є використання ваг моделі для інференсу.

4.2.1 Тренування модулю класифікатора

Для реалізації тренування модуля класифікатора ResNet-50 були використані такі бібліотеки, як PyTorch та Torchvision для створення й тренування нейронної мережі, бібліотека Datasets для доступу до датасету, а також допоміжні бібліотеки Scikit-Learn для оцінки моделі, Pillow для комфортної роботи з зображеннями, Random для розподілення сету та TQDM для відстежування прогресу навчання.

У лістингу 4.1 поданий програмний код ініціалізації моделі.

Лістинг 4.1 – Програмний код ініціалізації моделі ResNet-50

```
train_loader = DataLoader(dataset["train"], batch_size=32, shuffle=True, collate_fn=collate_fn)
val_loader = DataLoader(dataset["val"], batch_size=32, collate_fn=collate_fn)
criterion = nn.CrossEntropyLoss()
weights = ResNet50_Weights.IMAGENET1K_V2
model = resnet50(weights=weights)
num_features = model.fc.in_features
model.fc = nn.Linear(num_features, 3)
train_model(model, train_loader, device, lr=0.001, epochs=5)
```

Набір даних для оцінки точності моделі був поділений на дві частини, тренувальний та валідаційний сабсет (80% та 20%). За основу була взята друга версія ваг ResNet-50, натренована на наборі даних ImageNet, яка входить до бібліотеки Torchvision. Останній повнозв’язний шар було замінено на новий шар із трьома виходами. Швидкість навчання становила 0,001. Розмір батчу становив 32.

Програмний код попередньої обробки даних зображень поданий у лістингу 4.2.

Лістинг 4.2 – Програмний код реалізації обробки даних для ResNet-50

```
train_transforms = transforms.Compose([
    transforms.Resize((256, 256)),
    transforms.RandomResizedCrop(224),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(10),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
```

Продовження лістингу 4.2

```
])

val_transforms = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406],
                         std=[0.229, 0.224, 0.225])
])
```

У процесі попередньої обробки даних були застосовані методи аугментації зображень для підвищення узагальнювальної здатності моделі. Для тренувального набору використовувалися такі перетворення, як зміна розміру до 256x256 (відповідно до розмірів зображень з ImageNet), випадкове обрізання з масштабуванням, горизонтальне віддзеркалення та поворот на невеликий кут. Зображення було нормалізовано за допомогою середніх значень та стандартних відхилень. Для валідаційного набору застосовувались зміна розміру та нормалізація без додаткових випадкових змін.

Програмний код реалізації функції тренування поданий у лістингу 4.3.

Лістинг 4.3 – Програмний код функції тренування ResNet-50

```
def train_model(model, train_loader, device, lr, epochs):
    model = model.to(device)
    criterion = nn.CrossEntropyLoss()
    optimizer = torch.optim.Adam(model.parameters(),
lr=lr)
    model.train()

    for epoch in range(epochs):
        total_loss = 0
        progress_bar = tqdm(train_loader, desc=f"Epoch
{epoch+1}/{epochs}")
```

Продовження лістингу 4.3

```
            for images, labels in progress_bar:
                images, labels = images.to(device), 
labels.to(device)
                outputs = model(images)
                loss = criterion(outputs, labels)
                optimizer.zero_grad()
                loss.backward()
                optimizer.step()
                total_loss += loss.item()
                progress_bar.set_postfix({'loss': 
f'{loss.item():.4f}'})

            print(f"Epoch [{epoch+1}/{epochs}], Loss: 
{total_loss / len(train_loader):.4f}")
```

Оптимізація здійснювалась за допомогою алгоритму Adam. На початку модель переводиться в режим навчання. У циклі кожної епохи виконується поступова обробка усіх батчів зображень та міток із тренувального сету. Для кожного батчу дані передаються на пристрій GPU, потім модель здійснює передбачення, вичислюється функція втрат. Після цього градієнти обнуляються та виконується зворотне поширення помилки. Оптимізатор оновлює параметри моделі.

Для візуалізації прогресу використовується TQDM, який показує динамічний прогрес з поточними значеннями втрат. Після кожної епохи виводиться середня втрата за цю епоху для тренувального, а також для валідаційного сетів.

Навчання відбувалось протягом 5-ти епох, кінцева функція втрат становила 0,0871 для тренувального набору та 0,1228 для валідаційного набору.

Для оцінки моделі була використана функція, яка надає класифікаційний звіт по чотирьом параметрам: точність, влучність, повнота та f-1 міра. Оцінка моделі подана в таблиці 4.1.

Таблиця 4.1 – Оцінка моделі ResNet-50 після тренування

	precision	recall	f1-score	support
AI	0,90	1,00	0,95	672
FAKE	1,00	0,9	0,95	674
REAL	1,00	0,99	1,00	654
accuracy			0,96	2000

Повний код тренування та оцінки моделі поданий у додатку Б.

4.2.2 Інференс модулю класифікатора

Для інференсу модулю класифікатора була реалізована функція predict, яка повертає список класів та процент ймовірності відповідно. Код реалізації поданий у лістингу 4.4.

Лістинг 4.4 – Програмний код реалізації модулю класифікатора

```
class_names = ['Artificial', 'Deepfake', 'Real']
weights = ResNet50_Weights.IMAGENET1K_V2
model = resnet50(weights=weights).to(device)
num_features = model.fc.in_features
model.fc = torch.nn.Linear(num_features, 3).to(device)
model.load_state_dict(torch.load('models/model.pth',
map_location=device))
model.eval()
preprocess = weights.transforms()
def predict(image: Image.Image):
    image = Image.open(image_path).convert("RGB")
```

Продовження лістингу 4.4

```
        input_tensor = preprocess(image).unsqueeze(0)
        input_tensor = input_tensor.to(device)

        with torch.no_grad():
            logits = model(input_tensor)
            probs = F.softmax(logits, dim=1)
            percentages = [round(p.item() * 100, 1) for p in
probs[0]]

        return {class_names[i]: percentages[i] for i in
range(len(class_names))}
```

Для модуля класифікатора використовується попередньо навчена модель ResNet-50. Завантаження моделі здійснюється з попередньо збережених ваг за допомогою функції torch.load, зображення трансформуються через метод weights.transforms(). Після ініціалізації моделі, вона переводиться в режим оцінки. Вхідне зображення в форматі Image піддається обробці моделі.

Результати передаються через функцію softmax з torch.nn.functional, що дозволяє отримати ймовірності належності до кожного класу у відсотках. На виході функція повертає словник із назвами класів та відповідними відсотками впевненості моделі. Повний код модулю класифікатора поданий у додатку Б.

4.3 Модуль аналізу ключових точок

Першим кроком в створенні модулю аналізу ключових точок є попереднє тренування нейронної мережі, ваги якої зберігаються для подальшого користування. Другим кроком є використання ваг моделі для інференсу.

4.3.1 Тренування модулю аналізу ключових точок

Для навчання модулю аналізу ключових точок Mask R-CNN використовались бібліотеки PyTorch, Torchvision для побудови та навчання нейронної мережі, Pycocotools для обробки анотацій датасету, а також допоміжні інструменти, такі як Pillow та TQDM для роботи з зображеннями та відображення ходу навчання. Програмний код ініціалізації моделі поданий у лістингу 4.5.

Лістинг 4.5 – Програмний код ініціалізації моделі Mask R-CNN

```
dataset = COCOLimbSegmentationDataset(
    root="/kaggle/input/coco-2017-
dataset/coco2017/train2017",
    annFile="/kaggle/input/coco-2017-
dataset/coco2017/annotations/person_keypoints_train2017.json",
    transforms=get_transform())

num_samples = int(0.1 * len(dataset))
indices = random.sample(range(len(dataset)), num_samples)
subset_dataset = Subset(dataset, indices)
data_loader = DataLoader(subset_dataset,batch_size=4,
shuffle=True,collate_fn=lambda x: tuple(zip(*x)))

def segmentation_model(num_classes):
    backbone = resnet_fpn_backbone ('resnet50',
weights_only = True)
    model = MaskRCNN(backbone, num_classes=2)
    return model

model = segmentation_model(2).to(device)
```

Бібліотека Torchvision надає реалізації як моделі Mask R-CNN, так і натренованого бекбона на основі ResNet-50 з FPN. Для навчання

використовується підмножина набору даних COCO 2017 Keypoints, до якого створено власний клас COCOLimbSegmentationDataset, що зчитує зображення та відповідні маски сегментації лише для категорії "person". Для нейронної мережі визначено всього два класи: перший відповідає за кінцівки, другий за об’єкти, що залишились. Повний код реалізації класу поданий у додатку Б. Для експерименту використано лише 10% зображень з датасету, що дозволяє пришвидшити тренування на обмежених ресурсах.

У лістингу 4.6 подана програмна реалізація тренування.

Лістинг 4.6 – Програмний код реалізації тренування Mask R-CNN

```
    optimizer = torch.optim.AdamW(params, lr=1e-3, weight_decay=0.01)
    for epoch in range(epochs):
        model.train()
        epoch_loss = 0.0
        with tqdm(data_loader, desc=f"Epoch {epoch+1}/{epochs}") as pbar:
            for images, targets in pbar:
                images = list(img.to(device) for img in images)
                targets = [{k: v.to(device) for k, v in t.items()} for t in targets]
                loss_dict = model(images, targets)
                losses = sum(loss for loss in loss_dict.values())
                optimizer.zero_grad()
                losses.backward()
                optimizer.step()
                loss_value = losses.item()
                epoch_loss += loss_value
                pbar.set_postfix(loss=loss_value)
        avg_loss = epoch_loss / len(data_loader)
        print(f"[Epoch {epoch+1}] Average Loss: {avg_loss:.4f}")
```

Для навчання обрано оптимізатор AdamW із початковою швидкістю навчання 10^{-5} та регуляризацією через weight_decay. Модель навчалась протягом 10-ти епох, після кожної епохи навчання обчислювалась загальна втрата, яка включає помилки класифікації, регресії координат боксів та сегментації масок. Функція втрат після п’яти епох становила 0,2814 для тренувального сету та 0,4915 для валідаційного сету.

4.3.2 Інференс модулю аналізу ключових точок

Для інференсу модулю аналізу ключових точок були створені дві функції. Після створення сегментаційної моделі Mask R-CNN, яка була попередньо натренована, завантажуються збережені ваги та модель переводиться в режим оцінки. Перша функція, count_limbs, подана у лістингу 4.7.

Лістинг 4.7 – Програмний код реалізації функції count_limbs

```
    def count_limbs(image: Image.Image, threshold=confidence)
-> int:
        image_tensor = F.to_tensor(image).unsqueeze(0)

        with torch.no_grad():
            predictions = model(image_tensor)[0]
        limb_count = sum(
            1 for i in range(len(predictions["scores"]))
            if predictions["scores"][i].item() >= threshold and
predictions["labels"][i].item() == 1)

        return limb_count
```

Функція призначена для визначення кількості кінцівок (рук і ніг). На вході приймається безпосередньо зображення у форматі Image, яке

оброблюється моделлю для передбачення об’єктів. Рахуються лише ті об’єктів, які відносяться до класу кінцівок та мають ймовірність вище за 0,5.

У лістингу 4.8 подана програмна реалізація функції visualize_limbs.

Лістинг 4.8 – Програмний код реалізації функції visualize_limbs

```
    def visualize_limbs(image: Image.Image, threshold=confidence) -> Image.Image:
        image_tensor = F.to_tensor(image).unsqueeze(0)

        with torch.no_grad():
            predictions = model(image_tensor)[0]
        valid_masks = [predictions["masks"][i] > 0.5
            for i in range(len(predictions["scores"]))
            if predictions["scores"][i].item() >= threshold and predictions["labels"][i].item() == 1]
        image_np = np.array(image).copy()
        for mask in valid_masks:

            mask_np = mask.squeeze().cpu().numpy()
            contours = plt.contour(mask_np, levels=[0.5])
            for collection in contours.collections:
                for path in collection.get_paths():
                    coords = path.vertices
                    draw = ImageDraw.Draw(Image.fromarray(image_np))
                    draw.line([tuple(p) for p in coords], fill=(255, 0, 0), width=2)
            plt.clf()

        return Image.fromarray(image_np)
```

Функція візуалізації кінцівок працює за тим же принципом, що й попередня: на вхід подається зображення, а модель визначає об’єкти. Для

кожної знайденої маски об’єкта визначаються контури за допомогою matplotlib, після чого вони малюються червоним кольором на копії оригінального зображення. У результаті функція повертає нове зображення з візуально позначеними межами знайдених кінцівок.

Повний програмний код файлу з використанням моделі сегментації подано в додатку Б.

4.4 Модуль аналізу освітлення

Для модулю аналізу освітлення була створена окрема функція detect_light та використана OpenCV – бібліотека комп’ютерного зору, яка призначена для обробки та аналізу зображень та базується на класичних алгоритмах без застосування методів штучного інтелекту. Програмний код реалізації функції поданий у лістингу 4.9.

Лістинг 4.9 – Програмний код реалізації функції detect_light

```
def detect_light(image_input: Image.Image) -> Image.Image:
    image_np = np.array(image_input).astype(np.uint8)
    image = cv2.cvtColor(image_np, cv2.COLOR_RGB2BGR)

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (7, 7), 0)

    inverted = cv2.bitwise_not(blurred)
     _, shadow_mask = cv2.threshold(inverted, 50, 255,
cv2.THRESH_BINARY)

        shadow_boost = cv2.bitwise_and(image, image,
mask=shadow_mask)
    enhanced = cv2.addWeighted(image, 1.0, shadow_boost, -
0.5, 0)
```

Продовження лістингу 4.9

```
            return    Image.fromarray(cv2.cvtColor(enhanced,
cv2.COLOR_BGR2RGB))
```

Функція на вході приймає зображення в форматі Image. За допомогою cvtColor зображення перетворюється у відтінки сірого. Далі до нього застосовується згладжування через GaussianBlur, щоб зменшити шум. Зображення інвертується функцією bitwise_not, після threshold створює маску тіней. Ця маска накладається на оригінальне зображення за допомогою bitwise_and, утворюючи підсилене зображення тіней. У кінці функція addWeighted виконує зважене об’єднання оригінального зображення з тіньовими ділянками з негативною вагою, що дозволяє приглушити тіні. На виході повертається зображення з підсиленим світлом та тінями у форматі Image. Повний програмний код реалізації модулю освітлення поданий у додатку Б.

4.5 Модуль великої мовної моделі

Для останнього модулю великої мовної моделі були використані бібліотеки Google-GenerativeAi для підключення та використання моделі Gemini та Pillow для роботи з зображеннями. Зв’язок з моделлю Gemini 2.5 Pro представлений у вигляді ключа, який генерується за допомогою сервісу Google Cloud [76].

Модуль складається з двох частин: функції написання промпту та функції відправки до LLM.

Програмний код реалізації створення промпту поданий у лістингу 4.10.

Лістинг 4.10 – Програмний код реалізації створення промпту

```
def build_prompt(probabilities, limbs_count) -> str:
```

Продовження лістингу 4.10

```
        return f"""
    There is probabilities of the image being a deepfake, real or artificial, in percentage:
    {probabilities}

    Analyze the image based on the following parameters:
    1. Facial features: Look for inconsistencies in facial features such as eyes, mouth, and nose.
    2. Body: Check for unnatural body poses or extra limbs that do not match the body's natural anatomy.
    3. Lighting: Ensure that the lighting and shadows on the face matches the rest of the image.

    You are provided with three images:
    - Original image
    - Image with visualized limbs. The limbs count is {limbs_count}
    - Image with enhanced lighting and shadows

    Think step by step. Provide a detailed and structured analysis using all images and the probabilities. After that, write a clear and concise conclusion in the following format:
    Answer the question: Does this image look like a deepfake or not?
    Final answer: [Yes, it looks like a deepfake/No, it doesn't look like a deepfake], because [brief explaining].
    """
```

Цей промпт дає моделі чіткі інструкції для аналізу зображення на дипфейк, використовуючи ймовірності класів та три варіанти зображень (оригінал, з виділеними кінцівками, з підсиленим освітленням). Промпт пропонує звернути увагу на такі ключові ознаки, як обличчя, тіло,

фон та освітлення. Модель має міркувати послідовно, дати детальний аналіз і в кінці чітко відповісти, чи є зображення дипфейком, надаючи коротке пояснення, яке потім буде передано користувачеві.

Програмний код реалізації функції доступу до моделі поданий у лістингу 4.11.

Лістинг 4.11 – Програмний код реалізації доступу до Gemini

```
    def analyze_image(original: Image.Image, pose: Image.Image, shadow: Image.Image, probabilities: dict, limbs_count) -> str:
        prompt = build_prompt(probabilities, limbs_count)
        model = genai.GenerativeModel('gemini-2.5-pro-preview-06-05')

        response = model.generate_content([prompt, original, pose, shadow], stream=False)
        match = re.search(r"Final answer:\s*(.*)", response.text)

        if match:
            extracted_text = match.group(1).strip()
```

Продовження лістингу 4.11

```
            cleaned_text = extracted_text.replace('**', '')
            return cleaned_text
        else:
            return response.text
```

Функція приймає три зображення та словник з ймовірностями класифікації. Спершу вона створює запит за допомогою функції build_prompt, яка містить повний текст промпту. Далі створюється модель gemini-2.5-pro та передається запит разом з трьома зображеннями. Після завершення обробки модель повертає текстову відповідь, що містить структурований аналіз та фінальний висновок, чи є зображення дипфейком.

Текстова відповідь перед поверненням до серверу проходить етап регуляризації: після знаходження строки «Final answer:», усі попередні символи видаляються, скорочуючи відповідь, також зі строки видаляються зайві символи. Повний програмний код реалізації модулю GPT поданий у додатку Б.

4.6 Сервер

Для реалізації серверу була використана бібліотека Flask, що є фреймворком для створення веб-додатків, який надає простий та гнучкий спосіб для запуску веб-серверів, обробки HTTP-запитів та створення API. Код реалізації маршруту обробника запиту поданий у лістингу 4.12.

Лістинг 4.12 – Програмний код маршруту /analyze

```
@app.route('/analyze', methods=['POST'])
def analyze():
    if 'image' not in request.files:
        return jsonify({'error': 'No image uploaded'}), 400

    file = request.files['image']

          with    tempfile.NamedTemporaryFile(delete=False,
suffix=".jpg") as temp_file:
        image_path = temp_file.name
        file.save(image_path)

    try:
        print("STEP 1: Opening original image")
        original = Image.open(image_path).convert("RGB")
        print("original:", type(original))

        print("STEP 2: Running prediction")
```

Продовження лістингу 4.12

```
            probabilities = predict(original)
            print("probabilities:", probabilities)

            print("STEP 3: Counting limbs")
            limbs_count = count_limbs(original)
            print("limbs_count:", limbs_count)

            print("STEP 4: Visualizing limbs")
            visualized_limbs = visualize_limbs(original)
            print("visualized_limbs:", type(visualized_limbs))

            print("STEP 5: Detecting shadows")
            shadow_image = detect_light(original)
            print("shadow_image:", type(shadow_image))

            print("STEP 6: Calling analyze_image()")
                    result_text   =   analyze_image(original,
visualized_limbs, shadow_image, probabilities, limbs_count)

            return jsonify({'analysis': result_text})

        except Exception as e:
            print("Exception:", e)
            return jsonify({'error': str(e)}), 500

        finally:
            os.remove(image_path)
```

При зверненні до маршруту /analyze методом POST, сервер очікує отримання зображення у форматі файлу. Після завантаження зображення воно тимчасово зберігається на диск, що дозволяє працювати з ним за допомогою зовнішніх бібліотек OpenCV та PIL.

Далі викликається функція з модуля класифікатора predict, яка визначає ймовірності належності зображення до одного з класів. Після цього зображення передається до модулю аналізу ключових точок двома функціями (count_limbs та visualized_limbs) та виконується функція з модулю аналізу освітлення detect_light, яка оброблює освітлення на зображенні. Текстові та візуальні дані передаються в модуль GPT для аналізу оригінального зображення завдяки функції analyze_image. Результатом аналізу є згенерований текст, який повертається користувачу у форматі JSON. У випадку помилки (некоректне зображення або помилка в обробці) сервер повертає відповідне повідомлення. Після завершення обробки тимчасовий файл зображення видаляється для запобігання накопиченню файлів. Кожний етап супроводжується виведенням вихідних даних функції в консоль, тому, при наявній помилці, консоль попередить, в якому з модулів є помилка.

Повний програмний код реалізації серверу поданий у додатку Б.

4.7 Користувацький інтерфейс

Інтерфейс складається з трьох файлів: index.html (головна сторінка), about.html (сторінка з описом системи), styles.css, (визначає дизайн інтерфейсу), та script.js (реалізує логіку оброблення подій, завантаження зображення, відображення зображення і надсилання запиту на сервер). У лістингу 4.13 поданий код реалізації доступу до серверу з файлу script.js

Лістинг 4.13 – Програмний код реалізації доступу до серверу

```
try
{
const response = await fetch('http://localhost:5000/analyze', {
    method: 'POST',
```

Продовження лістингу 4.13

```
        body: formData,
    });
    const data = await response.json();
    resultDiv.style.display = 'block';
    if (data.analysis) {
        resultContent.textContent = data.analysis;
    } else if (data.error) {
        resultContent.textContent = 'Error: ' + data.error;
    } else {
        resultContent.textContent = 'Unexpected response from
server';
    }
    } catch (error) {
    resultDiv.style.display = 'block';
    resultContent.textContent = 'Request failed: ' +
error.message;
    } finally {
    analyzeBtn.disabled = false;
    analyzeBtn.textContent = 'Analyze Image';
    }
```

Цей фрагмент коду відповідає за надсилання зображення на сервер для аналізу та обробку відповіді від нього. Він використовується у функції uploadImage() після того, як користувач завантажив зображення та натиснув кнопку. Усередині блоку try виконується асинхронний HTTP-запит до локального сервера методом POST. У тілі запиту передається об’єкт formData, який містить вибране користувачем зображення.

Після отримання відповіді функція намагається розпарсити текст як JSON. У полі результатів відображаються текст аналізу (якщо присутнє поле analysis), або повідомлення про помилку. У блоці finally кнопка аналізу знову активується, незалежно від того, запит завершився успішно чи з помилкою.

У лістингу 4.14 подана частина коду програмної реалізації index.html, головної сторінки системи. Повний код файлу приведений в додатку Б.

Лістинг 4.14 – Програмний код реалізації головної сторінки index.html

```
    <div class="container">
      <div class="header">
        <h1>Common Sense Image Analysis</h1>
        <p class="subtitle">
          Upload an image for deepfake detection and
authenticity analysis
        </p>
      </div>
      <div class="upload-area" id="uploadArea">
        <div class="upload-icon">📁</div>
        <p class="upload-text">Drag & drop your image</p>
        <p class="upload-hint">or click to browse files (JPG,
PNG)</p>
        <input type="file" id="imageInput" accept="image/*"
/>
      </div>
      <img id="preview" />
      <button class="btn btn-primary"
onclick="uploadImage()" id="analyzeBtn">
        Analyze Image
      </button>
      <div id="result">
        <div class="result-header">
          <div class="result-icon">🔍</div>
          <div class="result-title">Analysis Result</div>
        </div>
        <div class="result-content"
id="resultContent"></div>
      </div>
    </div>
```

Основним контейнером виступає , який містить усі елементи сторінки. У верхній частині розміщений заголовок <h1> з назвою системи, а під ним підзаголовок <p class="subtitle">, який пояснює користувачу призначення сторінки. Далі йде область завантаження файлу , яка оформлена так, щоб користувач міг перетягнути зображення мишкою або клікнути для вибору файлу через файловий діалог. Після вибору зображення воно відображається у тегу .

Нижче розташована кнопка, яка запускає функцію uploadImage(), ініціюючи відправлення обраного зображення на сервер для аналізу.

Під кнопкою є блок з результатом аналізу , який містить заголовок та область для виведення текстового опису результатів. Спочатку цей блок прихований і відображається лише після отримання відповіді від сервера.

Код реалізації файлу about.html поданий у лістингу 4.15.

Лістинг 4.15 – Програмна реалізація сторінки about.html

```
  <div class="container">
    <div class="header">
      <h1>About</h1>
      <p class="subtitle2">
        This system is an experimental tool designed to assist users in detecting deepfake images through the application of commonsense reasoning, based common sense reasoning.
      </p> <ul class="subtitle2"> The model consists of several modules:
            <li>Keypoints analysis module</li>
            <li>Lighting analysis module</li>
            <li>Classifier module</li>
            <li>GPT module</li>
        </ul>
```

Продовження лістингу 4.15

```
        <p class="subtitle2">
          It helps to explain the results of the analysis and
provide insights into the authenticity of the image.
        </p>
      </div>
```

Інформаційна сторінка містить в собі короткий опис призначення про системи, а також модулі, з яких складається модель, у вигляді переліку.

Код реалізації панелі поданий у лістингу 4.16.

Лістинг 4.16 – Програмна реалізація навігаційної панелі

```
<nav class="navbar">
    <div class="nav-links">
    <a href="index.html" class="nav-link">Home</a>
    <a href="about.html" class="nav-link active">About</a>
    </div>
</nav>
```

Для зручності керування була створена навігаційна панель, яка складається з двох кнопок для переходу між сторінка. Активна кнопка виділяється в залежності від того, на якій сторінці знаходиться користувач.

Скріншоти повної реалізації системи та вигляду інтерфейсу подані в додатку В.

4.8 Результати виконання програми

Для демонстрації результатів роботи системи було обрано три різні зображення, які відрізняються типом генерації та візуальними ознаками. Усі скріншоти роботи системи, а також користувацького інтерфейсу, подані в додатку В.

Перше зображення було згенероване моделлю DALL-E [46] за допомогою запиту «Generate a deepfake image». Людина має складну позу, яку складно згенерувати реалістино. Дипфейк візуально містить такі характерні ознаки, як нереалістичне освітлення, однорідну текстуру та зайву кінцівку (рисунок 4.1).

Рисунок 4.1 – Зображення, згенероване DALL·E [46]

Модуль класифікатора виявив, що це дипфейк з ймовірністю 88,6 відсотків, модуль аналізу ключових точок виявив 5 кінцівок. Велика мовна модель виявила, що матеріал є дипфейком через причини, які подані в контексті, а також через однорідність текстур на зображенні. Повна відповідь системи подана в Додатку В.

Друге зображення взято з набору даних «DeepFakeFace» [77] та анотоване як дипфейк (рисунок 3.2). Візуально людина на зображенні має спотворені риси обличчя. Повна відповідь системи подана в Додатку В.

Рисунок 4.2 – Дипфейк-зображення з набору даних «DeepFakeFace» [77]

Третє зображення є справжнім та взяте із набору даних «AI vs Deepfake vs Real» [62], який використовувався під час тренування класифікатора (рисунок 4.3).

Рисунок 4.3 – Справжнє зображення з датасету «AI vs Deepfake vs Real» [62]

Повна відповідь системи подана в Додатку В.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено систему для обробки зображень та виявлення дипфейків із застосуванням технологій міркування здорового глузду. Основною метою розробки стало створення інструменту, здатного не лише класифікувати зображення як справжні або підроблені, а й пояснювати причини прийнятого рішення, спираючись на аналіз контексту, логіки та візуальних ознак.

У першій частині роботи було здійснено аналіз предметної галузі дипфейків. Розглянуто історію появи дипфейків, основні типи фальсифікацій, а також потенційні загрози, які вони становлять для суспільства, політики, медіа та безпеки. Окрему увагу приділено відомим випадкам застосування дипфейків та можливому позитивному потенціалу цієї технології, зокрема у творчій та розважальній галузі.

У другій частині було проведено аналіз генерації дипфейків. Розглянуто сучасні моделі генерації, а також типові візуальні артефакти, які можуть свідчити про штучність зображення. Проаналізовано стратегії, які застосовують люди під час спроб виявити дипфейки, та окреслені наявні проблеми. Важливим розділом стало введення поняття міркування здорового глузду в моделях штучного інтелекту. Також розглянуто фреймворки та бази даних, які дозволяють інтегрувати логічне міркування в розроблені системи.

У третій частині було розроблено власну архітектуру системи, яка складається з кількох функціональних модулів. Зокрема, описано модуль класифікації зображень, який дозволяє визначати клас зображення (штучне, дипфейк або реальне), модуль аналізу ключових точок для оцінки структури тіла людини, модуль аналізу освітлення для виявлення аномалій світла, а також інтеграцію великої мовної моделі для генерації текстового пояснення. Окрім цього, було спроектовано зручний користувацький інтерфейс, який

дозволяє завантажувати зображення, переглядати результати аналізу та за необхідністю отримати інформацію про систему.

У четвертій частині детально описано програмну модулів системи, серверної частини, що поєднує всі модулі, та користувацького інтерфейсу. Наведено лістинги коду для кожного з модулів: класифікації, аналізу ключових точок, освітлення та підключення до великої мовної моделі. Серверна частина забезпечує обробку запитів, послідовний виклик модулів та передачу результатів до інтерфейсу. Також подано реалізацію користувацького інтерфейсу у веб-браузері. Показано, як система функціонує як єдине ціле та представлено результати її роботи на прикладних зображеннях.

Результати роботи системи показали, що система функціонує вправно. Було проаналізовано три фотографії та надані результати обробки системою, у кожному випадку класифікатор правильно визначив тип зображення та супроводив його поясненням, згенерованим великою мовною моделлю на основі даних, отриманих від інших модулів системи.

У майбутньому система може розвиватись, розширивши або змінивши архітектури модулів на більш сучасні, ефективні та адаптовані до новітніх досліджень у галузі штучного інтелекту. Зокрема, можливе впровадження більш потужних нейромереж та великої мовної моделі для глибшого розуміння контексту або підключення зовнішніх джерел знань для покращення якості міркувань.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Eberl A., Kühn J., Wolbring T. Frontiers | Using deepfakes for experiments in the social sciences - A pilot study. *Frontiers*. URL: https://www.frontiersin.org/journals/sociology/articles/10.3389/fsoc.2022.907199/full (дата звернення: 04.06.2025).

2. AI-Assisted fake porn is here and we're all fucked. *VICE*. URL: https://www.vice.com/en/article/gal-gadot-fake-ai-porn/ (дата звернення: 12.05.2025).

3. TensorFlow. *TensorFlow*. URL: https://www.tensorflow.org/ (дата звернення: 12.05.2025).

4. PyTorch foundation. *PyTorch*. URL: https://pytorch.org/ (дата звернення: 12.05.2025).

5. The YouTube Team. How we're helping creators disclose altered or synthetic content. *blog.youtube*. URL: https://blog.youtube/news-and-events/disclosing-ai-generated-content/ (дата звернення: 12.05.2025).

6. Partnership on AI - home - partnership on AI. *Partnership on AI*. URL: https://partnershiponai.org/ (дата звернення: 12.05.2025).

7. Welcome to the artificial intelligence incident database. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/ (дата звернення: 12.05.2025).

8. Incident 39: deepfake obama introduction of deepfakes. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/39/ (дата звернення: 12.05.2025).

9. BuzzFeedVideo. You won't believe what obama says in this video! 😉, 2018. *YouTube*.
URL: https://www.youtube.com/watch?v=cQ54GDm1eL0 (дата звернення: 05.06.2025).

10. Incident 198: deepfake video of ukrainian president yielding to russia posted on ukrainian websites and social media. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/198/ (дата звернення: 12.05.2025).

11. Prof. hany farid debunks broadcast of ukrainian president zelensky supposedly using expletives to describe oval office encounter. *UC Berkeley School of Information*. URL: https://www.ischool.berkeley.edu/news/2025/prof-hany-farid-debunks-broadcast-ukrainian-president-zelensky-supposedly-using (дата звернення: 12.05.2025).

12. Incident 626: social media scammers used deepfakes of taylor swift and several other celebrities in fraudulent le creuset cookware giveaways. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/626/ (дата звернення: 12.05.2025).

13. Incident 634: alleged deepfake CFO scam reportedly costs multinational engineering firm arup $25 million. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/634/ (дата звернення: 12.05.2025).

14. Incident 510: viral image of pope francis in a puffer jacket revealed to be ai-generated. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/510/ (дата звернення: 12.05.2025).

15. Vatican News. Pope Francis urges ethical use of artificial intelligence - Vatican News. *News from the Vatican - News about the Church - Vatican News*. URL: https://www.vaticannews.va/en/pope/news/2023-03/pope-francis-minerva-dialogues-technology-artificial-intelligenc.html (дата звернення: 12.05.2025).

16. Fake photos of Pope Francis in a puffer jacket go viral, highlighting the power and peril of AI. *CBS News | Breaking news, top stories & today's latest headlines*. URL: https://www.cbsnews.com/news/pope-francis-puffer-jacket-fake-photos-deepfake-power-peril-of-ai/ (дата звернення: 05.06.2025).

17. Incident 499: parody AI images of donald trump being arrested reposted as misinformation. *Welcome to the Artificial Intelligence Incident Database*. URL: https://incidentdatabase.ai/cite/499/ (дата звернення: 12.05.2025).

18. Midjourney. URL: https://www.midjourney.com/ (дата звернення: 13.05.2025).

19. RealQuotesAI on Instagram: "Pope Leo draws standing ovation after giving a profound speech ". *Instagram*. URL: https://www.instagram.com/reel/DJaRd3RxFaf/?igsh=MXZ2cnRzd3Fpdmtj (дата звернення: 12.05.2025).

20. Shoard C. Peter Cushing is dead. Rogue One's resurrection is a digital indignity | Catherine Shoard. *the Guardian*. URL: https://www.theguardian.com/commentisfree/2016/dec/21/peter-cushing-rogue-one-resurrection-cgi (дата звернення: 12.05.2025).

21. ABC News. A genuinely believable CGI actor? It won't be long. *ABC (Australian Broadcasting Corporation)*. URL: https://www.abc.net.au/news/2017-01-19/a-genuinely-believable-cgi-actor-it-wont-be-long/8193454 (дата звернення: 05.06.2025).

22. The Irishman: how we made the actors decades younger. *BBC Home - Breaking News, World News, US News, Sports, Business, Innovation, Climate, Culture, Travel, Video & Audio*. URL: https://www.bbc.com/news/av/technology-51360643 (дата звернення: 12.05.2025).

23. Inside the revolutionary visual effects process of the irishman. *The Film Stage - Your Spotlight On Cinema*. URL: https://thefilmstage.com/inside-the-revolutionary-visual-effects-process-of-the-irishman/ (дата звернення: 05.06.2025).

24. Etienne H. The future of online trust (and why Deepfake is advancing it). *AI and ethics*. 2021. URL: https://doi.org/10.1007/s43681-021-00072-1 (дата звернення: 13.05.2025).

25. Vaccari C., Chadwick A. Deepfakes and disinformation: exploring the impact of synthetic political video on deception, uncertainty, and trust in news. *Social media + society*. 2020. Т. 6, № 1. С. 205630512090340. URL: https://doi.org/10.1177/2056305120903408 (дата звернення: 13.05.2025).

26. Decopy AI. *Decopy AI: Your All-in-One Writing Solution*. URL: https://decopy.ai/ (дата звернення: 12.05.2025).

27. Fake image detector | fake image detector online | fotoforensics | error level analysis. *Fake Image Detector | Fake Image Detector Online | FotoForensics | Error Level Analysis*. URL: https://www.fakeimagedetector.com/ (дата звернення: 12.05.2025).

28. AI generated image & deepfake detector - faceonlive : on-premises ID verification & biometrics solution provider. *FaceOnLive : On-Premises ID Verification & Biometrics Solution Provider*. URL: https://faceonlive.com/deepfake-detector/ (дата звернення: 05.06.2025).

29. 28. LeCun Y., Bengio Y., Hinton G. Deep learning. *Nature*. 2015. Т. 521, № 7553. С. 436–444. URL: https://doi.org/10.1038/nature14539 (дата звернення: 13.05.2025).

30. Deep residual learning for image recognition / K. He та ін. *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, м. Las Vegas, NV, USA, 27–30 черв. 2016 р. 2016. URL: https://doi.org/10.1109/cvpr.2016.90 (дата звернення: 13.05.2025).

31. Singh G., Guleria K. A dense residual network-50 model for identification of real and fake images. *2024 second international conference computational and characterization techniques in engineering & sciences (IC3TES)*, м. Lucknow, India, 15–16 листоп. 2024 р. 2024. С. 1–5. URL: https://doi.org/10.1109/ic3tes62412.2024.10877597 (дата звернення: 13.05.2025).

32. CIFAKE: real and ai-generated synthetic images. *Kaggle*. URL: https://www.kaggle.com/datasets/birdy654/cifake-real-and-ai-generated-synthetic-images (дата звернення: 13.05.2025).

33. Chollet F. Xception: deep learning with depthwise separable convolutions. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, м. Honolulu, HI, 21–26 лип. 2017 р. 2017. URL: https://doi.org/10.1109/cvpr.2017.195 (дата звернення: 13.05.2025).

34. Chollet F. Xception: deep learning with depthwise separable convolutions. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, м. Honolulu, HI, 21–26 лип. 2017 р. 2017. URL: https://doi.org/10.1109/cvpr.2017.195 (дата звернення: 13.05.2025).

35. GitHub - ondyari/FaceForensics: Github of the FaceForensics dataset. *GitHub*. URL: https://github.com/ondyari/FaceForensics (дата звернення: 12.05.2025).

36. An image is worth 16x16 words: transformers for image recognition at scale. / A. Dosovitskiy та ін. *9th international conference on learning representations* : матеріали Міжнар. наук. конф., 3–7 трав. 2021 р. 2021.

37. Deepfake image detection using vision transformer models / B. Ghita та ін. *2024 IEEE international black sea conference on communications and networking (blackseacom)*, м. Tbilisi, Georgia, 24–27 черв. 2024 р. 2024. С. 332–335.
URL: https://doi.org/10.1109/blackseacom61746.2024.10646310 (дата звернення: 13.05.2025).

38. Deepfake and real images. *Kaggle*. URL: https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images (дата звернення: 13.05.2025).

39. Xu Y., Yayilgan S. Y. When handcrafted features and deep features meet mismatched training and test sets for deepfake detection. 2022.

40. Common sense reasoning for deepfake detection / Y. Zhang та ін. *Computer vision – ECCV 2024* / ред.: A. Leonardis та ін. Cham, 2025. Т. 15146. URL: https://doi.org/10.1007/978-3-031-73223-2_22.

41. Generative adversarial nets / I. J. Goodfellow та ін. *Advances in neural information processing systems* : матеріали Міжнар. наук. конф., м. Montreal, 8 груд. 2014 р. Montreal, 2014.

42. GAN Machine Learning: putting fictitious faces into practice. *DataScientest*. URL: https://datascientest.com/en/gan-machine-learning-putting-fictitious-faces-into-practice (дата звернення: 05.06.2025).

43. Kingma D. P., Welling M. Auto-Encoding variational bayes. *Conference proceedings: papers accepted to the international conference on learning representations* : матеріали Міжнар. наук. конф., м. Banff, 14 квіт. 2014 р. 2014.

44. GAN Machine Learning: putting fictitious faces into practice. *DataScientest*. URL: https://datascientest.com/en/gan-machine-learning-putting-fictitious-faces-into-practice (дата звернення: 05.06.2025).

45. Deep unsupervised learning using nonequilibrium thermodynamics / J. Sohl-Dickstein та ін. *32nd international conference on machine learning* : матеріали Міжнар. наук. конф., м. Lille, 6 лип. 2015 р. 2015.

46. DALL·E: creating images from text. URL: https://openai.com/index/dall-e/ (дата звернення: 13.05.2025).

47. Improving diffusion models as an alternative to gans, part 2 | NVIDIA technical blog. *NVIDIA Technical Blog*. URL: https://developer.nvidia.com/blog/improving-diffusion-models-as-an-alternative-to-gans-part-2/ (дата звернення: 05.06.2025).

48. Characterizing photorealism and artifacts in diffusion model-generated images / N. Kamali та ін. *CHI 2025: CHI conference on human factors in computing systems*, м. Yokohama Japan. New York, NY, USA, 2025. С. 1–26. URL: https://doi.org/10.1145/3706598.3713962 (дата звернення: 06.06.2025).

49. Deepfake media forensics: status and future challenges / I. Amerini та ін. *Journal of imaging*. 2025. Т. 11, № 3. С. 73. URL: https://doi.org/10.3390/jimaging11030073 (дата звернення: 06.06.2025).

50. Boudníková O., Kleisner K. AI-generated faces show lower morphological diversity then real faces do. *Anthropological review*. 2024. Т. 87, № 1. С. 81–91. URL: https://doi.org/10.18778/1898-6773.87.1.06 (дата звернення: 06.06.2025).

51. Bray S. D., Johnson S. D., Kleinberg B. Testing human ability to detect 'deepfake' images of human faces. *Journal of cybersecurity*. 2023. Т. 9, № 1. URL: https://doi.org/10.1093/cybsec/tyad011 (дата звернення: 06.06.2025).

52. Deepfake detection by human crowds, machines, and machine-informed crowds / M. Groh та ін. *Proceedings of the national academy of sciences*. 2021. Т. 119, № 1. URL: https://doi.org/10.1073/pnas.2110013119 (дата звернення: 06.06.2025).

53. Caporusso N., Zhang K., Carlson G. Using eye-tracking to study the authenticity of images produced by generative adversarial networks. *2020 international conference on electrical, communication, and computer engineering (ICECCE)*, м. Istanbul, Turkey, 12–13 черв. 2020 р. 2020. URL: https://doi.org/10.1109/icecce49384.2020.9179472 (дата звернення: 06.06.2025).

54. Analysis of human perception in distinguishing real and ai-generated faces: an eye-tracking based study / J. Huang та ін. Ithaca : Cornell University Library, 2024. 9 с. (Препринт. University of Notre Dame ; arXiv:2409.15498v1).

55. Attention is all you need / A. Vaswani та ін. *Proceedings of the 31st international conference on neural information processing systems (neurips 2017)* : матеріали Міжнар. наук. конф., м. Long Beach, 4–9 груд. 2017 р. Red Hook, NY, 2017.

56. Davis E., Marcus G. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*. 2015. Т. 58,

№ 9. С. 92–103. URL: https://doi.org/10.1145/2701413 (дата звернення: 06.06.2025).

57. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models / J. Li та ін. arXiv, 2023. 13 с. (Препринт. arXiv:2301.12597). URL: https://doi.org/10.48550/arXiv.2301.12597 (дата звернення: 15.06.2025).

58. LLaMA: open and efficient foundation language models / H. Touvron та ін. Cornell University Library, 2023. (Препринт. Meta AI ; arXiv:2302.13971). URL: https://doi.org/10.48550/arXiv.2302.13971 (дата звернення: 15.06.2025).

59. Wase Z. M., Madisetti V. K., Bahga A. Object detection meets llms: model fusion for safety and security. *Journal of software engineering and applications*. 2023. Т. 16, № 12. С. 672–684. URL: https://doi.org/10.4236/jsea.2023.1612034 (дата звернення: 15.06.2025).

60. Flamingo: a visual language model for few-shot learning / J.-B. Alayrac та ін. Cornell University Library, 2022. (Препринт. DeepMind ; arXiv:2204.14198). URL: https://doi.org/10.48550/arXiv.2204.14198 (дата звернення: 15.06.2025).

61. Tsang S.-H. Review – flamingo: a visual language model for few-shot learning. *Medium*. URL: https://sh-tsang.medium.com/review-flamingo-a-visual-language-model-for-few-shot-learning-ec477d47e7bf (date of access: 15.06.2025).

62. ConceptNet. *ConceptNet*. URL: https://conceptnet.io (дата звернення: 05.06.2025).

63. VisualCOMET: reasoning about the dynamic context of a still image / J. S. Park та ін. *Computer vision – ECCV 2020*. Cham, 2020. С. 508–524. URL: https://doi.org/10.1007/978-3-030-58558-7_30 (дата звернення: 06.06.2025).

64. ATOMIC: an atlas of machine commonsense for if-then reasoning / M. Sap та ін. *Proceedings of the AAAI conference on artificial intelligence*. 2019.

T. 33. С. 3027–3035. URL: https://doi.org/10.1609/aaai.v33i01.33013027 (дата звернення: 06.06.2025).

65. NeuroAILab S. Physion: evaluating physical prediction from vision in humans and machines. *physion-benchmark.github.io*. URL: https://physion-benchmark.github.io/ (дата звернення: 05.06.2025).

66. Multimodal Chain-of-Thought Reasoning in Language Models / Z. Zhang та ін. Cornell University Library, 2023. (Препринт. Amazon Science ; arXiv:2302.00923). URL: https://doi.org/10.48550/arXiv.2302.00923 (дата звернення: 15.06.2025).

67. Sapireddy S. R. ResNet-50: introduction. *Medium*. URL: https://srsapireddy.medium.com/resnet-50-introduction-b5435fdba66f (дата звернення: 05.06.2025).

68. PrithivMLmods/AI-vs-Deepfake-vs-Real · datasets at hugging face. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/datasets/prithivMLmods/AI-vs-Deepfake-vs-Real (дата звернення: 11.06.2025).

69. ILSVRC/imagenet-1k · datasets at hugging face. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/datasets/ILSVRC/imagenet-1k (дата звернення: 11.06.2025).

70. Mask R-CNN / K. He та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2020. Т. 42, № 2. С. 386–397. URL: https://doi.org/10.1109/tpami.2018.2844175 (дата звернення: 11.06.2025).

72. Faster R-CNN: towards real-time object detection with region proposal networks / S. Ren та ін. *IEEE transactions on pattern analysis and machine intelligence*. 2017. Т. 39, № 6. С. 1137–1149. URL: https://doi.org/10.1109/tpami.2016.2577031 (дата звернення: 11.06.2025).

72. Podder S., Bhattacharjee S., Roy A. An efficient method of detection of COVID-19 using Mask R-CNN on chest X-Ray images. *AIMS biophysics*. 2021. Т. 8, № 3. С. 281–290. URL: https://doi.org/10.3934/biophy.2021022 (дата звернення: 11.06.2025).

73. Feature pyramid networks for object detection / T.-Y. Lin та ін. *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, м. Honolulu, HI, 21–26 лип. 2017 р. 2017. URL: https://doi.org/10.1109/cvpr.2017.106 (дата звернення: 11.06.2025).

74. COCO - common objects in context. *COCO - Common Objects in Context*. URL: https://cocodataset.org/index.htm#home (дата звернення: 11.06.2025).

75. What is Gemini and how it works. *Gemini*. URL: https://gemini.google/overview/ (дата звернення: 11.06.2025).

76. Cloud computing services | google cloud. *Google Cloud*. URL: https://cloud.google.com/ (дата звернення: 11.06.2025).

77. OpenRL/DeepFakeFace at main. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/datasets/OpenRL/DeepFakeFace/tree/main (дата звернення: 11.06.2025).