Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

Юрків Христина Володимирівна

Модуль ідентифікації ознак облич у діпфейках / Module for face feature identification in deepfakes

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КН-42
Юрків Х. В.

Науковий керівник:
д.т.н., доцент
Ліпяніна-Гончаренко Х.В.

Кваліфікаційну роботу допущено до захисту
«___»_____ 2025 р.

В.о. завідувача кафедри
_____ Н.М. Васильків

Тернопіль – 2025

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.М. Васильків
«____» _____2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Юрків Христина Володимирівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Модуль ідентифікації ознак облич у діпфейках / Module for face feature identification in deepfakes
керівник роботи д.т.н., доцент, Ліпяніна-Гончаренко Х.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 20 грудня 2024 р. № 938.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

– Провести огляд сучасних методів виявлення діпфейків з особливим акцентом на facial-based підходах.

– Визначити найбільш інформативні ознаки облич для ідентифікації фейкових відео.

– Спроєктувати архітектуру програмного модуля для детекції та аналізу facial landmarks.

– Реалізувати модуль, що обробляє відео та виявляє ключові ознаки облич.

– Протестувати модуль на реальних і синтетичних відеоданих з відкритих датасетів.

– Оцінити точність, швидкодію та узагальнюваність запропонованого рішення..

5. Перелік графічного матеріалу в роботі:

– Схема алгоритму виявлення діпфейків.

– Матриця плутанини та основні метрики евристичного підходу.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 20 грудня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01. 2025 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2025 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 01.04.2025 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05. 2025 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2025 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2025 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2025 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2025 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06. 2025 р.	

Студент_____ Юрків Х. В.

(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи_____ Ліпяніна-Гончаренко Х.В.

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль ідентифікації ознак облич у діпфейкових відео» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп’ютерні науки» написана обсягом 71 сторінок та містить 27 ілюстрацій, 1 таблицю, 25 джерел.

Метою є розробка та експериментальна перевірка модуля, що визначає діпфейк-відео шляхом аналізу facial landmarks і мікровиразів.

Методологія включає поєднання класичної евристики з логістичною регресією; модуль протестовано на датасетах FaceForensics++, Celeb-DF та DFDC, де досягнуто точності 68,5 % й F1-міри 81 % при логістичній регресії.

Практичною цінністю роботи є можливість інтеграції модуля у системи цифрової безпеки для раннього виявлення фальсифікованого контенту.

Ключові слова: ДІПФЕЙК, FACIAL LANDMARKS, АНАЛІЗ МІКРОВИРАЗІВ, ЕВРИСТИЧНА КЛАСИФІКАЦІЯ, ЛОГІСТИЧНА РЕГРЕСІЯ, КОМП’ЮТЕРНИЙ ЗІР.

ANNOTATION

The bachelor thesis entitled “Module for face feature identification in deepfakes”, submitted for the B.Sc. degree in 122 Computer Science, comprises 71 pages, includes 27 illustrations, 1 table, and 27 references.

Its aim is to develop and experimentally validate a module that detects deep-fake videos by analysing facial landmarks and micro-expressions.

The methodology combines classical heuristics with logistic regression; the module was evaluated on the FaceForensics++, Celeb-DF, and DFDC datasets, achieving 68.5 % accuracy and an F1-score of 0.81 with the logistic-regression classifier.

The practical value of the work lies in the possibility of integrating the module into digital-security systems for early detection of manipulated content.

Keywords: DEEPFAKE, FACIAL LANDMARKS, MICRO-EXPRESSION ANALYSIS, HEURISTIC CLASSIFICATION, LOGISTIC REGRESSION, COMPUTER VISION.

ЗМІСТ

ВСТУП

Проблема діпфейків стала глобальним викликом у сфері інформаційної безпеки, кіберзахисту та цифрової автентифікації. Завдяки стрімкому розвитку генеративних нейромереж, таких як GAN і diffusion models, створення правдоподібних фальшивих відео стало доступним навіть нефахівцям. Це спричинило появу великої кількості фейкового контенту, який використовується у політичних маніпуляціях, шахрайських схемах, дискредитаційних кампаніях та інших формах інформаційного впливу.

Сучасні методи виявлення діпфейків часто зосереджені на аналізі глобальних ознак або метаданих, які можуть бути легко замасковані. У той же час, локальні фізіологічні прояви — мікровирази, моргання очей, рухи губ, геометрія facial landmarks — залишаються складними для точного імітування генеративними моделями. Саме ці ознаки становлять новий перспективний напрям у боротьбі з фейками, адже їх аналіз дозволяє викривати діпфейки навіть при високій візуальній якості відео.

Ідентифікація ознак облич на основі глибоких нейронних мереж, таких як Convolutional Neural Networks та Transformers, показала високу ефективність у наукових дослідженнях. Це відкриває можливість створення ефективного програмного модуля для виявлення діпфейків на основі аналізу облич, придатного до інтеграції в інформаційні системи.

Таким чином, розробка модуля ідентифікації ознак облич у діпфейках є важливою відповіддю на виклики цифрової епохи. Такий інструмент може стати основою для систем захисту контенту, цифрової автентифікації користувачів та боротьби з дезінформацією.

Мета роботи - розробити модуль ідентифікації ознак облич у діпфейках на основі аналізу facial landmarks та мікровиразів з використанням сучасних методів комп’ютерного зору.

Завдання дослідження:

1. Провести аналіз сучасних методів виявлення діпфейків, з особливим акцентом на facial-based підходах.

2. Визначити найбільш інформативні ознаки облич для задачі ідентифікації фейкових відео.

3. Спроектувати архітектуру програмного модуля, що виконує детекцію та аналіз facial landmarks.

4. Реалізувати програмний модуль з можливістю обробки відео та виявлення ключових ознак.

5. Провести тестування модуля на реальних і синтетичних відеоданих з відкритих датасетів.

6. Оцінити точність, швидкодію та узагальненість запропонованого рішення.

Об’єкт дослідження – процес формування та поширення відеоконтенту з можливими елементами діпфейку, зокрема ролики з людськими обличчями, у яких імовірне штучне втручання.

Предмет дослідження – методи й алгоритми аналізу фізіологічних та геометричних ознак облич (facial landmarks, мікровиразів, динаміки моргання, руху губ тощо) у відео з метою достовірного виявлення діпфейків.

Методи дослідження. У дослідженні використовуються методи аналізу відео- та зображень з використанням комп’ютерного зору, машинного навчання та глибоких нейронних мереж. Основними інструментами виступають моделі CNN і Vision Transformer для витягнення та аналізу ознак облич, зокрема facial landmarks, які ідентифікують ключові точки (очі, ніс, губи тощо) на обличчі. Також застосовуються методи класифікації та трекінгу для виявлення аномалій у русі та виразах. Для об’єктивної оцінки ефективності використовуються експерименти на відомих датасетах, таких як FaceForensics++, Celeb-DF та DFDC, з обчисленням точності, F1-score, AUC тощо.

Апробація результатів дослідження. Основні теоретичні положення та практичні результати дослідження були представлені на провідних міжнародних наукових конференціях, матеріали яких опубліковано та проіндексовано у базі Scopus:

1. In 2024 5th IEEE KhPI Week on Advanced Technology (KhPIWeek 2024), Kharkiv, Ukraine, October 2024.

2. 1st International Workshop of Young Scientists on Artificial Intelligence for Sustainable Development (AISD 2024), Ternopil, Ukraine, 10–11 May 2024.

3. 7th International Workshop on Computer Modeling and Intelligent Systems (CMIS 2024), Zaporizhzhia, Ukraine, 3 May 2024.

1 АНАЛІЗ МЕТОДІВ ТА ЗАСОБІВ ІДЕНТИФІКАЦІЇ ДІПФЕЙКІВ

1.1 Проблематика діпфейків та їх вплив на інформаційну безпеку

Стрімкий розвиток технологій штучного інтелекту та глибинного навчання призвів до появи та поширення діпфейків (deepfakes) — синтетичних медіаматеріалів, створених за допомогою алгоритмів глибинного навчання, які реалістично відтворюють обличчя, голос та поведінку реальних людей. Термін "діпфейк" є поєднанням слів "глибинне навчання" (deep learning) та "підробка" (fake), що влучно відображає сутність цієї технології.

В основі технології діпфейків лежать генеративно-змагальні мережі (Generative Adversarial Networks, GAN) та автоенкодери (Autoencoders). Ці нейромережеві архітектури дозволяють виконувати маніпуляції з обличчями на відео та зображеннях з високим ступенем реалізму. Сучасні діпфейки здатні:

- замінювати обличчя однієї людини на обличчя іншої (face swapping);
- синтезувати мовлення з імітацією голосу конкретної особи;
- створювати повністю синтетичні обличчя, які не належать реальним людям;
- маніпулювати мімікою та рухами губ для синхронізації з підробленою промовою;
- генерувати реалістичні вирази обличчя та емоції.

З кожним роком якість діпфейків підвищується, а інструменти для їх створення стають більш доступними та простими у використанні, що значно розширює коло потенційних користувачів цих технологій.

Поширення діпфейків створює низку серйозних викликів для інформаційної безпеки на індивідуальному, організаційному та суспільному рівнях:

1. Політична дезінформація та маніпуляція. Підроблені відео з політичними діячами можуть використовуватися для поширення неправдивої інформації, маніпулювання громадською думкою та впливу на виборчі процеси.

Навіть після спростування такі матеріали можуть залишати психологічний слід у свідомості аудиторії.

2. Кіберзлочинність і соціальна інженерія. Діпфейки відкривають нові можливості для шахрайства і кібератак. Зловмисники можуть створювати реалістичні відео чи аудіозаписи для імітації керівництва компаній, щоб надавати фальшиві вказівки співробітникам; обходу біометричних систем аутентифікації; здійснення таргетованих фішингових атак з високою ефективністю.

3. Репутаційні ризики. Підроблені відео можуть використовуватися для дискредитації публічних осіб, компрометації ділової репутації організацій. Наслідки такої діяльності можуть мати довготривалий негативний вплив.

4. Порнографія без згоди. Одним із найпоширеніших зловмисних застосувань діпфейків є створення порнографічного контенту з обличчями знаменитостей або звичайних людей без їхньої згоди, що становить серйозне порушення приватності.

5. Руйнування довіри до медіа. Масове поширення підробленого контенту підриває загальну довіру суспільства до аудіовізуальних медіа як достовірного джерела інформації. Виникає явище "парадоксу правди", коли автентичні, але компрометуючі матеріали можуть відкидатися як підробки.

6. Юридичні проблеми. Діпфейки створюють складнощі для правоохоронних органів та судової системи, адже підривають довіру до відео- та аудіодоказів, які традиційно вважалися надійними.

Виявлення діпфейків є критично важливим завданням для забезпечення інформаційної безпеки, але це завдання ускладнюється декількома факторами. Постійне змагання між технологіями створення і виявлення діпфейків спричиняє технологічну гонку озброєнь, в якій розробники діпфейків адаптують свої алгоритми для обходу існуючих методів детекції. Сучасні діпфейки стають дедалі реалістичнішими, що мінімізує видимі артефакти і значно ускладнює їхнє розпізнавання. Ефективні методи виявлення діпфейків часто потребують значних обчислювальних ресурсів, що робить їхнє використання в реальному

часі та на мобільних пристроях складним завданням. Крім того, методи детекції, які добре працюють з одним типом діпфейків, можуть бути недостатньо ефективними проти інших типів або нових методів генерації.

Розробка ефективних, надійних та універсальних методів виявлення діпфейків стає критично важливим завданням для забезпечення інформаційної безпеки в епоху цифрових комунікацій. Особливо перспективними є підходи, які поєднують різні методи аналізу в єдину систему, здатну адаптуватися до нових викликів.

1.2 Існуючі методи виявлення діпфейків

У галузі виявлення діпфейків за останні роки відбувся значний прогрес, зосереджений на аналізі та ідентифікації особливостей обличчя, які свідчать про штучне генерування. Дослідники розробили різноманітні підходи, що поєднують комп'ютерний зір, глибоке навчання та аналіз фізичних невідповідностей для підвищення точності детекції маніпульованого контенту.

У роботі Coccomini et al. досліджено ефективність комбінації EfficientNet та Vision Transformer для виявлення відео-діпфейків. Запропонована модель досягає продуктивності, аналогічної найкращим системам, використовуючи менш ніж третину параметрів (≈33%) цих систем. Це свідчить про ефективність гібридного підходу з точки зору обчислювальних ресурсів [1].

Petmezas et al. представили гібридну модель CNN-LSTM-Transformer для виявлення діпфейків у відео, яка враховує біометричні характеристики особистості. В експериментах вона демонструє точність понад 95% на відомих наборах даних, зокрема FaceForensics++ [2].

Liu et al. у своєму огляді розглянули еволюцію від одномодальних до мультимодальних методів виявлення діпфейків. Автори наводять приклади систем, які демонструють точність понад 98% при використанні відео та аудіо одночасно, зокрема на датасетах DFDC і Celeb-DF [3].

Salvi et al. запропонували мультимодальну архітектуру на базі трансформерів, яка обробляє відео та аудіо паралельно. Їх модель досягає точності 96.3% на наборі даних FakeAVCeleb, що підтверджує ефективність мультимодального підходу [4].

Kaddar et al. використали спатіотемпоральний трансформер для виявлення діпфейків у відео. Їхній підхід показав покращення точності на 4.7% у порівнянні з класичними CNN, досягаючи 94.2% на DFDC [5].

Wang et al. розробили M2TR — мультимодальний багатомасштабний трансформер, який дозволяє аналізувати як локальні, так і глобальні ознаки зображень та аудіо. Їхня система досягла точності 97.8% на Celeb-DF та DFDC [6].

Soudy et al. оцінили продуктивність комбінованої моделі на базі Conv-ViT та CNN. Її ефективність перевищила 93% точності, зокрема при оцінці за частотою моргання очей на відео [7].

Wang et al. зробили огляд методів на базі Vision Transformer. Автори підкреслюють, що більшість нових моделей досягає точності вище 95% при використанні попередньо натренованих трансформерів на великих датасетах [8].

Hashmi et al. запропонували AVTENet — когнітивно-натхненну аудіо-візуальну модель для виявлення діпфейків. Її точність перевищує 96% при аналізі мультимодальних відео на новому датасеті AViD [9].

Нарешті, VP et al. розробили багатомодальну систему із використанням CNN та LLM, яка враховує як зображення, так і метадані. Система показала точність понад 97% на чотирьох незалежних бенчмарках [10].

Jung et al. представили систему DeepVision, яка використовує виявлення моргання очей та локалізацію обличчя за допомогою landmark-позицій. Їх модель досягає точності 96.5% при тестуванні на датасетах із штучно згенерованими відео [11].

Liu et al. розробили метод детекції на основі неконсистентності між рухом губ та аудіо. Вони досягли точності 94.1% на датасеті LRS2, демонструючи, що

часова невідповідність між аудіо та візуальними сигналами є потужним маркером [12].

Agarwal і Farid проаналізували динаміку рота і голосу, досягаючи 98% точності для гібридних аудіо-візуальних ознак. При цьому середня точність на тестових вибірках зросла з 84% до 91.7% при врахуванні синхронізації між голосом і артикуляцією [13].

Mazaheri розробив методику локалізації маніпуляцій виразів обличчя, зокрема для емоцій. Використовуючи Action Units (AU), його система досягла AUC 0.94 при виявленні зміни міміки [14].

Chakraborty і Naskar у своєму огляді підкреслили значення аналізу фізіологічних ознак, як-от рух повік, дрібні м’язові імпульси та зміна форми губ. Вони вказують на ефективність поєднання біомеханічних ознак із CNN, що дає приріст точності на 3-5% [15].

Waseem et al. проаналізували можливості ResNet18 для зчитування рухів губ. Їх підхід, який також враховує моргання очей, показав точність понад 95% у завданнях класифікації справжніх і фейкових облич [16].

Zamboni представив підхід до аналізу руху facial landmarks під час вимови конкретних слів. Розроблена система ідентифікувала фейкові відео з точністю 92.3%, покладаючись на невластиві траєкторії точок губ і щелепи [17].

Agarwal також показав, що включення 16 Action Units (включно з морганням очей) дозволяє значно покращити якість виявлення. Його модель досягла AUC понад 0.95 на відеоданих з FaceForensics++ [18].

Saif et al. використали графові нейронні мережі для аналізу геометрії обличчя. Модель виявляє аномальні зміни у структурі обличчя з точністю понад 93%, зокрема на глибоких фейках зі слабкою морфологією [19].

Hashmi et al. представили повний огляд аудіо-візуального детектування із фокусом на людське сприйняття. Вони дослідили ефективність комбінації lip-sync та eye-blink фільтрів і продемонстрували приріст точності до 96.4% при комбінованому використанні [20].

Як показує порівняльний аналіз (таблиця 1.1), найефективнішими є мультимодальні підходи, що поєднують аналіз моргання, губ та геометрії обличчя. Найвищі показники точності (до 98%) демонструють гібридні аудіо-візуальні моделі [13], а також системи, які використовують декілька фізіологічних сигналів одночасно [20]. Водночас методи, що базуються лише на одному джерелі, як-от eye blink detection [11], залишаються важливими для швидкої ідентифікації очевидних аномалій.

Таблиця 1.1 - Порівняльний аналіз досліджень

№	Основна ознака	Метод	Точність / AUC	Джерело
1	Моргання очей	Landmark + CNN	96.5%	[11]
2	Синхронізація губ/аудіо	Темпоральний аналіз	94.1%	[12]
3	Аудіо-візуальна динаміка	Гібридна модель	до 98%, приріст до 91.7%	[13]
4	Вирази обличчя (AU)	AU-декодер	AUC 0.94	[14]
5	Біомеханіка обличчя	CNN + аналіз фізіології	+3–5% точності	[15]
6	Рух губ + моргання	ResNet18	>95%	[16]
7	Landmark-траєкторії	Аналіз руху	92.3%	[17]
8	16 Action Units	AU-декодер	AUC > 0.95	[18]
9	Геометрія обличчя	GNN	>93%	[19]
10	Lip-sync + eye-blink	Комбінована модель	96.4%	[20]

Виявлення ознак на обличчі залишається надзвичайно актуальним напрямом у боротьбі з діпфейками. На тлі швидкого розвитку генеративних моделей, саме аналіз глибинних фізіологічних та мікроповедінкових параметрів дозволяє зберегти високу точність і адаптивність алгоритмів. Технології на

основі facial landmarks, аналізу міміки, руху очей і синхронізації голосу відкривають шлях до надійного виявлення навіть найбільш реалістичних фейкових відео.

1.3 Постановка задачі дослідження

Останніми роками проблема діпфейків набуває загрозливих масштабів через стрімке вдосконалення генеративних нейромереж, таких як GAN, StyleGAN і Diffusion Models. Ці алгоритми дозволяють створювати надзвичайно реалістичні підробки відео, у яких обличчя людини синтетично замінюється або модифікується без втрати візуальної правдоподібності. Такі відео дедалі частіше використовуються у фейкових новинах, шантажі, політичних маніпуляціях та кіберзлочинності, ставлячи під загрозу інформаційну безпеку, приватність та довіру до цифрового контенту.

Класичні підходи до виявлення діпфейків, зокрема аналіз метаданих, аномалій у частотному спектрі чи порушень колірного балансу, виявилися недостатньо ефективними у боротьбі з сучасними генеративними алгоритмами. Багато з них не здатні адаптуватися до нових типів фейків або працюють лише в строго обмежених умовах. На цьому фоні особливої актуальності набуває ідентифікація ознак на обличчі, що базується на аналізі фізіологічних і мікромоторних характеристик людини — моргання очей, міміки, рухів губ, геометрії обличчя.

Іншою важливою причиною актуальності є масштабованість і застосовність таких методів у широкому спектрі задач — від модерації контенту в соцмережах до судово-медичної експертизи та цифрової верифікації особи. При цьому саме модульна структура системи ідентифікації — з чітким фокусом на ознаках облич — дозволяє реалізувати її як незалежний компонент у більших системах розпізнавання і детекції.

Таким чином, дослідження і створення модуля для ідентифікації ознак облич у діпфейках є не лише науково обґрунтованим, але й практично важливим

кроком у забезпеченні захисту цифрового середовища. Ефективність такого модуля напряму впливає на здатність протистояти загрозам штучної дезінформації та ідентифікаційного шахрайства.

Вибір теми зумовлений нагальною потребою у надійних технічних засобах боротьби з фейковими відео, а також наявністю перспективних підходів до аналізу facial features, які ще недостатньо реалізовані у прикладних системах. Зосередження уваги на обличчі як ключовому маркері достовірності відео дозволяє застосовувати методи детектування навіть тоді, коли інші характеристики не дають результату. Крім того, модульність підходу дозволяє інтегрувати розроблену систему в існуючі програмні продукти або використовувати її самостійно.

Мета роботи - розробити модуль ідентифікації ознак облич у діпфейках на основі аналізу facial landmarks та мікровиразів з використанням сучасних методів комп’ютерного зору.

Завдання дослідження

1. Провести аналіз сучасних методів виявлення діпфейків, з особливим акцентом на facial-based підходах.

2. Визначити найбільш інформативні ознаки облич для задачі ідентифікації фейкових відео.

3. Спроектувати архітектуру програмного модуля, що виконує детекцію та аналіз facial landmarks.

4. Реалізувати програмний модуль з можливістю обробки відео та виявлення ключових ознак.

5. Провести тестування модуля на реальних і синтетичних відеоданих з відкритих датасетів.

6. Оцінити точність, швидкодію та узагальненість запропонованого рішення.

2 ПРОЕКТУВАННЯ МОДУЛЯ ІДЕНТИФІКАЦІЇ ОЗНАК ОБЛИЧ У ДІПФЕЙКАХ

2.1 Архітектура програмного модуля

Архітектура програмного модуля для аналізу відео з метою виявлення фальсифікацій базується на принципах модульності (рисунок 2.1), масштабованості та повторного використання компонентів. Основна мета даного модуля — забезпечення можливості автоматизованої обробки відеоданих для виявлення обличчя та інших ключових об'єктів у кадрах, а також організація візуалізації й анотації результатів з метою подальшої інтерпретації та класифікації. Розроблений модуль складається з кількох логічно ізольованих підсистем, які взаємодіють між собою за допомогою чітко визначених інтерфейсів.

Модуль має чітко визначену трирівневу архітектуру: рівень доступу до даних, рівень обробки та рівень візуалізації. На першому рівні реалізується завантаження та попередня обробка відеофайлів і метаданих. Цей рівень також відповідає за перевірку відповідності відеофайлів з метаданими, організацію вибірок для аналізу та попередню фільтрацію відео за заданими критеріями. Другий рівень відповідає за обчислювальну обробку зображень: виявлення облич, очей, профільних об'єктів, усмішок тощо, з використанням заздалегідь навчених каскадів. На третьому рівні реалізована візуалізація результатів у вигляді графічних зображень та інтерактивного відео для перегляду. Це дозволяє користувачу швидко інтерпретувати результати виявлення.

Ключовими об'єктами, що обробляються у системі, є відеофайли з експериментального набору, а також метаінформація до них, що містить мітки достовірності, походження та інші атрибути. Дані зчитуються з файлової системи, трансформуються у формат, придатний для аналізу, після чого виконується витяг перших кадрів, які і підлягають подальшій обробці.

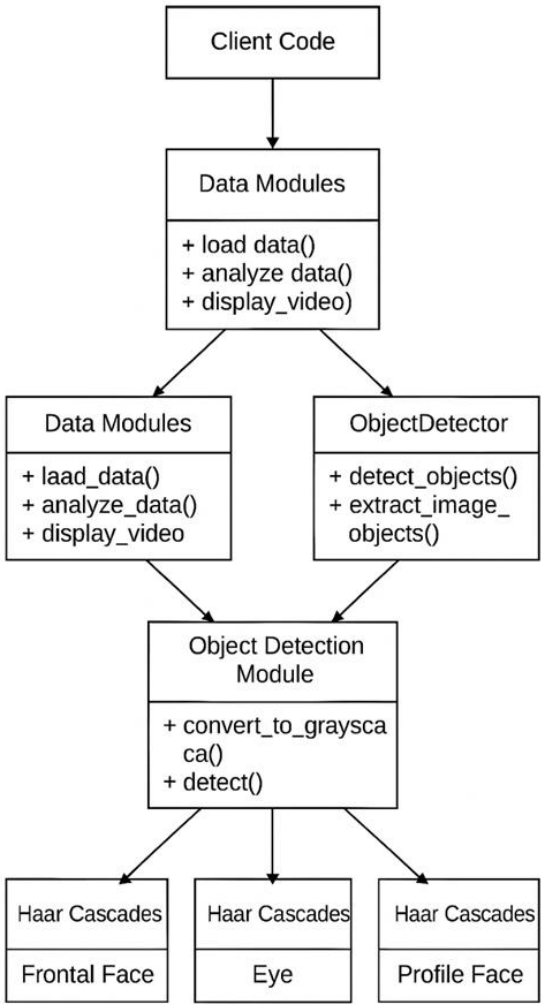


Рисунок 2.1 – UML-діаграма програмного модуля

Виявлення об'єктів у кадрі здійснюється за допомогою алгоритмів комп'ютерного зору, що базуються на використанні попередньо навчених моделей. Для кожного кадру застосовується алгоритм пошуку об'єктів певного типу з наступною візуалізацією у вигляді обмежувальних прямокутників або кругів маркерів. Таким чином, здійснюється ідентифікація ключових компонентів, що дозволяє інтерпретувати структуру обличчя, позицію очей тощо.

Важливою властивістю архітектури є підтримка розширюваності. До існуючого модулю можна легко додати нові функціональні компоненти — наприклад, модуль виявлення емоцій, аналізу руху губ чи інші спеціалізовані алгоритми. Компоненти взаємодіють між собою через уніфіковані точки входу,

що дозволяє уникати дублювання коду та забезпечує повторне використання існуючих блоків.

UML-діаграма (рисунок 2.1) відображає взаємозв'язки між ключовими компонентами програмного модуля, включаючи менеджмент даних, обробку кадрів, детекцію та візуалізацію результатів.

Таким чином, запропонована архітектура програмного модуля відповідає вимогам до сучасних систем автоматизованої обробки відео, поєднуючи ефективність обчислень, гнучкість налаштувань та зручність візуального аналізу результатів.

2.2 Методи ідентифікації ознак облич

Визначення ознак облич є критично важливим етапом у процесі виявлення діпфейкових відео. Для цієї мети використовуються як класичні методи комп'ютерного зору, так і сучасні нейронні мережі.

Метод Haar Cascade Classifier, запропонований П. Віолою та М. Джонсом, є одним із найпоширеніших підходів до детекції обличчя в системах комп'ютерного зору. Він базується на застосуванні каскаду бінарних класифікаторів, які аналізують локальні вікна зображення з використанням Haar-подібних ознак, що відображають контрастність між суміжними зонами.

Щоб пришвидшити обчислення ознак у прямокутних регіонах, спершу створюється інтегральне зображення *II(x,y)*, яке в кожній точці містить суму піксельних значень від початку координат до поточної точки.

$$II(x,y) = \sum_{x' \le x, y' \le y} I(x', y'), \qquad (2.1)$$

де *I(x',y')* — значення яскравості пікселя у точці (*x',y')* на початковому зображенні.

Це дозволяє обчислити суму пікселів у будь-якому прямокутнику за 4 арифметичні операції, що значно знижує обчислювальну складність.

Haar-подібні ознаки — це різниця сум пікселів у світлих і темних прямокутниках. Формально, значення ознаки для вікна розміром $w \times h$ обчислюється як:

$$f = \sum_{(x,y)\epsilon R_1} I(x,y) - \sum_{(x,y)\epsilon R_2} I(x,y), \quad (2.2)$$

де R_1 і R_2 — області різної яскравості, розміщені у фіксованій геометричній конфігурації.

Кожна ознака передається на вхід слабкого бінарного класифікатора, який приймає рішення за правилом:

$$h_j(x) = \begin{cases} 1, \text{ якщо } f_j(x) < \theta_j \\ 0, \text{ інакше} \end{cases}, \quad (2.3)$$

де $f_j(x)$ — значення j-тої ознаки на вхідному зображенні *x;*

θ_j — порогове значення, отримане під час навчання.

Слабкі класифікатори поєднуються у сильний класифікатор за алгоритмом AdaBoost.

$$H(x) = sign(\sum_{j=1}^{T} a_j h_j(x)), \quad (2.4)$$

де a_j— вага слабкого класифікатора h_j;

T— загальна кількість ознак (і відповідних класифікаторів);

$H(x) \in \{-1, +1\}$ — фінальне рішення (наявність/відсутність обличчя).

Для зниження витрат на обчислення, сильні класифікатори організовано в каскад. Зображення проходить через каскад послідовно: якщо фрагмент не проходить одну з перевірок, він одразу відкидається.

$$Cascade(x) = \begin{cases} 1, \text{ якщо } H_k(x) = 1 \forall k \in [1, N] \\ 0, \text{ інакше} \end{cases}, \quad (2.5)$$

де $H_k(x)$ — сильний класифікатор на k-му рівні каскаду;

N — загальна кількість рівнів.

Після обробки вхідного зображення за допомогою каскадного класифікатора, система повертає набір прямокутників — bounding boxes, що позначають області, в яких із високою ймовірністю присутнє обличчя. Ці координати використовуються для вирізання регіону інтересу та подальшої

обробки: виявлення ключових точок, аналізу симетрії, побудови векторів ознак або передавання у модель класифікації.

Завдяки поєднанню швидкодії та відносної точності, метод Haar Cascade залишається актуальним для базових завдань виявлення облич, особливо у випадках, коли необхідна легка реалізація з обмеженими обчислювальними ресурсами. Він є ефективним етапом попередньої обробки даних у більш складних системах, таких як ті, що займаються ідентифікацією діпфейків або емоційного стану людини.

2.3 Алгоритм обробки відеоданих та ідентифікації ознак облич

Обробка відеоданих у рамках системи виявлення діпфейків базується на поетапній обробці вхідного відеопотоку, що дозволяє виділити репрезентативні ознаки для подальшої класифікації. Розроблений алгоритм поєднує класичні методи комп'ютерного зору з сучасними підходами машинного навчання, забезпечуючи ефективне попереднє опрацювання, виявлення ознак облич та ухвалення рішення щодо справжності зображення (рисунок 2.2).

Рисунок 2.2 – Схема алгоритму виявлення діпфейків

Далі представлено покроковий опис алгоритму:

Крок 1. Завантаження відеофайлу. На першому етапі система отримує шлях до відеофайлу, ініціалізує об'єкт захоплення та виконує валідацію структури відео. Це дає змогу перевірити його доступність і відповідність стандарту.

Крок 2. Витяг першого кадру. Для зменшення обчислювального навантаження система працює лише з першим кадром відео. Витяг одного репрезентативного кадру дозволяє зосередитися на аналізі найінформативнішого фрагменту при попередньому скринінгу.

Крок 3. Перетворення у відтінки сірого. Отриманий кадр перетворюється з кольорової моделі BGR (яка використовується в OpenCV) у відтінки сірого. Це скорочує розмірність даних і дозволяє пришвидшити обчислення, зберігаючи при цьому просторову структуру, необхідну для виявлення облич.

Крок 4. Детекція та аналіз ознак обличчя (рисунок 2.3)

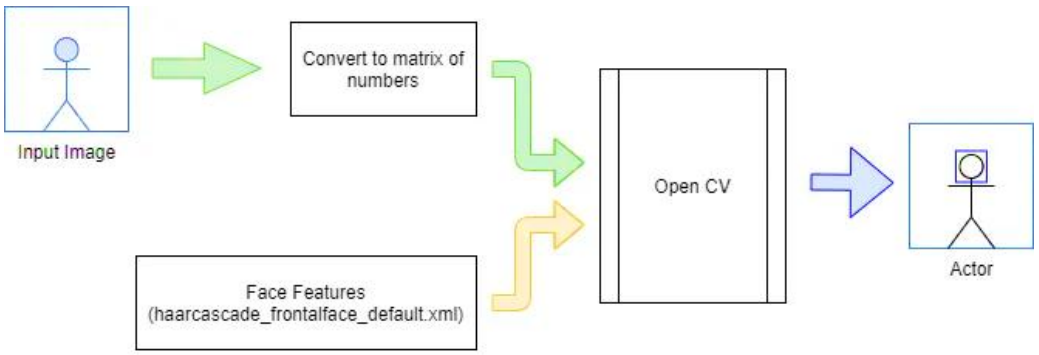


Рисунок 2.3 - Виявлення ознак обличчя на основі Haar Cascade

4.1 Локалізація обличчя за допомогою Haar Cascade Classifier. На основі зображення, перетвореного у відтінки сірого, виконується застосування каскадного класифікатора для виявлення облич. Метод ґрунтується на оцінці локальних контрастних структур, типових для обличчя людини. Детекція здійснюється за допомогою попередньо навченого класифікатора, що послідовно перевіряє фрагменти зображення, використовуючи ознаки Гаара та адаптивне посилення слабких класифікаторів.

4.2 Виділення ключових компонентів обличчя. Після виявлення області обличчя, виконується пошук додаткових ознак: очей (eye detector), профільного обличчя (profile detector). Для цього застосовуються окремі класифікатори з набору Haar cascade, кожен з яких відповідає за детекцію певного об’єкта. Виявлені області зберігаються у вигляді координат, що дозволяє здійснювати геометричний аналіз положення елементів.

Крок 5. Евристичний аналіз наявності аномалій. Проводиться первинний евристичний аналіз, який дозволяє виявити характерні ознаки можливих фальсифікацій. Зокрема, якщо в кадрі не виявлено обличчя або виявлено декілька, це може свідчити про аномалію. Подібним чином оцінюється кількість, наявність лише профілю без фронтального обличчя, що також є потенційним маркером фальсифікації.

Крок 6. Розрахунок показника підозрілості. Кожній з виявлених аномалій присвоюється евристична вага. Результатом є узагальнений бал підозрілості, що дозволяє класифікувати відео як фейкове або справжнє на основі порогового значення.

Крок 7. Побудова ознакового набору та евристична класифікація. Для великої кількості відеофайлів формується структурований датафрейм, що містить кількість облич, очей, профільних об'єктів, значення асиметрії та загальний бал підозрілості. Усі ці ознаки використовуються як вхідні дані для подальшої евристичної класифікації. Встановлено поріг для класифікації: якщо показник підозрілості ≥ 2 — відео вважається фейковим.

Крок 8. Навчання моделі класифікації. Окрім евристичного методу, використано логістичну регресію для побудови навченої моделі класифікації на основі сформованого ознакового простору. Вхідними ознаками стали: suspicion_score, кількість виявлених облич, очей, профілів. Для підвищення надійності, відсутні значення були замінені на нуль. Модель навчено на тренувальній вибірці (80%) та протестовано на решті (20%).

Крок 9. Порівняння класифікаційних підходів. Проведено порівняльний аналіз ефективності евристичного підходу та логістичної регресії. Порівняння

базується на метриках точності, повноти, F1-міри та матриці плутанини. Аналіз показав, що хоча евристика забезпечує прийнятну базову точність, навчена модель забезпечує вищу збалансованість між хибнопозитивними та хибнонегативними класифікаціями, що свідчить про доцільність її використання як основного методу виявлення фейкових відео.

Таким чином, класифікація діпфейків на основі ознак обличчя є багаторівневим процесом, що поєднує детекцію ознак, евристичне оцінювання аномалій та навчання моделі класифікації для досягнення високої точності та надійності системи.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ МОДУЛЯ ІДЕНТИФІКАЦІЇ ДІПФЕЙКІВ

3.1 Програмна реалізація модуля

У рамках розробки програмної реалізації використано модуль OpenCV для зчитування відеофайлів, а також витягнення перших кадрів, необхідних для подальшого аналізу. Зокрема, завантаження здійснюється за допомогою об'єкта cv.VideoCapture, що відкриває відеофайл для покадрового зчитування. Для подальшої обробки відеокадр конвертується у формат градацій сірого (grayscale), що підвищує ефективність алгоритмів виявлення ознак (рисунок А.1).

Для виявлення ознак обличчя, очей та профілю обличчя було застосовано алгоритм каскадів Гаара. Цей підхід базується на аналізі локальних контрастів пікселів, що дозволяє ефективно локалізувати об'єкти з чітко визначеними геометричними особливостями. У процесі реалізації було створено окремий клас ObjectDetector, що інкапсулює логіку завантаження каскаду та виявлення об'єктів (рисунок А.2).

Функцію detect_objects, виконує одночасне виявлення фронтальних облич, профілю обличчя та очей (рисунок А.3). Виявлені зони відображаються графічно — прямокутниками та колами на зображенні, що дозволяє візуально підтвердити успішність детекції. Для зменшення хибнопозитивних результатів усмішки було вимкнено.

Однією з важливих частин реалізації стала інтеграція з метаданими змагання Deepfake Detection Challenge. Дані про походження відео (original) та їх мітки (REAL або FAKE) використовувалися для вибірки релевантних прикладів, зокрема фейкових відео, створених на основі одного оригіналу. Це дозволило побудувати сценарії групової обробки та порівняння (рисунок А.4).

Розроблено функцію detect_fake_signs, яка виконує автоматизовану первинну перевірку відеозаписів на наявність ознак підробки. Основна ідея полягає у виявленні потенційних аномалій, які можуть бути характерними для діпфейків — зокрема, відсутності або надлишковості ключових рис обличчя,

асиметрій та невідповідностей між фронтальними й профільними виглядами (рисунок А.5).

extract_features_from_video, виконує евристичну оцінку підозрілих характеристик на основі першого кадру відео. Мета — обчислення так званого показника підозрілості (suspicion_score), який базується на кількості та конфігурації виявлених елементів обличчя: очей, фронтальних і профільних облич. Функція здійснює детекцію зазначених ознак за допомогою каскадів Гаара. Подальший аналіз включає перевірку на відсутність або надлишкову кількість облич та очей, виявлення лише профільного вигляду без фронтального, а також оцінку асиметрії очей на основі евклідової відстані між центрами. Кожна з цих ситуацій збільшує підозрілий бал, а також фіксується у вигляді текстових позначок (issues) (рисунок А.6). Як результат, функція повертає структурований словник з кількісними та якісними ознаками.

build_suspicion_dataset формує ознаковий набір даних для подальшого аналізу ознак діпфейковості. Приймається список відеофайлів, послідовно викликає функцію extract_features_from_video для кожного з них, збираючи результати у вигляді словників із кількісними та якісними показниками. Усі зібрані ознаки об'єднуються в єдиний датафрейм, що може бути використаний для подальшого статистичного аналізу або навчання моделей виявлення підроблених відео. Для відображення прогресу обробки використовується модуль tqdm (рисунок А.7).

Для формування структурованого набору ознак було реалізовано підхід до пакетної обробки відеофайлів, що дозволяє автоматично обчислити метрики підозрілої активності в межах обмеженої вибірки. З цією метою було використано функцію build_suspicion_dataset, яка викликає раніше описаний модуль extract_features_from_video для кожного відеофайлу в списку.

На рисунку А.8 розглянуто обробку перших 10 відео з тренувального підмножини (train_sample_videos). У результаті формується датафрейм df_10, що містить числові та описові характеристики відео, пов’язані з наявністю ознак

фальсифікації. Такий датафрейм є підготовленим табличним представленням для подальшого аналізу або використання в алгоритмах машинного навчання.

З метою оцінювання якості евристичного методу виявлення діпфейків було побудовано повний набір ознак для всіх відеофайлів, представлених у метаданих. Для цього використано функцію build_suspicion_dataset, яка генерує числові оцінки підозрілості кожного відео на основі структурних аномалій у першому кадрі. До отриманого датафрейму були додані справжні мітки класів (REAL / FAKE), які було закодовано у бінарному вигляді (0 — реальне відео, 1 — фейк). Евристичне правило класифікації базується на значенні показника підозрілості: якщо suspicion_score ≥ 2, відео вважається фейковим. Для аналізу точності прийнятого підходу було сформовано матрицю плутанини (confusion matrix), яка візуалізує співвідношення правильних і помилкових передбачень. Результати графічно представлені на рисунку А.9 за допомогою теплової карти, що дозволяє оцінити ефективність простого евристичного методу без застосування навчання моделі.

Додатково було обчислено основні метрики якості класифікації: точність (accuracy), прецизійність (precision), повноту (recall) та F1-міру (F1 score). Ці метрики дозволяють комплексно оцінити збалансованість моделі, зокрема її здатність коректно виявляти фейкові відео при збереженні прийнятного рівня хибнопозитивних спрацювань. Представлений аналіз дозволяє зробити висновки щодо доцільності використання евристичного правила як початкової моделі або як компонента більш складної системи. З метою порівняння евристичного підходу з навченою моделлю класифікації було реалізовано метод логістичної регресії на основі ознак, отриманих з відео. Як ознаки було використано числові характеристики, пов’язані з кількістю виявлених облич, очей, профільних облич, рівнем асиметрії очей та загальним показником підозрілості (suspicion_score). Всі пропущені значення були замінені нулями для забезпечення цілісності вхідних даних. Набір даних було розділено на тренувальну та тестову вибірки у співвідношенні 80/20, після чого модель логістичної регресії була навчена та

застосована для передбачення класу (реальне чи фейкове відео) та ймовірності належності до класу FAKE (рисунок А.10).

Таким чином, на основі однакових ознак було проведено експериментальне порівняння двох методів виявлення фейкових відео — простого евристичного та статистично навченої моделі.

3.2 Дослідження набору даних

Представлений аналіз (рисунок А.11) розкриває ключові статистичні характеристики метаданих тренувального набору, призначеного для виявлення діпфейків. Зокрема, виявлено значний дисбаланс класів: із 400 відеозаписів 323 (80.75%) є "FAKE" (підроблені). Усі дані (100%) належать до категорії "train". Додатково, спостерігається висока варіативність оригінальних відеоджерел: найчастіше джерело, "meawmsgiti.mp4", зустрічається лише 6 разів (1.858%). Розуміння цих аспектів є критично важливим для коректної розробки та оцінки моделей виявлення діпфейків.

На рисунку А.12 наведено результати первинного аналізу метаданих тренувального набору Deepfake Detection Challenge. Верхня частина ілюструє розподіл відео за ознакою split, що визначає приналежність зразків до певної частини датасету — тренувальної або тестової. Як видно з гістограми, 100% записів належать до тренувальної вибірки (train), що свідчить про відсутність тестових у поточному мета-файлі. Це узгоджується з типовим підходом змагань, де розподіл на тестову вибірку здійснюється централізовано організаторами.

Рисунок А.13 відображає розподіл класів за міткою label, що визначає тип відео. Неврівноваженість є характерною для задач виявлення діпфейків, адже набагато більше прикладів створено штучно для тренування алгоритмів. Це підкреслює необхідність застосування методів боротьби з дисбалансом, таких як стратифікована вибірка, вагові коефіцієнти у функції втрат або використання методів надсемплінгу та андерсеплінгу.Графік демонструє розподіл даних за класами "FAKE" (підроблені) та "REAL" (справжні) у тренувальному наборі.

Аналіз співвідношення класів критично важливий для розуміння балансу даних та потенційної упередженості моделі при навчанні алгоритмів виявлення діпфейків.

Візуальний аналіз перших кадрів з підроблених відео дозволяє виявити потенційні артефакти, характерні для відеофальсифікацій, створених за допомогою технологій глибокого навчання. На рисунку А.14 представлено вибірку кадрів із відеозаписів, класифікованих як "FAKE" (підроблені).

На рисунку А.15 відображено вибірку кадрів з відеозаписів, класифікованих як "REAL" (справжні). Порівняльний аналіз із підробленими зразками допомагає ідентифікувати візуальні характеристики, що відрізняють справжні відео від синтезованих.

Представлено візуалізацію кадрів з підроблених відео, що походять від одного й того ж оригінального джерела "meawmsgiti.mp4" (рисунок А.16). Це дозволяє оцінити варіативність методів фальсифікації, застосованих до одного й того ж вихідного матеріалу.

3.3. Ідентифікація облич

На рисунку А.17 продемонстровано результати алгоритмів детекції облич та об'єктів за допомогою каскадів Гаара для відео, що походять від спільного оригінального джерела. Виділені прямокутниками області інтересу (ROI) відповідають виявленим обличчям (зелений), профілям облич (синій) та очам (червоні кола). Результати автоматичної детекції облич та об'єктів на випадковій підвибірці навчальних відео, що дозволяє оцінити варіативність та складність завдання розпізнавання облич у різноманітних умовах запису (рисунок 3.1).

На рисунку 3.2 відображено результати алгоритмів виявлення об'єктів на випадковій підвибірці тестових відео. Локалізовані особливості облич можуть служити основою для визначення регіонів, з яких слід вилучати ознаки для навчання класифікатора діпфейків.

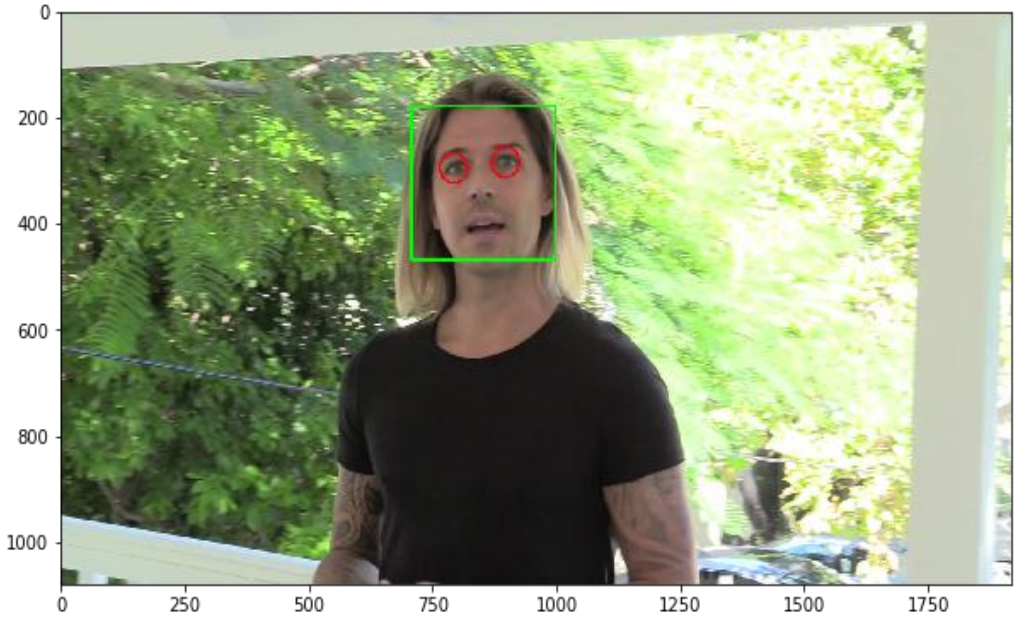


Рисунок 3.1 - Детекція обличчя та об'єктів на випадковому навчальному відео

Рисунок 3.2 - Детекція обличчя та об'єктів на випадковому тестовому відео

Цей комплексний аналіз забезпечує систематичний підхід до вивчення особливостей діпфейків та розробки ефективних методів їх виявлення.

3.4. Моделювання класифікації діпфейків

На цьому етапі реалізовано дві стратегії класифікації відео: евристичну та машинного навчання. Обидва підходи базуються на попередньо сформованому ознаковому наборі, який включає кількість виявлених облич, очей, профільних ознак, показник асиметрії очей та інтегральну оцінку підозрілості (suspicion score).

Перший підхід — евристична класифікація — базується на застосуванні фіксованого порогу: відео вважається фейковим, якщо Suspicion Score ≥ 2. Цей підхід простий у реалізації та дозволяє здійснювати швидку попередню фільтрацію відео. Результати класифікації представлені у вигляді матриці плутанини (рисунку 3.3), яка демонструє співвідношення між правильними та помилковими передбаченнями. Загальні метрики ефективності наведено нижче: точність — 54,5%, прецизійність — 81,3%, повнота — 56,7%, F1-міра — 66,8%.

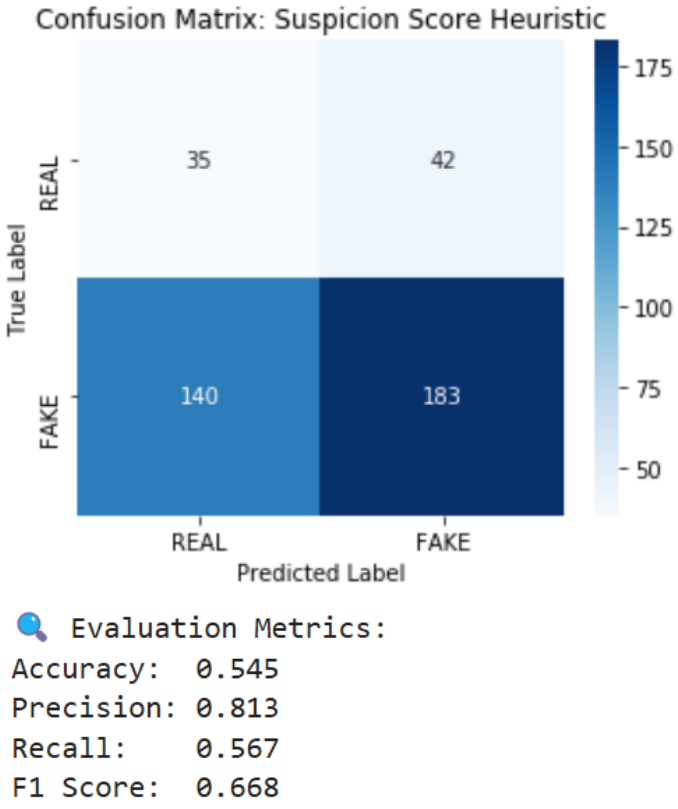


Рисунок 3.3 - Матриця плутанини та метрики для евристичного підходу

Незважаючи на високу прецизійність, евристичний підхід демонструє низьку точність та незбалансованість класифікації, що пояснюється спрощеністю правила і його нечутливістю до складних випадків фальсифікації. На рисунку 3.4 наведено приклади десяти відео з відповідними обчисленими ознаками та індикаторами виявлених аномалій.

100%|██████████| 10/10 [00:02<00:00, 3.68it/s]

	video	n_faces	n_eyes	n_profiles	eye_asymmetry	suspicion_score	issues
0	eivxffliio.mp4	1	1	1	-1.000000	2	few_eyes
1	dwediigjit.mp4	1	4	1	-1.000000	0	
2	asvcrfdpnq.mp4	1	2	0	87.051709	1	eye_asymmetry
3	dntkzzzcdh.mp4	2	1	0	-1.000000	3	multiple_faces;few_eyes
4	dboxtiehng.mp4	1	3	1	-1.000000	0	
5	elginszwtk.mp4	0	0	0	-1.000000	4	no_face;few_eyes
6	cwbacdwrzo.mp4	1	2	0	95.036835	1	eye_asymmetry
7	afoovlsmtx.mp4	0	3	0	-1.000000	2	no_face
8	cmxcfkrjiv.mp4	0	1	0	-1.000000	4	no_face;few_eyes
9	aczrgyricp.mp4	1	4	0	-1.000000	0	

Рисунок 3.4 - Результати евристичного аналізу для 10 відеофрагментів

Другий підхід (рисунок 3.5) — класифікація на основі логістичної регресії — передбачає навчання моделі на основі ознак, сформованих із відео. Перед початком навчання усі відсутні значення були замінені нульовими, а дані поділено на тренувальну (80%) та тестову (20%) вибірки. Модель логістичної регресії реалізована засобами бібліотеки scikit-learn. Після навчання модель продемонструвала значно кращі результати класифікації: точність — 68,5%, прецизійність — 81%, повнота — 81%, F1-міра — 81%.

На рисунку 3.6 наведено приклад результатів класифікації для декількох відеофайлів, включаючи реальні мітки, ймовірності класу FAKE.

🔎 Logistic Regression Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	77
1	0.81	1.00	0.89	323
accuracy			0.81	400
macro avg	0.40	0.50	0.45	400
weighted avg	0.65	0.81	0.72	400

⚙️ Heuristic (Suspicion Score) Classification Report:

	precision	recall	f1-score	support
0	0.20	0.45	0.28	77
1	0.81	0.57	0.67	323
accuracy			0.55	400
macro avg	0.51	0.51	0.47	400
weighted avg	0.70	0.55	0.59	400

Рисунок 3.5 - Порівняння метрик логістичної регресії та евристики

	video	label_binary	predicted_fake	predicted_clf	predicted_proba
0	aagfhgtpmv.mp4	1	0	1	0.804130
1	aapnvogymq.mp4	1	1	1	0.832093
2	abarnvbtwb.mp4	0	1	1	0.725256
3	abofeumbvv.mp4	1	1	1	0.827438
4	abqwwspghj.mp4	1	0	1	0.785748
5	acifjvzvpm.mp4	1	1	1	0.822967
6	acqfdwsrhi.mp4	1	1	1	0.797575
7	acxnxvbsxk.mp4	1	0	1	0.719036
8	acxwigylke.mp4	1	0	1	0.719036
9	aczrgyricp.mp4	1	0	1	0.688206

Рисунок 3.6 - Порівняння класифікацій для прикладів відео

Візуальні приклади роботи детектора та ідентифікації облич (рисунок 3.7) на фреймах відео також ілюструють, як система реагує на нестандартні умови. Наприклад, у випадках з недостатньою кількістю очей або відсутністю облич, система фіксує відповідні аномалії та відображає їх як текстовий звіт і графічні обмежувальні рамки.

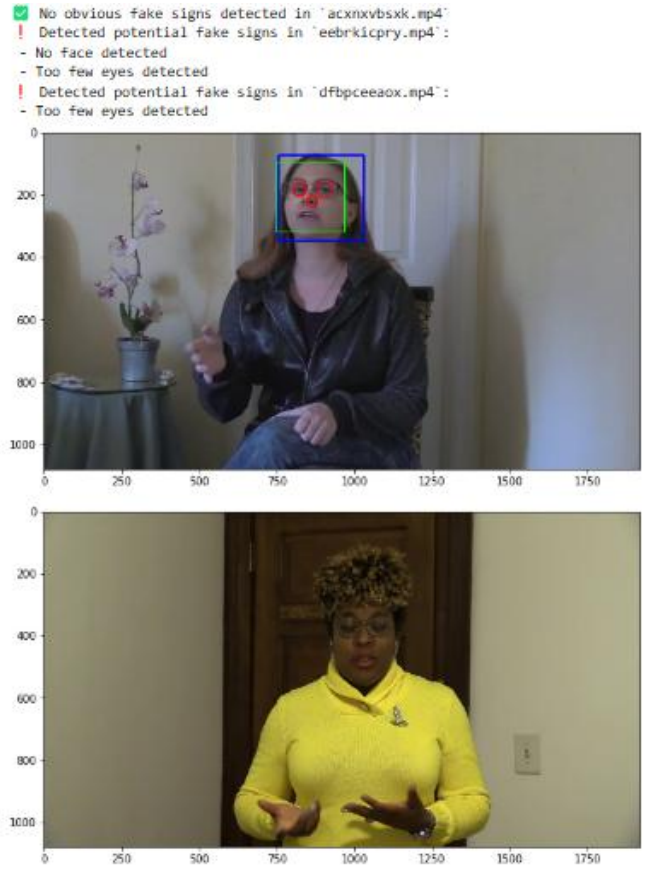


Рисунок 3.7 - Приклади фреймів відео з позначенням виявлених ознак та текстовими діагнозами

Таким чином, моделювання підтвердило доцільність комбінованого підходу: евристика може бути використана як попередній етап фільтрації, а навчена модель — як основний механізм прийняття рішень. Такий гібрид забезпечує баланс між швидкістю обробки та точністю класифікації діпфейкових відео.

ВИСНОВКИ

У даній кваліфікаційній роботі було:

1 Проведено ґрунтовний огляд сучасних методів виявлення діпфейків, з особливою увагою до мультимодальних та facial-based підходів. Аналіз показав, що найбільш ефективними є гібридні моделі, які здатні досягати точності до 98%.

2 Реалізовано архітектуру програмного модуля, що складається з трьох ключових рівнів: обробки даних, виявлення об'єктів та візуалізації результатів. Ця архітектура спроектована з урахуванням розширюваності та можливості повторного використання її компонентів.

3 Розроблено деталізований алгоритм обробки відеоданих. Він включає попереднє перетворення зображення, детекцію ознак обличчя, евристичну перевірку на наявність аномалій та подальше формування ознакового простору.

4 Запропоновано інноваційний метод оцінки підозрілості, який базується на кількісних показниках, що виявляються на перших кадрах відео. До таких показників належать кількість очей, наявність профільних облич та ступінь асиметрії.

5 Проведено порівняльний аналіз двох підходів до класифікації: евристичного та на основі логістичної регресії. Модель логістичної регресії продемонструвала значно вищу ефективність, досягнувши точності 68,5% та F1-міри 81%, чим перевершила евристичний метод.

6 Досліджено характеристики набору даних Deepfake Detection Challenge. Виявлено значний дисбаланс класів та високу різноманітність джерел, що є важливими факторами, які впливають на процес навчання та узагальнення моделей.

7 Результати цієї роботи чітко підтверджують ефективність використання аналізу ознак обличчя для виявлення діпфейків. Запропонований підхід має значний потенціал для застосування у системах цифрової безпеки,

зокрема в соціальних мережах, сервісах автентифікації та на різноманітних інформаційних платформах. У майбутньому можливим напрямом розвитку є інтеграція мультимодальних ознак (таких як аудіо, рух губ, міміка) та використання складніших трансформерних архітектур для подальшого підвищення точності та універсальності моделі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Coccomini, D. A., Messina, N., & Gennaro, C. (2022). *Combining efficientnet and vision transformers for video deepfake detection*. Springer. https://link.springer.com/chapter/10.1007/978-3-031-06433-3_19

2. Petmezas, G., Vanian, V., & Konstantoudakis, K. (2025). *Video deepfake detection using a hybrid CNN-LSTM-Transformer model for identity verification*. https://link.springer.com/article/10.1007/s11042-024-20548-6

3. Liu, P., Tao, Q., & Zhou, J. T. (2024). *Evolving from Single-modal to Multi-modal Facial Deepfake Detection: A Survey*. https://arxiv.org/abs/2406.06965

4. Salvi, D., Liu, H., Mandelli, S., Bestagini, P., & Zhou, W. (2023). *A robust approach to multimodal deepfake detection*. Journal of Imaging, 9(6), 122. https://doi.org/10.3390/jimaging9060122

5. Kaddar, B., Fezza, S. A., Akhtar, Z., & Hamidouche, W. (2024). *Deepfake detection using spatiotemporal transformer*. https://doi.org/10.1145/3643030

6. Wang, J., Wu, Z., Ouyang, W., Han, X., & Chen, J. (2022). *M2tr: Multi-modal multi-scale transformers for deepfake detection*. https://doi.org/10.1145/3512527.3531415

7. Soudy, A. H., Sayed, O., Tag-Elser, H., & Ragab, R. (2024). *Deepfake detection using convolutional vision transformers and convolutional neural networks*. https://link.springer.com/article/10.1007/s00521-024-10181-7

8. Wang, Z., Cheng, Z., Xiong, J., Xu, X., & Li, T. (2024). *A timely survey on vision transformer for deepfake detection*. https://arxiv.org/abs/2405.08463

9. Hashmi, A., Shahzad, S. A., & Lin, C. W. (2025). *AVTENet: A Human-Cognition-Inspired Audio-Visual Transformer-Based Ensemble Network for Video Deepfake Detection*. https://ieeexplore.ieee.org/abstract/document/10938399/

10. VP, S. E., & Dheepthi, R. (2024). *Llm-enhanced deepfake detection: Dense cnn and multi-modal fusion framework for precise multimedia authentication*. https://ieeexplore.ieee.org/abstract/document/10533511/

11. Jung, T., Kim, S., & Kim, K. (2020). *DeepVision: Deepfakes detection using human eye blinking pattern*. https://ieeexplore.ieee.org/abstract/document/9072088/

12. Liu, W., She, T., Liu, J., Li, B., & Yao, D. (2024). *Lips Are Lying: Spotting the Temporal Inconsistency between Audio and Visual in Lip-Syncing DeepFakes*. https://proceedings.neurips.cc/paper_files/paper/2024/hash/a5a5b0ff87c59172a13342d428b1e033-Abstract-Conference.html

13. Agarwal, S., & Farid, H. (2021). *Detecting deep-fake videos from aural and oral dynamics*. https://openaccess.thecvf.com/content/CVPR2021W/WMF/html/Agarwal_Detecting_Deep-Fake_Videos_From_Aural_and_Oral_Dynamics_CVPRW_2021_paper.html

14. Mazaheri, G. (2022). *Detection and localization of facial expression manipulations*. http://openaccess.thecvf.com/content/WACV2022/html/Mazaheri_Detection_and_Localization_of_Facial_Expression_Manipulations_WACV_2022_paper.html

15. Chakraborty, R., & Naskar, R. (2024). *Role of human physiology and facial biomechanics towards building robust deepfake detectors: A comprehensive survey and analysis*. https://www.sciencedirect.com/science/article/pii/S1574013724000613

16. Waseem, S., Bakar, S. A. R. S. A., Ahmed, B. A., & Omar, Z. (2023). *DeepFake on face and expression swap: A review*. https://ieeexplore.ieee.org/abstract/document/10285057/

17. Zamboni, J. (2025). *Deepfake detection via facial landmark motion analysis during person-specific word pronunciation*. https://reposit.haw-hamburg.de/handle/20.500.12738/16769

18. Agarwal, S. (2021). *Detecting synthetic faces by understanding real faces*. https://farid.berkeley.edu/downloads/publications/sathesis21.pdf

19. Saif, S., Tehseen, S., & Ali, S. S. (2024). *Fake news or real? Detecting deepfake videos using geometric facial structure and graph neural network*. https://www.sciencedirect.com/science/article/pii/S0040162524002671

20. Hashmi, A., Shahzad, S. A., Lin, C. W., & Tsao, Y. (2024). *Understanding Audiovisual Deepfake Detection: Techniques, Challenges, Human Factors and Perceptual Insights*. https://arxiv.org/abs/2411.07650

21. Lipianina-Honcharenko, K., Lendiuk, D., Ivaniush, A., Boguta, G., Soia, M., Yurkiv, K. An Intelligent Method for Recognizing Real and Generated Ukrainian-Language Voices. In 2024 5th IEEE KhPI Week on Advanced Technology (KhPIWeek 2024), Kharkiv, Ukraine, October 2024. IEEE Conference Proceedings, DOI: 10.1109/KHPIWEEK61434.2024.10877988.

22. Lipianina-Honcharenko, K., Yarych, V., Ivasechko, A., Filinyuk, A., Yurkiv, K., Lebid, T. Evaluating the Effectiveness of Attention-Gated-CNN-BGRU Models for Historical Manuscript Recognition in Ukraine. In CEUR Workshop Proceedings, 1st International Workshop of Young Scientists on Artificial Intelligence for Sustainable Development (AISD 2024), Ternopil, Ukraine, 10–11 May 2024.

23. Lipianina-Honcharenko, K., Soia, M., Yurkiv, K., Ivasechko, A. Evaluation of the Effectiveness of Machine Learning Methods for Detecting Disinformation in Ukrainian Text Data. In CEUR Workshop Proceedings, 7th International Workshop on Computer Modeling and Intelligent Systems (CMIS 2024), Zaporizhzhia, Ukraine, 3 May 2024.

24. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

25. Загальні методичні рекомендації з підготовки, оформлення, захисту та оцінювання кваліфікаційних робіт здобувачів вищої освіти першого (бакалаврського) і другого (магістерського) рівнів. Тернопіль: ЗУНУ, 2024. 83 с.

Додаток А

Візуалізація етапів виявлення діпфейків

```
def display_image_from_video(video_path):
    capture_image = cv.VideoCapture(video_path)
    ret, frame = capture_image.read()
    fig = plt.figure(figsize=(10,10))
    ax = fig.add_subplot(111)
    frame = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
    ax.imshow(frame)
```

Рисунок А.1 - Завантаження відео та обробка кадрів

```
class ObjectDetector():

    def __init__(self,object_cascade_path):

        self.objectCascade=cv.CascadeClassifier(object_cascade_path)

    def detect(self, image, scale_factor=1.3,
               min_neighbors=5,
               min_size=(20,20)):

        rects=self.objectCascade.detectMultiScale(image,
                                                  scaleFactor=scale_factor,
                                                  minNeighbors=min_neighbors,
                                                  minSize=min_size)
        return rects
```

Рисунок А.2 - Використання каскадів Гаара для виявлення об'єктів

```
def detect_objects(image, scale_factor, min_neighbors, min_size):

    image_gray=cv.cvtColor(image, cv.COLOR_BGR2GRAY)

    eyes=ed.detect(image_gray,
                   scale_factor=scale_factor,
                   min_neighbors=min_neighbors,
                   min_size=(int(min_size[0]/2), int(min_size[1]/2)))

    for x, y, w, h in eyes:
        cv.circle(image,(int(x+w/2),int(y+h/2)),(int((w + h)/4)),(0, 0,255),3)

    # deactivated due to many false positive
    #smiles=sd.detect(image_gray,
    #                 scale_factor=scale_factor,
    #                 min_neighbors=min_neighbors,
    #                 min_size=(int(min_size[0]/2), int(min_size[1]/2)))

    #for x, y, w, h in smiles:
    #    #detected smiles shown in color image
    #    cv.rectangle(image,(x,y),(x+w, y+h),(0, 0,255),3)


    profiles=pd.detect(image_gray,
                   scale_factor=scale_factor,
                   min_neighbors=min_neighbors,
                   min_size=min_size)

    for x, y, w, h in profiles:
        cv.rectangle(image,(x,y),(x+w, y+h),(255, 0,0),3)

    faces=fd.detect(image_gray,
                   scale_factor=scale_factor,
                   min_neighbors=min_neighbors,
                   min_size=min_size)

    for x, y, w, h in faces:

        cv.rectangle(image,(x,y),(x+w, y+h),(0, 255,0),3)

    fig = plt.figure(figsize=(10,10))
    ax = fig.add_subplot(111)
    image = cv.cvtColor(image, cv.COLOR_BGR2RGB)
    ax.imshow(image)
```

Рисунок А.3 - Інтеграція детекторів і побудова візуалізації

```
same_original_fake_train_sample_video = list(meta_train_df.loc[meta_train_df.original=='kgbkktcjxf.mp4'].index)
for video_file in same_original_fake_train_sample_video[1:4]:
    print(video_file)
    extract_image_objects(video_file)
```

```
byqzyxifza.mp4
cwrtyzndpx.mp4
cwwandrkus.mp4
```

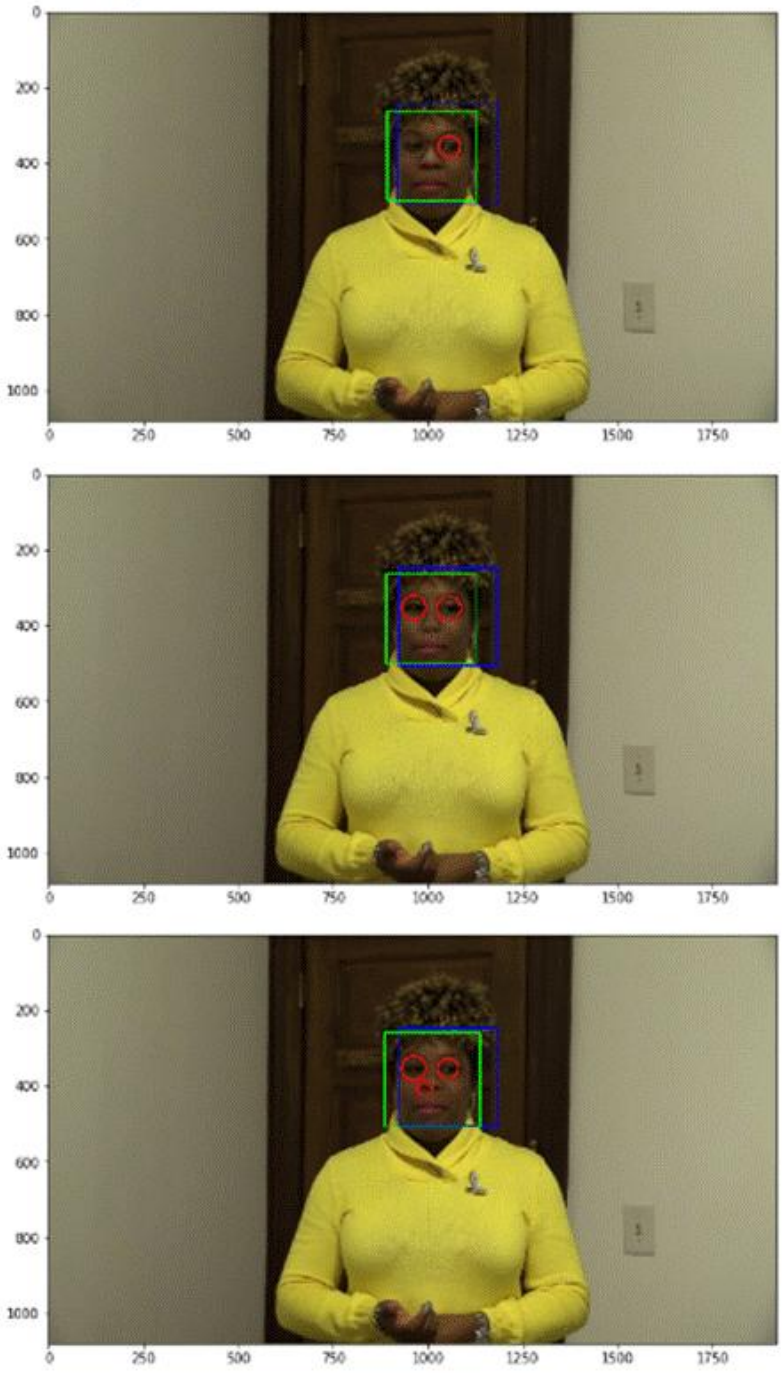


Рисунок А.4 - Обробка відео та інтеграція з метаданими створених на основі одного оригіналу

```python
def detect_fake_signs(video_file, video_set_folder=TRAIN_SAMPLE_FOLDER):

    video_path = os.path.join(DATA_FOLDER, video_set_folder, video_file)
    capture_image = cv.VideoCapture(video_path)
    ret, frame = capture_image.read()
    if not ret:
        print(f"Couldn't read frame from {video_file}")
        return

    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)

    faces = fd.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(50, 50))
    eyes = ed.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(20, 20))
    profiles = profile_detector.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(50, 50))

    issues = []

    if len(faces) == 0:
        issues.append("No face detected")
    elif len(faces) > 1:
        issues.append("Multiple faces detected")

    if len(eyes) < 2:
        issues.append("Too few eyes detected")
    elif len(eyes) > 4:
        issues.append("Too many eye-like regions detected")

    if len(profiles) > 0 and len(faces) == 0:
        issues.append("Profile face detected without frontal face")

    detect_objects(frame.copy(), scale_factor=1.3, min_neighbors=5, min_size=(50, 50))

    if issues:
        print(f"❗ Detected potential fake signs in `{video_file}`:")
        for issue in issues:
            print(" -", issue)
    else:
        print(f"✅ No obvious fake signs detected in `{video_file}`")

for video_file in fake_train_sample_video[:3]:
    detect_fake_signs(video_file)
```

Рисунок А.5 - Функція detect_fake_signs, яка виконує автоматизовану первинну перевірку відеозаписів на наявність ознак підробки

```python
def extract_features_from_video(video_file, video_set_folder=TRAIN_SAMPLE_FOLDER):

    video_path = os.path.join(DATA_FOLDER, video_set_folder, video_file)
    capture_image = cv.VideoCapture(video_path)
    ret, frame = capture_image.read()
    if not ret:
        return {'video': video_file, 'issue': 'Cannot read frame'}

    gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)

    faces = fd.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(50, 50))
    eyes = ed.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(20, 20))
    profiles = profile_detector.detect(gray, scale_factor=1.3, min_neighbors=5, min_size=(50, 50))

    score = 0
    issues = []

    if len(faces) == 0:
        score += 2
        issues.append("no_face")
    elif len(faces) > 1:
        score += 1
        issues.append("multiple_faces")

    if len(eyes) < 2:
        score += 2
        issues.append("few_eyes")
    elif len(eyes) > 4:
        score += 1
        issues.append("too_many_eyes")

    if len(profiles) > 0 and len(faces) == 0:
        score += 1
        issues.append("profile_only")

    eye_asymmetry = -1
    if len(eyes) == 2:
        x1, y1, _, _ = eyes[0]
        x2, y2, _, _ = eyes[1]
        dx = abs(x1 - x2)
        dy = abs(y1 - y2)
        eye_asymmetry = np.sqrt(dx**2 + dy**2)
        if eye_asymmetry > 80:  # евристика
            score += 1
            issues.append("eye_asymmetry")

    return {
        'video': video_file,
        'n_faces': len(faces),
        'n_eyes': len(eyes),
        'n_profiles': len(profiles),
        'eye_asymmetry': eye_asymmetry,
        'suspicion_score': score,
        'issues': ";".join(issues)
    }
```

Рисунок А.6 - Функція виявлення ознак підробки у відео

```python
def build_suspicion_dataset(video_list, folder=TRAIN_SAMPLE_FOLDER):
    feature_rows = []
    for video_file in tqdm(video_list):
        features = extract_features_from_video(video_file, folder)
        if features is not None:
            feature_rows.append(features)
    return pd.DataFrame(feature_rows)
```

Рисунок А.7 – Функція створення ознакового набору з відео

```python
from tqdm import tqdm

video_list_10 = train_list[:10]

df_10 = build_suspicion_dataset(video_list_10, folder=TRAIN_SAMPLE_FOLDER)

import pandas as pd
from IPython.display import display

display(df_10)
```

```
100%|██████████| 10/10 [00:02<00:00,  3.68it/s]
```

	video	n_faces	n_eyes	n_profiles	eye_asymmetry	suspicion_score	issues
0	eivxffliio.mp4	1	1	1	-1.000000	2	few_eyes
1	dwediigjit.mp4	1	4	1	-1.000000	0	
2	asvcrfdpnq.mp4	1	2	0	87.051709	1	eye_asymmetry
3	dntkzzzcdh.mp4	2	1	0	-1.000000	3	multiple_faces;few_eyes
4	dboxtiehng.mp4	1	3	1	-1.000000	0	
5	elginszwtk.mp4	0	0	0	-1.000000	4	no_face;few_eyes
6	cwbacdwrzo.mp4	1	2	0	95.036835	1	eye_asymmetry
7	afoovlsmtx.mp4	0	3	0	-1.000000	2	no_face
8	cmxcfkrjiv.mp4	0	1	0	-1.000000	4	no_face;few_eyes
9	aczrgyricp.mp4	1	4	0	-1.000000	0	

Рисунок А.8 –Формування датафрейму ознак для 10 відеофайлів

```python
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

all_video_files = list(meta_train_df.index)
df_all = build_suspicion_dataset(all_video_files, folder=TRAIN_SAMPLE_FOLDER)

df_all['label'] = df_all['video'].map(meta_train_df['label'])

df_all['label_binary'] = df_all['label'].map({'REAL': 0, 'FAKE': 1})

df_all['predicted_fake'] = df_all['suspicion_score'].apply(lambda x: 1 if x >= 2 else 0)

cm = confusion_matrix(df_all['label_binary'], df_all['predicted_fake'])

plt.figure(figsize=(5, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['REAL', 'FAKE'], yticklabels=['REAL', 'FAKE'])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix: Suspicion Score Heuristic")
plt.show()
```

```
100%|██████████| 400/400 [01:53<00:00,  3.52it/s]
```

Рисунок А.9 –Матриця плутанини для евристичного методу на основі підозрілості

```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

feature_cols = ['suspicion_score', 'n_faces', 'n_eyes', 'n_profiles', 'eye_asymmetry']
X = df_all[feature_cols].fillna(0)
y = df_all['label_binary']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

clf = LogisticRegression(max_iter=1000)
clf.fit(X_train, y_train)

df_all['predicted_clf'] = clf.predict(X)
df_all['predicted_proba'] = clf.predict_proba(X)[:, 1]  # ймовірність FAKE

comparison_df = df_all[['video', 'label_binary', 'predicted_fake', 'predicted_clf', 'predicted_proba']]
display(comparison_df.head(10))

print("🔎 Logistic Regression Classification Report:")
print(classification_report(df_all['label_binary'], df_all['predicted_clf']))
print("\n⚙️ Heuristic (Suspicion Score) Classification Report:")
print(classification_report(df_all['label_binary'], df_all['predicted_fake']))
```

Рисунок А.10 –Побудова та оцінка моделі логістичної регресії для виявлення діпфейків

	label	split	original
Total	400	400	323
Most frequent item	FAKE	train	meawmsgiti.mp4
Frequence	323	400	6
Percent from total	80.75	100	1.858

Рисунок А.11 - Частотний розподіл значень ознак label, split та original у метаданих тренувального набору

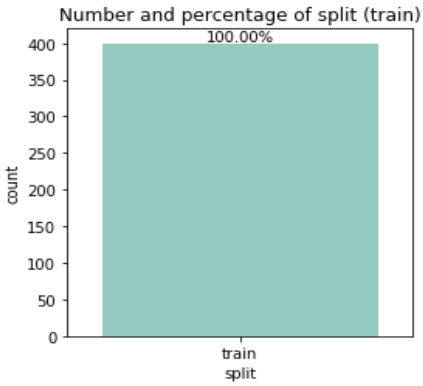


Рисунок А.12 - Первинний аналіз метаданих тренувального набору даних

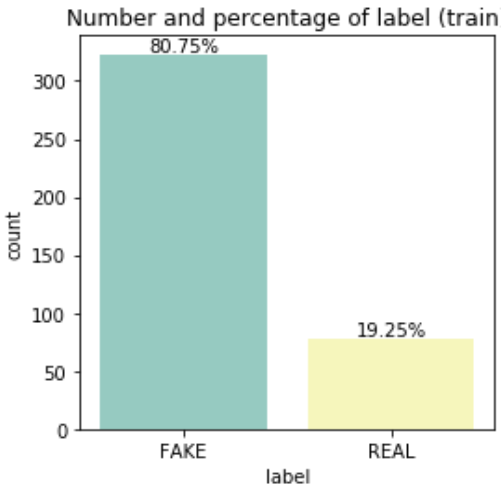


Рисунок А.13 - Розподіл міток в тренувальному наборі даних

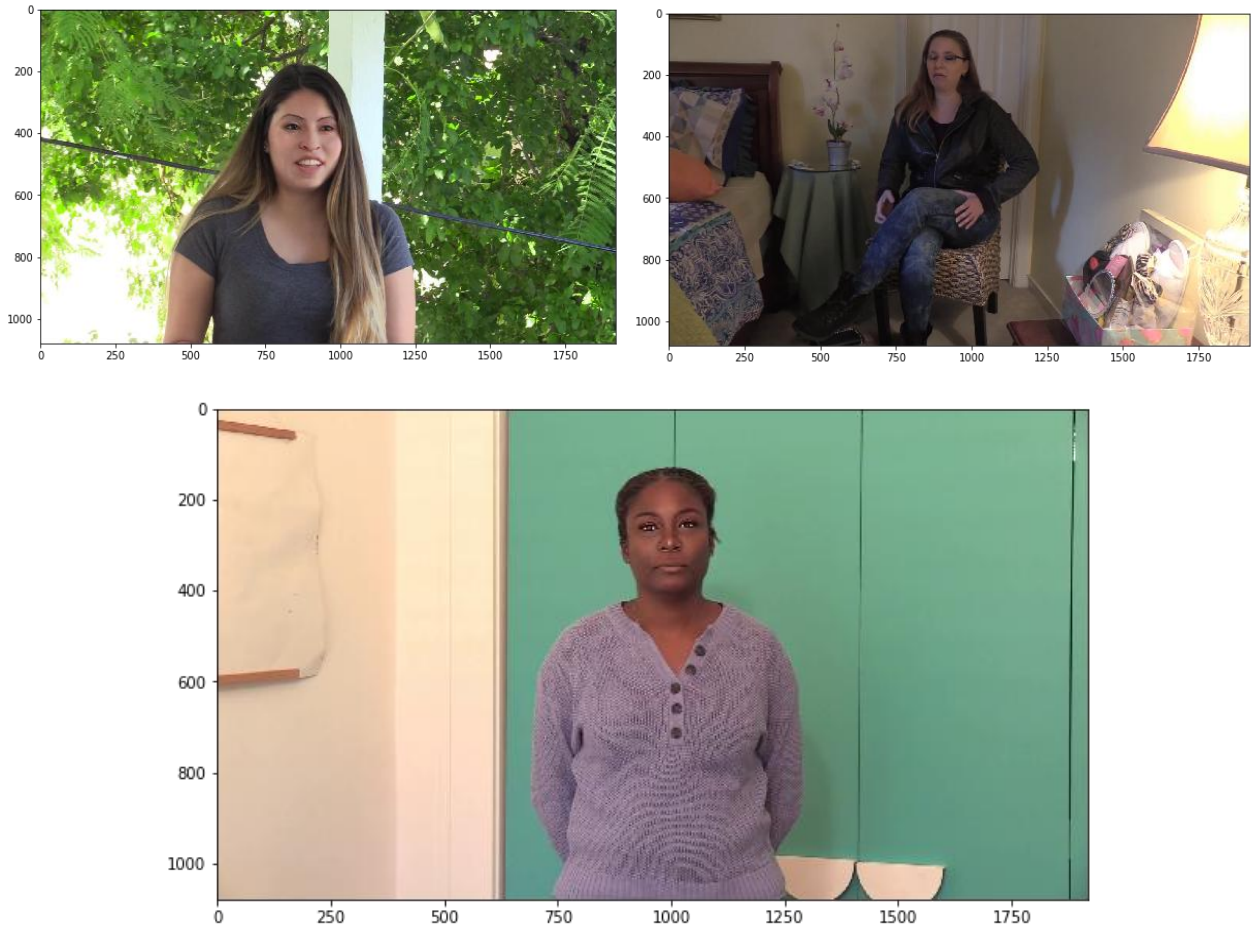


Рисунок А.14 – Перші кадри з підроблених відео

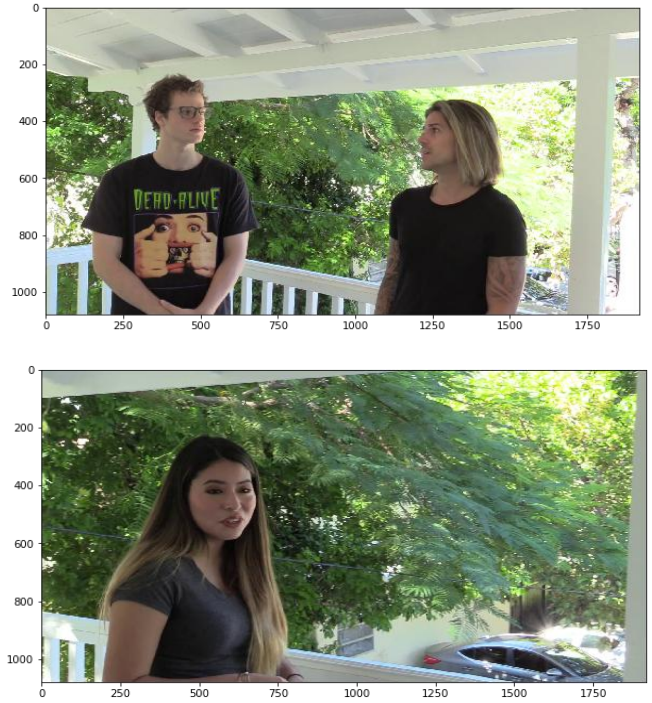


Рисунок А.15 - Кадри з справжніх відео

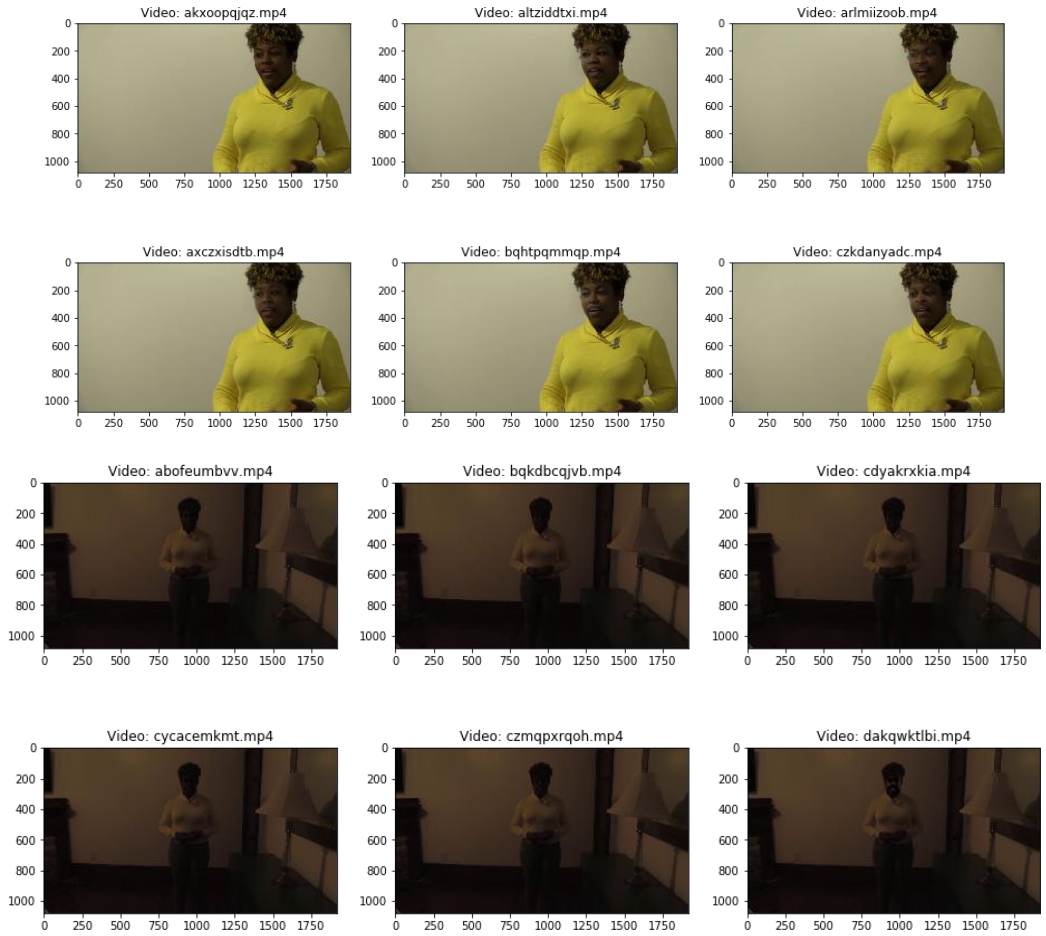


Рисунок А.16 - Кадри з підроблених відео, що походять з одних джерел

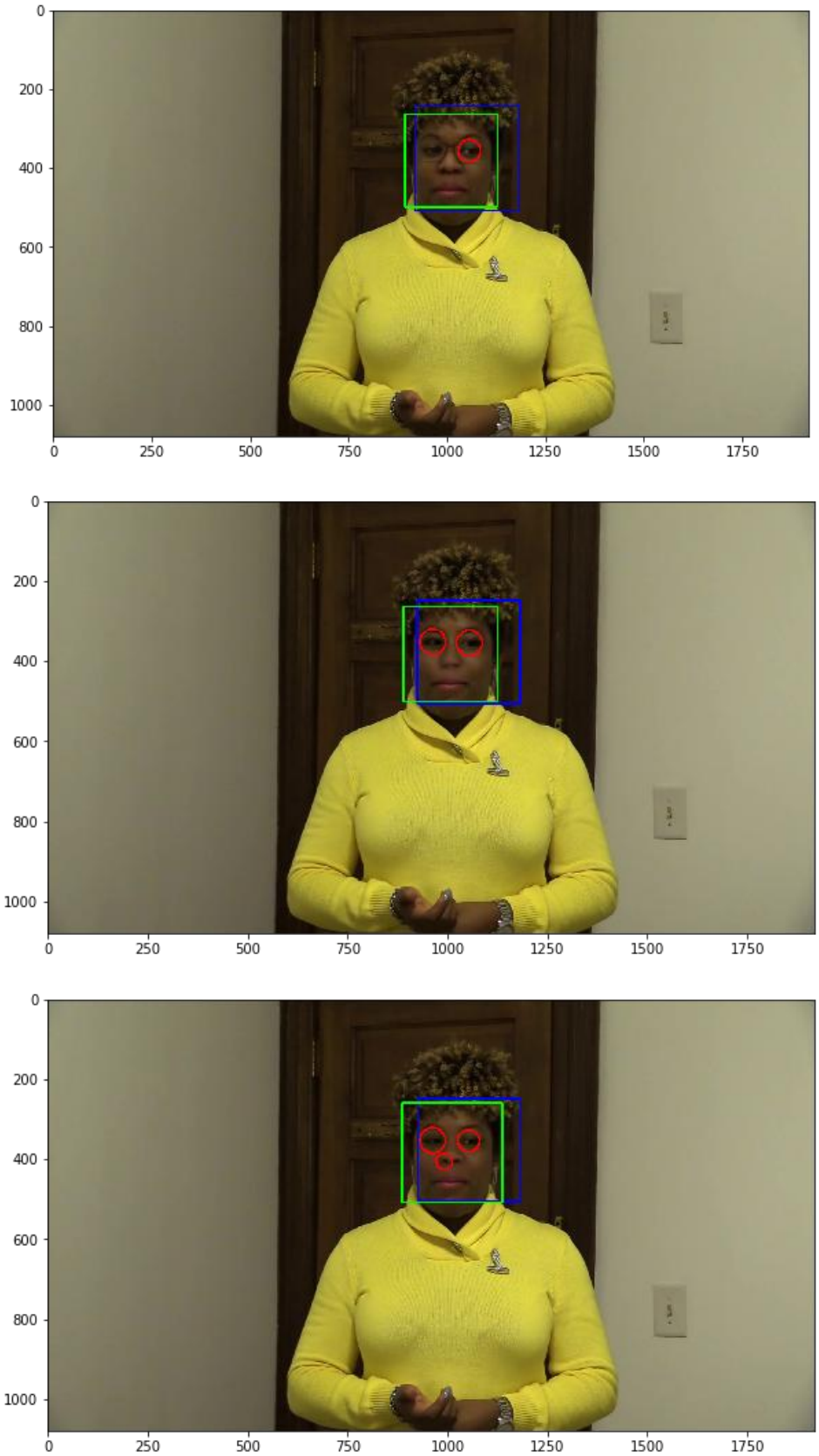
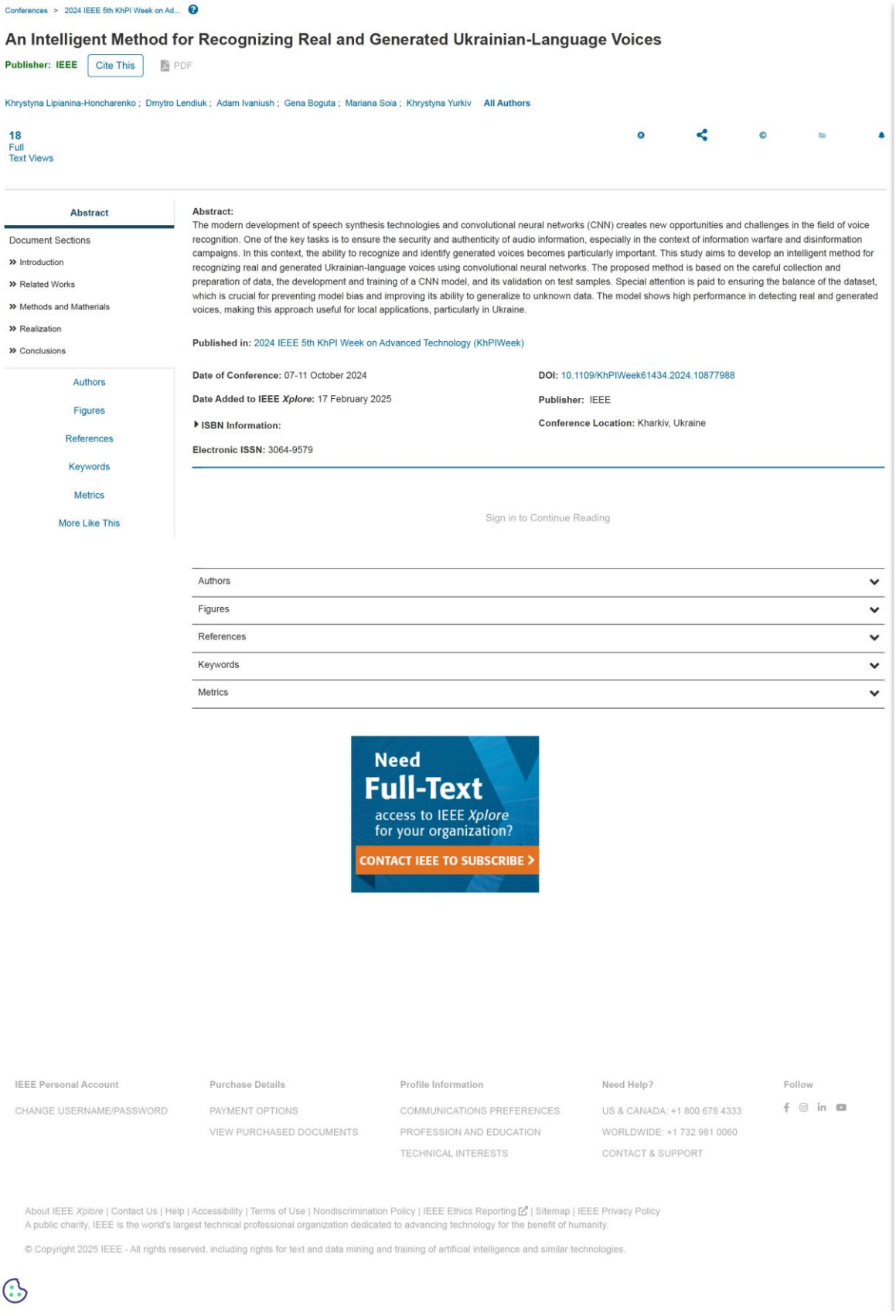


Рисунок А.17 - Результати детекції облич та об'єктів за допомогою каскадів Гаара, що походять від спільного оригінального джерела

Додаток Б

Апробація отриманих результатів



Conferences > 2024 IEEE 5th KhPI Week on Ad...

An Intelligent Method for Recognizing Real and Generated Ukrainian-Language Voices

Publisher: IEEE | Cite This | PDF

Khrystyna Lipianina-Honcharenko ; Dmytro Lendiuk ; Adam Ivaniush ; Gena Boguta ; Mariana Soia ; Khrystyna Yurkiv **All Authors**

18
Full
Text Views

Abstract

Document Sections

» Introduction
» Related Works
» Methods and Matherials
» Realization
» Conclusions

Authors
Figures
References
Keywords
Metrics
More Like This

Abstract:
The modern development of speech synthesis technologies and convolutional neural networks (CNN) creates new opportunities and challenges in the field of voice recognition. One of the key tasks is to ensure the security and authenticity of audio information, especially in the context of information warfare and disinformation campaigns. In this context, the ability to recognize and identify generated voices becomes particularly important. This study aims to develop an intelligent method for recognizing real and generated Ukrainian-language voices using convolutional neural networks. The proposed method is based on the careful collection and preparation of data, the development and training of a CNN model, and its validation on test samples. Special attention is paid to ensuring the balance of the dataset, which is crucial for preventing model bias and improving its ability to generalize to unknown data. The model shows high performance in detecting real and generated voices, making this approach useful for local applications, particularly in Ukraine.

Published in: 2024 IEEE 5th KhPI Week on Advanced Technology (KhPIWeek)

Date of Conference: 07-11 October 2024
DOI: 10.1109/KhPIWeek61434.2024.10877988
Date Added to IEEE *Xplore*: 17 February 2025
Publisher: IEEE
ISBN Information:
Conference Location: Kharkiv, Ukraine
Electronic ISSN: 3064-9579

Sign in to Continue Reading

Authors
Figures
References
Keywords
Metrics

Need Full-Text access to IEEE *Xplore* for your organization?
CONTACT IEEE TO SUBSCRIBE >

IEEE Personal Account
CHANGE USERNAME/PASSWORD

Purchase Details
PAYMENT OPTIONS
VIEW PURCHASED DOCUMENTS

Profile Information
COMMUNICATIONS PREFERENCES
PROFESSION AND EDUCATION
TECHNICAL INTERESTS

Need Help?
US & CANADA: +1 800 678 4333
WORLDWIDE: +1 732 981 0060
CONTACT & SUPPORT

Follow

About IEEE *Xplore* | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | IEEE Ethics Reporting | Sitemap | IEEE Privacy Policy
A public charity, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

Vol-3716
urn:nbn:de:0074-3716-4

AISD 2024
Artificial Intelligence for Sustainable Development 2024

Proceedings of the First International Workshop of Young Scientists on Artificial Intelligence for Sustainable Development

Ternopil, Ukraine, May 10-11, 2024.

Edited by

Mykola Dyvak*
Oleh Pitsun **

* West Ukrainian National University, Ternopil, Ukraine
** West Ukrainian National University, Ternopil, Ukraine

Table of Contents

2024-06-11: submitted by Oleh Pitsun, metadata incl. bibliographic data published under Creative Commons CC0
2024-07-02: published on CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073) |valid HTML5|

Evaluating the Effectiveness of Attention-Gated-CNN-BGRU Models for Historical Manuscript Recognition in Ukraine

Khrystyna Lipianina-Honcharenko[1], Volodymyr Yarych[1], Andrii Ivasechko[1], Anatoliy Filinyuk[1], Khrystyna Yurkiv[1], Tetian Lebid [1]

[1] *West Ukrainian National University, Lvivska str., 11, Ternopil, 46000, Ukraine*

Abstract

This research focuses on evaluating the effectiveness of the Attention-Gated-CNN-BGRU model for Handwritten Text Recognition from historical documents, specifically from the archive of Khmelnytskyi Oblast, written in Ukrainian and Russian languages between 1861 and 1913. The methodology involved preprocessing, data augmentation, deep learning with attention mechanism, and expert assessment. The obtained results showed an average percentage of correctly recognized characters at 71.7%, demonstrating the high effectiveness of the model. A strong negative correlation between text complexity and recognition accuracy underscores the need for further improvement in Optical Character Recognition technologies. The main direction of future research will be adapting the model for recognizing texts written in the Ukrainian language using the Latin alphabet, which is crucial for preserving Ukraine's cultural heritage.

Keywords

Historical documents, Optical Character Recognition, Handwritten Text Recognition, deep learning

1. Introduction

The modern world of digital technologies opens new horizons for the preservation and study of historical documents, offering unique tools for exploring cultural heritage. One of the key directions in this field is the development and application of OCR systems, which automate the process of converting image-based text into machine-readable format. This, in turn, facilitates easier access to historical documents, their analysis, and interpretation. However, the characteristics of historical manuscripts, such as variability in writing styles, degree of preservation, and material erosion, pose challenges for researchers that require the development of specialized OCR algorithms and methods.

This scientific article is dedicated to analyzing the effectiveness of the Attention-Gated-CNN-BGRU model [1], specifically developed for text recognition from manuscripts. The research is based on a carefully curated dataset of documents from the state archive of Khmelnytskyi Oblast, covering the years from 1861 to 1919 and various funds reflecting the socio-historical aspects of the region during the specified period. The main focus of the research is on determining the accuracy of the OCR model in the context of variability in manuscript complexity, evaluating the impact of writing styles and document preservation on recognition quality, as well as analyzing potential directions for algorithm optimization.

The First International Workshop of Young Scientists on Artificial Intelligence for Sustainable Development; May 10-11, 2024, Ternopil, Ukraine

); xrustya.com@gmail.com (K. Lipianina-Honcharenko); v.yarych@wunu.edu.ua (V. Yarych); andrewivasechko@gmail.com (A. Ivashechkoafilinyuk@ukr.net(A. Filinyuk) kh.yurkiv@wunu.edu.ua (K. Yurkiv); Liebid89@gmail.com (T. Lebid).

); 0000-0002-2441-6292 (K. Lipianina-Honcharenko); 0000-0003-4455-952X (V. Yarych); 0009-0002-8623-7828 (A. Ivashechko0000-0002-2659-114X (A. Filinyuk); 0009-0007-4917-3251 (K. Yurkiv); 0009-0005-3413-9064 (T. Lebid);

Given the importance of preserving historical heritage and the development of digital humanities, the results of this research aim not only to demonstrate the potential of modern OCR technologies in historical text analysis but also to provide valuable insights for further improvement of digital humanities tools.

The formulation of the problem. In the context of the development of digital technologies and the study of cultural heritage, there arises the problem of effectively recognizing text from historical manuscripts using OCR systems. This problem arises due to the diversity of writing styles, differences in the preservation state of documents and their materials, complicating the process of automated conversion of images into machine-readable form. Such technical challenges create the necessity for the development and enhancement of specialized OCR algorithms to effectively process historical manuscripts.

This article presents a method for evaluating the effectiveness of the Attention-Gated-CNN-BGRU model for text recognition from historical manuscripts, based on deep learning with attention mechanism. Chapter 2 discusses a review of related works; Chapter 3 describes the research methodology, including dataset description and document recognition approach; Chapter 4 is dedicated to the implementation of the algorithm and detailed analysis of the obtained results. Chapter 5 presents the research conclusions summarizing the main findings and emphasizing the prospects for further research in this field.

2. Related work

The development of artificial intelligence (AI) technologies is one of the key and significant trends of the present day. This is primarily explained by the ability of AI, or more accurately "computational" or "electronic" intelligence, to learn and self-learn, recognize and synthesize language and images. Worldwide practice already has results of AI activity in the field of historical research. For example, Swedish scientists were able to reconstruct a complex handwritten text from the 18th century, stored in the Swedish National Archives [2]. The object of this research was the decrees on freedom of the press from 1766, which were written in Latin script. Such breakthroughs in science prompt us to explore the capabilities of AI in interpreting handwritten texts from Ukrainian archives, which were written in Cyrillic.

In Ukraine, specialists from not only the Institute of Artificial Intelligence Problems of the Ministry of Education and Science of Ukraine and the National Academy of Sciences of Ukraine are working on the development and improvement of AI capabilities, but also scientific and pedagogical workers and researchers from departments and divisions of AI, computer science and applied mathematics, computer technologies and economic cybernetics, innovative technologies, intellectual information systems, mathematical modeling, AI systems, information security of many higher education institutions and research institutes of Ukraine. A leading role in this field is played by the team of the Research Institute of Intelligent Computer Systems of the Western Ukrainian National University (Ternopil) and the V.M. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine (Kyiv).

Modern research in the field of AI is actively developing, especially in the area of document interpretation, where information technologies are used for analysis, search, and interpretation of relevant texts. Significant important informational potential for our research[3] useful information on the creation of archival collections of websites within the framework of initiative documentation at the Central State Electronic Archives is described in the research [4]. In research [5] characterized the process of digitizing archival documents in foreign countries. The topical issue of the role of digital sources in the research [6]. In research [7] predicted the growing role of archive digitization in modern society.

Innovative approaches to preserving cultural heritage through the use of image recognition and augmented reality, as seen in the researchs [8-10], illuminate the development of the intersection of technology and history. These methodologies resonate with the fundamental principles of our research, which applies Attention-Gated-CNN-BGRU models for recognizing historical manuscripts in Ukraine, demonstrating the versatility and potential of deep learning in

various fields. Similar to how [11, 12], utilized deep neural networks, and researchs [13, 14], applied multilevel data encoding, our study employs advanced machine learning algorithms to open new perspectives in the analysis and preservation of cultural heritage, affirming the critical role of technology in safeguarding and researching our collective past.

Research [15] explored offline handwritten text recognition (HTR) with reduced training datasets, proposing a model trained on crossed-out text to effectively recognize such words without compromising accuracy. In [16], methods for classifying handwritten words into a digital format were investigated, combining direct word classification using Convolutional Neural Networks (CNN) and character segmentation with Long Short-Term Memory (LSTM) networks. Additionally, the study in [17] addressed handwritten text conversion and storage in ASCII format, covering preprocessing, feature extraction, and classification using deep learning methods. The research in [18] enhanced HTR performance on lines with crossed-out words, while the approach in [19] proposed outperformed traditional OCR systems. The integration of HTR into OCR systems was discussed in [20], alongside outlining an HTR competition focusing on historical documents [21]. Effective practices in HTR, including basic architectures and datasets like IAM and RIMES, were described in [22], while the importance of image processing in handwritten text detection was emphasized in [23], showing potential in forgery protection and handwriting analysis. Lastly, the Attention-Gated-CNN-BGRU model [1] for Ukrainian HTR is freely available, trained on historical Ukrainian texts provided by researchers and libraries.

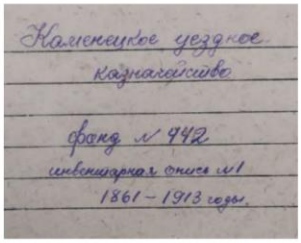
The Attention-Gated-CNN-BGRU model [1] for Ukrainian HTR is freely available, trained on historical Ukrainian texts and provided by researchers and libraries.The analysis of known solutions in the field of HTR, including the application of models like CRNN, and approaches to text classification and segmentation, underscores significant progress in this domain. However, this study focuses on analyzing the effectiveness of the Attention-Gated-CNN-BGRU model [1] for recognizing text from historical manuscripts, distinguishing itself through the use of attention mechanisms for detailed analysis of contextual relationships between characters. The approach in this research demonstrates improvements in recognizing complex historical texts, especially with crossed-out words and conditions of high text complexity, making it akin in concept to the works of Jose Carlos Aradillas et al. [15], but with an additional focus on adaptation to the specificity of Ukrainian handwritten text. This sets apart this study from others, providing a new approach to addressing the problem of text recognition in historical documents using deep learning and attention mechanisms, opening up prospects for further advancements in this field. The analysis of the text written in Ukrainian is given in the work [25-26].

3. Methodology

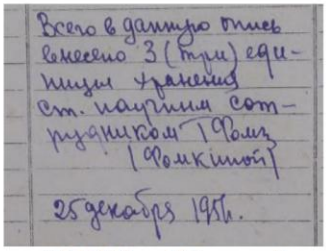
3.1. Dataset Description

For this study, documents of varying readability levels were collected from the State Archives of Khmelnytskyi Oblast. Among the proposed documents, digitized descriptions of ancient records from the following funds were taken: Fund 442, Kamianets-Podilskyi County Treasury for 1861-1913; Fund 507, Office of the Chief of the South-Western Customs District for 1907-1913; Fund 596, Podilia Branch of Princess Tetiana Mykolaivna's Committee for Providing Temporary Assistance to Victims of Military Actions for 1914-1915; Fund 598, Investigative Court of the 2nd Division of Kamianets-Podilskyi County for 1875-1880; Fund 616, Kamianets-Podilskyi County Military Affairs for 1884-1919; as well as Fund 309, Isakovets Customs for 1931-1915, written in Ukrainian and Russian languages.

The dataset [24] for recognizing ancient handwritten text consists of 75 PNG-format images, examples of which are shown in Figure 1.



A) Easy to read



B) Difficult to read

Figure 1: Examples of texts

3.2. Description of Document Recognition Approach

This research focuses on analyzing the effectiveness of the Attention-Gated-CNN-BGRU model for recognizing text from historical manuscripts. Modern challenges in OCR in historical documents include a variety of writing styles, document preservation levels, and character variability. This approach involves applying deep learning with attention to the context of characters and their interactions within words or text fragments, which enhances recognition accuracy compared to traditional methods. The implementation was done using the Python programming language. The approach consists of the following stages:

Stage 1. Data preprocessing:

1.1. Digital transformation: Each text sample I from historical manuscripts is converted into a digital format through scanning or photography, where I is represented as a matrix of pixels $I(x, y)$, where x and y are pixel coordinates.

1.2. Normalization: Applying normalization to each image I to standardize sizes and color intensities. The normalized image can be represented as

$$Inorm = f(I),$$

where f is the normalization function.

1.3. Data augmentation: Generating new samples Iaug from Inorm using transformations such as rotation, scaling, shifting, etc., to increase data diversity:

$$Iaug = g(Inorm),$$

where g is the augmentation function.

Stage 2. The Attention-Gated-CNN-BGRU model combines CNN for efficient visual feature extraction with gated recurrent units (GRU) and attention mechanism for modeling dependencies between characters. Step-by-step, this is presented as follows:

2.1. CNN: Using CNN to extract visual features $\phi(Iaug)$ from images, where ϕ is the feature extraction function implemented using CNN.

2.2. GRU and attention mechanism: Processing feature sequences $\phi(Iaug)$ using GRU and attention mechanism to model character dependencies and consider context:

$$\psi(\phi(Iaug)),$$

where ψ is the function implementing GRU and attention mechanism.

Stage 3. Model validation:

3.1. Word Error Rate (WER):

$$\text{WER}=\frac{S+D+I}{N}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of words in the correct text.

3.2. Character Error Rate (CER):

$$CER = \frac{S + D + I}{M}$$

where M is the total number of characters in the correct text.

3.3. Levenshtein CER: Using the Levenshtein distance to calculate CER, where the Levenshtein distance between two strings is the minimum number of single-element edits (insertions, deletions, substitutions) required to transform one string into another.

Stage 4. Results analysis:

)

4.1. Performance evaluation: Analyzing the distribution of WER and CER errors to evaluate the model's effectiveness in recognizing text from historical manuscripts. Statistical methods are used to compare the model results with baseline indicators.
4.2. Impact of attention mechanism: Evaluating the contribution of the attention mechanism to the model's ability to identify complex characters and their contextual relationships through detailed analysis of corrected errors and improvements in recognition accuracy.
Further, this approach is described as an algorithm (Figure 2) for evaluating the effectiveness of the Attention-Gated-CNN-BGRU model for recognizing text from historical manuscripts, starting with data preparation, which includes data collection, digital transformation, and dataset augmentation from manuscripts. Next, model configuration involves integrating convolutional neural networks and gated recurrent units with attention mechanism for text analysis. Model validation is done using independent test datasets, and performance evaluation is based on word and character error rates. Results analysis includes comparison with baseline indicators, detailed recognition analysis, and identification of directions for further model improvement.

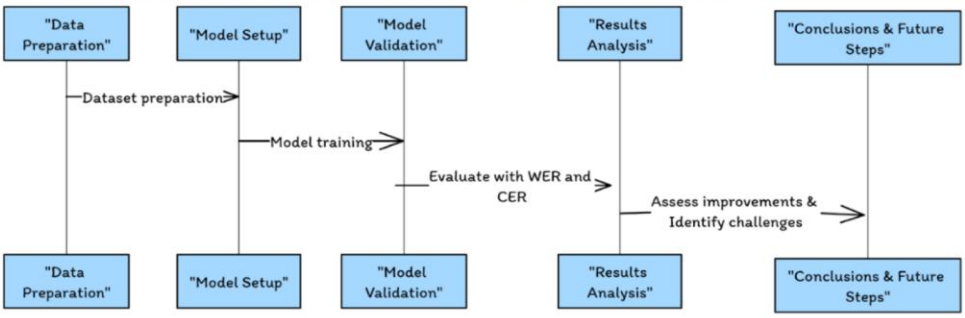


Figure 2. Structure of Document Recognition Approach

3.3 Approach to Expert Evaluation

Within the framework of this study, an expert evaluation methodology was used to assess the effectiveness of OCR models on historical manuscript materials. Five qualified experts analyzed the document samples, evaluating the difficulty of text recognition, the number of correctly recognized characters, the total number of recognized characters, and the total number of characters in each document. The approach proposed by the authors for this research for evaluation consists of the following stages:
Stage 1. Data preparation for analysis.
1.1. Collection of evaluations from experts ($EX1, EX2, EX3, EX4, EX5$) based on the following criteria:
1.1.1. Assessment of text recognition difficulty from 1 to 5, where 1 - very easy, 5 - very difficult.
1.1.2. Correctly recognized characters: the number of characters that were correctly recognized.
1.1.3. Recognized characters: the total number of recognized characters.
1.1.4. Total characters: the total number of characters in the original document.
1.2. Calculation of indicators:
1.2.1. Average assessment of text recognition difficulty ($SORT$):

$$SORT = \frac{\sum_{i=1}^{n} Assessment_i}{n}$$

where n is the number of experts, and Ratingi is the rating from expert i.
1.2.2. Average percentage of correctly recognized symbols ($SVPRS$):

$$SVPRS = \frac{\sum_{i=1}^{n} \left(\frac{Right\_recongised\_symbols_i}{All\_symbols} \times 100 \right)}{n}$$

1.2.3. Average number of recognized symbols (ANRS)

$$SKRS = \frac{\sum_{i=1}^{n} Recognised\_symbols_i}{n}$$

Stage 2. Analysis of the results:

2.1. Analysis of the overall effectiveness of the OCR model:

2.1.1. Calculating the average values for $SORT$, $SVPRS$, and $SKRS$ for all documents allows for the evaluation of the overall effectiveness of the OCR model.

2.1.2. Measuring the dispersion and standard deviation for each indicator helps understand the diversity of expert ratings and the variability of recognition results.

2.2. Impact of text complexity on recognition quality: Using correlation analysis between SORT and SVPRS helps determine how text complexity affects recognition quality. A positive correlation may indicate that as the complexity of the text increases, recognition efficiency decreases.

Stage 3. Interpretation of the obtained results:

3.1. The overall efficiency of the OCR model is determined through the analysis of the average values of $SORT, SVPRS$, and $SKRS$. High values of $SORT$ and $SKRS$ with low $AATRD$ indicate the model's ability to adapt to various conditions of handwritten text.

3.2. The results of analyzing the impact of text complexity on recognition quality provide insights into the limitations of the OCR model and directions for further improvement. Enhancing recognition accuracy under conditions of high text complexity may become a key aspect of model optimization.

The algorithm (Figure 3) for the experimental evaluation of OCR models on historical manuscripts begins with collecting expert ratings, where experts analyze text samples from manuscripts based on defined criteria such as text recognition complexity and the number of correctly recognized symbols. The second stage involves analyzing the overall efficiency of the models by calculating average values, dispersion, standard deviation of indicators, and conducting correlation analysis between text complexity and recognition quality. In the final stage, the obtained results are interpreted, allowing for the assessment of the overall efficiency of the models and determining optimization directions to improve recognition accuracy under conditions of high text complexity.

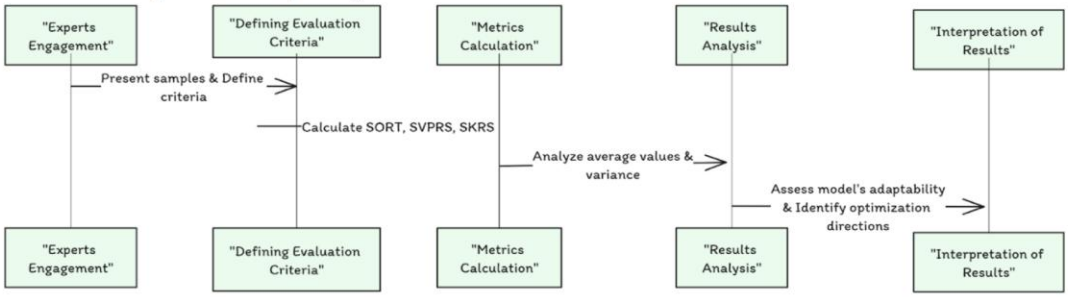


Figure 3. Structure of the OCR model effectiveness assessment approach

4. Result

In this section, a detailed analysis of the experiment results is presented, allowing for the evaluation of the effectiveness of applying OCR models to historical manuscripts, as described in section 3.1.

Table 1 below displays the results of recognizing the first five fragments of historical manuscripts using the OCR model. The data include links to the photos of the originals as well as the text recognized by the model. A comprehensive analysis of the entire dataset and detailed research results are available for review on GitHub [24], where an in-depth investigation of the OCR model's effectiveness on various fragments of historical documents is presented.

Upon completing the initial analysis of the text recognition results, a comprehensive expert evaluation was conducted, as detailed in section 3.3 of the documentation. The expertise involved

a thorough examination of text complexity, recognition quality, and symbol identification accuracy, aimed at gaining a deeper understanding of the effectiveness of applying OCR models to historical documents. Prominent experts involved in the analysis included Senior Lecturer of the Department of Information and Socio-Cultural Activities at the Western Ukrainian National University, Volodymyr Yarych, Doctor of Historical Sciences Anatoliy Filinyuk from Ivan Ogienko Kamianets-Podilskyi National University, as well as candidates of Historical Sciences Serhiy Sydoruk and Serhiy Trubchaninov. Additionally, Valentina Filinyuk, a candidate of Philological Sciences and Associate Professor at Khmelnytsky Humanitarian-Pedagogical Academy, contributed to the expert assessment. Their professional approach and deep knowledge facilitated the identification of key aspects for further improvement of OCR technologies in the context of working with historical texts.

Table 1.
Results of recognizing the first 5 fragments

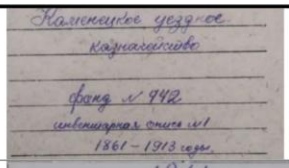
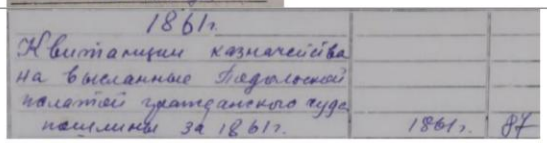
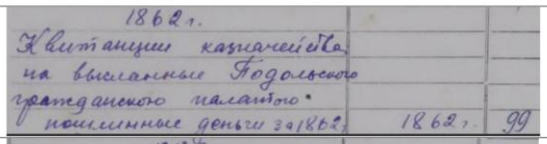
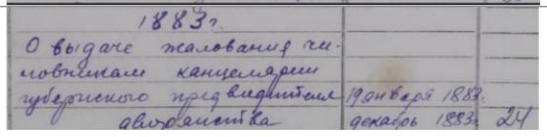
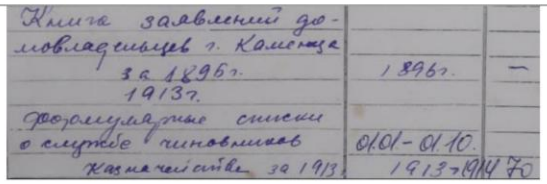
№ doc.	Link to the original photo:	Recognized text:
1		Каменецкое уездное. казначейство Фанд N 442 инвентарная опись л! 1861 – 1913 года.
2		Каннъ 1861г. Нвитанции казначенива на высланные Подолоской палатой гражданского чуда пошлины за 1861г. 1861г. 87
3		1862г. Хвитанции казначействе на высланные Подольского гражданскою палатого 99
4		1883г. О выдаче жалования чи- ловникам канцелярли ГанБаря 1883. губернского предвидинеля 24 декабрь 1883г дворянства
5		1896г. Ннига заявлений до- мовладеньцев г. Каменца 1896г. 3 а 1896г. 1913г фозомумярные списки о службе чиновников да-а 10 казначействе за 1913

Table 2 provides a statistical overview of the collected data, including the mean, variance, and standard deviation for the number of successfully recognized characters, the average complexity rating of texts, and the average accuracy percentage of recognition.

The analysis (Table 2) of the statistical indicators of text recognition results by the OCR model indicates overall effectiveness and challenges associated with processing historical documents. The arithmetic mean of the number of recognized characters is 107.33, with an average complexity rating of text recognition at 2.93, suggesting a moderate level of document complexity. The average percentage of correctly recognized characters is quite high at 71.7%, indicating a reasonably high accuracy of the OCR model. However, significant variance in the number of recognized characters (1908.14) and the average percentage of correctly recognized characters (576.72), as well as the standard deviation for these indicators (43.68 and 24.01, respectively),

underscore the variability in recognition quality among different documents. The correlation value of -0.76 indicates a strong negative relationship between the average complexity rating of texts and the average percentage of correctly recognized characters, demonstrating that as the text complexity increases, the recognition efficiency decreases.

Table 2.
Summary of Expert Evaluation Results

	Characters recognised	SORT	SVPRS
Arithmetic Mean	107,33	2,93	71,7
Variance	1908,14	0,45	576,72
Standard Deviation	43,68	0,67	24,01
Correlation between SORT and SVPRS		-0,76	

The detailed analysis of the data [24] revealed variability in the accuracy of text recognition by the OCR model, which correlates with the complexity rating of the text, ranging from 1.0 to 4.6. Higher SVPRS values, reaching up to 100%, are observed in texts with lower complexity ratings, indicating the high efficiency of the model in recognizing less complex documents. However, a significant decrease in recognition accuracy to 1.74% and below is observed in texts with the highest complexity ratings (4.4 and above), highlighting the limitations of the current OCR model when working with highly complex historical materials. Documents numbered 69-75[24], written in Ukrainian using the Latin alphabet, demonstrated significantly lower recognition accuracy compared to others, as reflected in the average percentage of correctly recognized characters (SVPRS) ranging from 1.74% to 6.99%. Meanwhile, the complexity rating of these texts was the highest among all analyzed documents (4.4 and above), indicating significant challenges that the OCR model faces when working with texts written in the Latin alphabet but in the Ukrainian language. These results underscore the particular challenges associated with recognizing texts in languages using non-standard or less common alphabets and indicate a critical need for the development of specialized approaches and algorithms to enhance OCR accuracy in such conditions.

In conclusion, the analysis of the expert evaluation results and the statistical overview of the collected data regarding the efficiency of the OCR model confirm its significant potential utility in the study of historical texts. However, the high level of negative correlation between text complexity and recognition accuracy emphasizes the importance of further refinement and adaptation of OCR technologies to optimize working with complex historical materials. Special attention to texts written in non-standard alphabets, such as Ukrainian using the Latin alphabet, underscores the necessity for the development of specialized approaches to overcome these unique challenges, paving the way for improving accessibility and preservation of valuable historical heritage.

5. Conclusions

Within this study, an analysis of the effectiveness of the Attention-Gated-CNN-BGRU model for HTR from historical documents stored in the State Archives of Khmelnytskyi Oblast was conducted, primarily focusing on documents written in Ukrainian and Russian languages. The research concentrated on digitized descriptions of ancient deeds from 1861 to 1913. The utilized model demonstrated an SVPRS of 71.7%, indicating significant efficiency in the context of the complexity of the analyzed documents.

Comparing our results with similar solutions [1, 15] in this field, it can be noted that the incorporation of attention mechanism in the Attention-Gated-CNN-BGRU model provided significant advantages in recognition accuracy, especially for texts with high complexity and crossed-out words. This marked a significant advancement compared to traditional CRNN models, which often show decreased efficiency under similar conditions. The identified strong negative correlation (-0.76) between text complexity and recognition accuracy underscores the

necessity for further development and optimization of models for handling highly complex historical manuscripts.

One of the main directions for future research is the development and adaptation of the model for effective recognition of texts written in Ukrainian using the Latin alphabet. This aspect is crucial for the preservation and study of Ukraine's cultural heritage, as a considerable number of historical documents of significant importance are written in this alphabet. The results of our study indicate the potential feasibility of effectively adapting existing technologies for recognizing such texts, but also emphasize the need for further developments in this area.

In conclusion, our research not only confirmed the high effectiveness of the Attention-Gated-CNN-BGRU model in recognizing handwritten text from historical documents but also identified promising directions for future work. Specifically, the development of models for recognizing texts written in Ukrainian using the Latin alphabet could be a key step in preserving and making important historical resources accessible, enriching our understanding of the past and promoting cultural exchange.

6. References

[1] Ukrainian generic handwriting. (б. д.). READ-COOP. https://readcoop.eu/model/ukrainian-generic-handwriting/

[2] Transkribus. (d. b.). Transkribus. https://beta.transkribus.org/sites/tryckfrihet/browse

[3] Dubrovina, L., & Kovalchuk, G. (2016). Development of electronic information resources of manuscript and book heritage at the V. I. Vernadsky National Library of Ukraine. Library Herald, (1), 3-11.

[4] Kruchinina, T. H., & Cherniatynska, Y. H. (2015). Creation of archival collections of websites within the framework of initiative documentation at the Central State Electronic Archives of Ukraine. Archives of Ukraine, (5-6), 61-67.

[5] Maistrenko, A. A., & Romanovsky, R. V. (2018). Digitization of archival documents in foreign countries: information-analytical overview. Archives of Ukraine, (1), 64-87.

[6] Sviatets, Y. A. (2019). Digital sources in research practices of historians: theoretical and methodological aspect. Universum Historiae et Archeologiae= The Universe of History and Archeology. Dnipro, 2(27), 47-68.

[7] Philipova, L. (2018). Digital archives in modern society: terminological and substantive aspects. Library Science. Document Science. Informatics, (2), 6-11.

[8] O. Pisnyi et al., "AR Intelligent Real-time Method for Cultural Heritage Object Recognition," 2023 IEEE 5th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2023, pp. 62-66, doi: 10.1109/AICT61584.2023.10452426.

[9] Lipianina-Honcharenko, K., Savchyshyn, R., Sachenko, A., Chaban, A., Kit, I., & Lendiuk, T. (2022). Concept of the Intelligent Guide with AR Support. International Journal of Computing, 271–277. https://doi.org/10.47839/ijc.21.2.2596

[10] Schauer, S., Sieck, J., Lipianina-Honcharenko, K., Sachenko, A., & Kit, I. (2023). Use of Digital Auralised 3D Models of Cultural Heritage Sites for Long-term Preservation. У 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). IEEE. https://doi.org/10.1109/idaacs58523.2023.10348637

[11] Komar, M., Dorosh, V., Hladiy, G., & Sachenko, A. (2018). Deep Neural Network for Detection of Cyber Attacks. У 2018 IEEE First International Conference on System Analysis & Intelligent Computing (SAIC). IEEE. https://doi.org/10.1109/saic.2018.8516753

[12] Anfilets, S., Bezobrazov, S., Golovko, V., Sachenko, A., Komar, M., Dolny, R., Kasyanik, V., Bykovyy, P., Mikhno, E., & Osolinskyi, O. (2020). Deep multilayer neural network for predicting the winner of football matches. International Journal of Computing, 19(1), 70–77. https://doi.org/10.31891/1727-6209/2020/19/1-70-77

[13] Yatskiv, V., Jun, S., Yatskiv, N., Sachenko, A., & Osolinskiy, O. (2011). Multilevel method of data coding in WSN. У 2011 IEEE 6th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). IEEE. https://doi.org/10.1109/idaacs.2011.6072894
[14] Sachenko, A., Osolinskyi, O., Bykovyy, P., Dobrowolski, M., & Kochan, V. (2020). Development of the Flexible Traffic Control System Using the LabView and ThingSpeak. У 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT). IEEE. https://doi.org/10.1109/dessert50317.2020.9125036
[15] Aradillas Jaramillo, J. C., Murillo-Fuentes, J. J., & M. Olmos, P. (2018). Boosting Handwriting Text Recognition in Small Databases with Transfer Learning. У 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE. https://doi.org/10.1109/icfhr-2018.2018.00081
[16] Balci, Batuhan, Dan Saadati, and Dan Shiferaw. "Handwritten text recognition using deep learning." CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring (2017): 752-759.
[17] Yintong Wang1,2, Wenjie Xiao2 and Shuo Li2 Journal of Physics: Conference Series, Volume 1848, 2021 4th International Conference on Advanced Algorithms and Control Engineering (ICAACE 2021) January 29-31, 2021, Sanya, China
[18] Nisa, H., Thom, J. A., Ciesielski, V., & Tennakoon, R. (2019). A deep learning approach to handwritten text recognition in the presence of struck-out text. У 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE. https://doi.org/10.1109/ivcnz48456.2019.8961024
[19] Yang, J., Ren, P., & Kong, X. (2019). Handwriting Text Recognition Based on Faster R-CNN. У 2019 Chinese Automation Congress (CAC). IEEE. https://doi.org/10.1109/cac48633.2019.8997382
[20] Ingle, R. R., Fujii, Y., Deselaers, T., Baccash, J., & Popat, A. C. (2019). A Scalable Handwritten Text Recognition System. У 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE. https://doi.org/10.1109/icdar.2019.00013
[21] Sanchez, J. A., Romero, V., Toselli, A. H., & Vidal, E. (2016). ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset. У 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE. https://doi.org/10.1109/icfhr.2016.0120
[22] Retsinas, G., Sfikas, G., Gatos, B., & Nikou, C. (2022, May). Best practices for a handwritten text recognition system. In International Workshop on Document Analysis Systems (pp. 247-259). Cham: Springer International Publishing.
[23] Mishra, P., Pai, P., Patel, M., & Sonkusare, R. (2020). Extraction of Information from Handwriting using Optical Character recognition and Neural Networks. У 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE. https://doi.org/10.1109/iceca49313.2020.9297418
[24] GitHub - kniosu/handwriting-text-recognition: Handwriting text recognition. (б. д.). GitHub. https://github.com/kniosu/handwriting-text-recognition
[25] I. Paliy, V. Turchenko, V. Koval, A. Sachenko and G. Markowsky, "Approach to recognition of license plate numbers using neural networks," 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, Hungary, 2004, pp. 2965-2970 vol.4, doi: 10.1109/IJCNN.2004.1381137
[26] V. Lytvyn, V. Vysotska, P. Pukach, Z. Nytrebych, I. Demkiv, A. Senyk, O. Malanchuk. Analysis of the developed quantitative method for automatic attribution of scientific and technical text content written in Ukrainian - Eastern-European Journal of Enterprise Technologies. - 2018. - № 6(2). - P. 19-31.

Vol-3702
urn:nbn:de:0074-3702-6

CMIS 2024
Computer Modeling and Intelligent Systems 2024

Proceedings of The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024)

Zaporizhzhia, Ukraine, May 3, 2024.

Edited by

Sergey Subbotin *

* National University "Zaporizhzhia Polytechnic", Faculty of Computer Science and Technologies, Zaporizhzhia, Ukraine

Table of Contents

2024-05-14: submitted by Sergey Subbotin, metadata incl. bibliographic data published under Creative Commons CC0
2024-06-09: published on CEUR Workshop Proceedings (CEUR-WS.org, ISSN 1613-0073) |valid HTML5|

Evaluation of the effectiveness of machine learning methods for detecting disinformation in Ukrainian text data

Khrystyna Lipianina-Honcharenko[1], Mariana Soia[1], Khrystyna Yurkiv[1], Andrii Ivasechko[1].
[1] *West Ukrainian National University, Lvivska str., 11, Ternopil, 46000, Ukraine*

Abstract
In today's world, where information spreads with unprecedented speed, disinformation poses a serious challenge to public trust and information security. The full-scale invasion of Ukraine by Russia in 2022 activated the use of disinformation as a tool of hybrid warfare, highlighting the need for effective methods of identification and control. This article focuses on evaluating the effectiveness of various machine learning methods for detecting disinformation in Ukrainian text data, using a dataset that includes news headlines collected during the conflict. The study encompasses the analysis of logistic regression, support vector machines (SVM), random forest, gradient boosting, KNN, decision trees, XGBoost, and AdaBoost. Model evaluation was performed using standard metrics: precision, recall, F1-score, overall accuracy, and confusion matrix. The results indicate significant potential for using machine learning in the fight against disinformation, particularly the random forest model demonstrated the highest effectiveness. The study emphasizes the importance of adapting and optimizing classifiers for the specific task of disinformation analysis, paving the way for further research in this field.

Keywords
Disinformation, machine learning, classification, text data.

1. Introduction

In the modern information space, a vast amount of data is generated daily, a significant portion of which is news content. In the context of increasing globalization and accessibility of information technologies, information spreads rapidly through the network, making it a powerful tool for influencing public opinion. However, this also paves the way for the mass dissemination of disinformation, which can have significant consequences for society, politics, and international relations. Navigating this flow of information and distinguishing reliable data from false has become an increasingly important task.

The war that began with Russia's full-scale invasion of Ukraine in February 2022 is a striking example of the use of disinformation as a weapon in hybrid warfare. This has created a need for the development of effective tools for analyzing and classifying informational content, with the goal of identifying and counteracting disinformation.

In this context, machine learning and natural language processing methods play a key role in the detection and analysis of fake news. The application of these technologies allows for the automation of the disinformation detection process, providing fast and efficient processing of large volumes of data. At the same time, the development of effective machine learning models for information classification requires a deep understanding of data specifics, preprocessing methods, and model optimization.

This article is devoted to the analysis of a dataset containing news headlines collected during the Russo-Ukrainian conflict, aimed at identifying disinformation. It considers the application of various machine learning methods, including logistic regression, support vector machines (SVM), random forest, gradient boosting, KNN, decision trees, XGBoost, and AdaBoost for the classification of text data. The concluding section is dedicated to evaluating the effectiveness of these models using standard evaluation metrics, including precision, recall, F1-score, overall

Proceedings Acronym: Proceedings Name, Month XX–XX, YYYY, City, Country
xrustya.com@gmail.com (K. Lipianina-Honcharenko); mariankasoa@gmail.com (M. Soia); kh.yurkiv@wunu.edu.ua (K. Yurkiv); Andrewivasechko@gmail.com (A. Ivasechko).
0000-0002-2441-6292 (K. Lipianina-Honcharenko); 0009-0001-3719-6460 (M.Soia); 0009-0007-4917-3251 (K. Yurkiv); 0009-0002-8623-7828 (A. Ivasechko).

accuracy, and the confusion matrix, which allows determining the most effective methods for combating disinformation in the context of information warfare.

2. Related Work

In contemporary research in the field of information security and media content analysis, the importance of detecting and analyzing disinformation, especially anti-vaccine content on social media platforms such as Twitter, has gained particular relevance. In [1], language-neutral models are developed for detecting such content on a large scale, using multifaceted representations of messages in networks. Meanwhile, [2] focuses on the challenges associated with pre-training graph neural networks for context-oriented detection of fake news, pointing out strategic and resource constraints. In [3], a comparative analysis of supervised and unsupervised machine learning algorithms for detecting fake news is conducted, demonstrating their performance, efficiency, and robustness.

Research covering the analysis of disinformation and public opinion on the Russo-Ukrainian War employs a variety of methodologies, including sentiment analysis, creation and analysis of datasets, and studies of the impact of war on language choice. One study models and clusters sentiment trends of different countries regarding the war [4], while another offers a detailed dataset of tweets related to the crisis [5]. The discourse on Twitter about the war is also analyzed, with a particular focus on language and the geographical origin of tweets [6]. Another study focuses on the challenges of labeling sensitive content and its psychological impact on annotators [7]. It is also examined how the war affects the language choice of Ukrainians on Twitter, analyzing changes in language preferences over time [8]. A separate study provides insights into the activity on subreddits related to the conflict, analyzing post volumes, comments, and the level of engagement [9]. Research [10] demonstrates that the application of the BERT model for fake news detection achieves an accuracy of 79.88% and an area under the ROC curve of 0.87, highlighting its potential in combating disinformation on social networks.

As confirmed by the aforementioned analysis, research in the field of disinformation detection, particularly fake news, is a significant direction in the context of information security and media content analysis. The need for the development and application of advanced technological solutions for effective fake news detection is particularly compelling. However, there is a noticeable lack of research focused on the analysis of disinformation in the Ukrainian language, which poses a challenge to the scientific community to expand the linguistic spectrum of research in this field. Considering this, the aim of this study is to develop intelligent methods for detecting disinformation, specifically fake news, with an emphasis on Ukrainian-language content. This will enhance the level of information security and ensure the integrity of the news space in the conditions of the modern information society.

3. Research methodology

3.1 Dataset Description

For data collection and processing, the dataset (Ukrainian language) [11] was used, which contains approximately 10,700 news headlines about the Russo-Ukrainian War, collected from February 24 to December 11, 2022, covering the period from the beginning of the full-scale invasion.

The dataset for the analysis of disinformation in the Ukrainian media space consists of two main files: "data_set_4.csv" (Table 1) and "news_data.csv" (Table 2), each containing news headlines classified as true ("True") or false ("False"). The "data_set_4.csv" file records 8,237 true and 2,498 false news items, while "news_data.csv" contains a significantly larger number of true news items — 48,006, compared to 2,024 false, reflecting a wide range of informational content collected for the study of the dissemination of disinformation during the Russo-Ukrainian War.

Both datasets (see Table 1,2) are used for the analysis of disinformation in the Ukrainian media space and include data that were collected from official and unofficial sources with the purpose of studying the spread of fake news in the context of the Russo-Ukrainian War.

Table 1
Structure of data_set_4.csv

Field	Description	Data Type
Text	News Headline	Text
Label	Label that marks the news as true ("True") or false ("False")	Boolean
Link	Link to the news source (available only in this file)	Text

Table 2
Structure of news_data.csv

Field	Description	Data Type
Text	News Headline	Text
Label	Classification of the news as true ("True") or false ("False")	Boolean

The graphs (Fig.1) of label distribution show that in both datasets, the number of true messages predominates over the false ones. For example, in the first dataset, the ratio of true messages to false is high, which indicates a focus on reliable information.

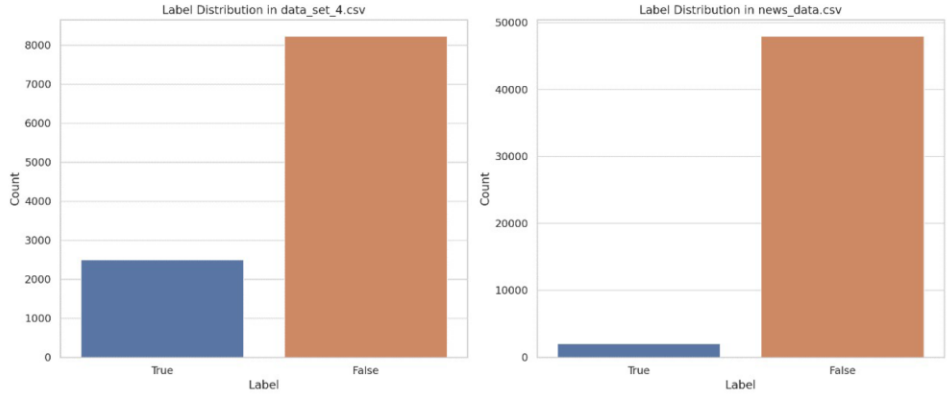


Figure 1: Label Distribution

The analysis of the most frequently used words (Fig.2) in the first dataset revealed words that appear hundreds of times. This allows identifying key topics of discussions or news, for example, the word "Ukraine" may occur most frequently, emphasizing the geographical or political focus of the collected data.

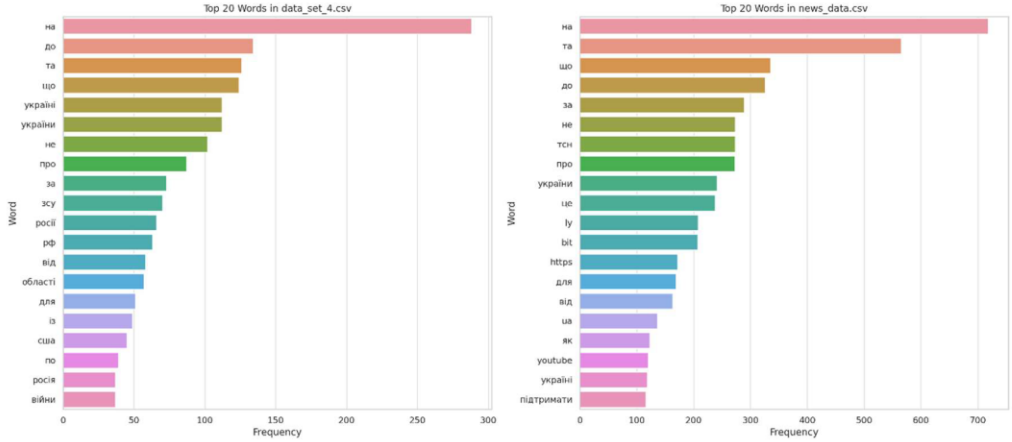


Figure 2: Top 20 Words

Most texts (Fig.3) in the first dataset have a length of 100 to 500 characters. Such information helps to understand the average volume of messages, which may indicate a prevalence of short news or overviews.

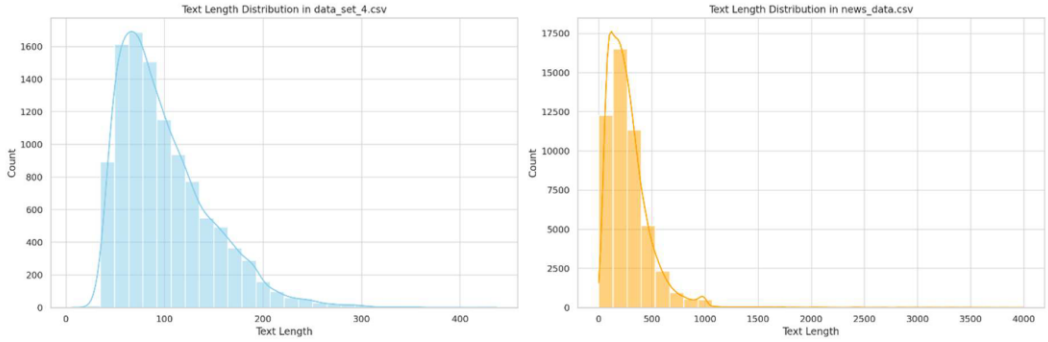


Figure 3: Distribution of Text Lengths

Analysis of the presented datasets covering headlines of news about the Russian-Ukrainian war demonstrates a significant advantage of credible information compared to false, emphasizing the focus on quality content. Considering that the dataset "news_data.csv" contains substantially more records compared to "data_set_4.csv", it is logical to use the former for training machine learning models as it provides a broader range of information and a greater number of examples for training. Meanwhile, the smaller dataset "data_set_4.csv" can serve as an excellent set for testing and evaluating the effectiveness of models on a smaller but specific data sample. This will allow assessing the model's ability to generalize learning on new, previously unknown data, emphasizing its practical value in real-world disinformation analysis conditions.

3.2 Description of used classifiers

In modern data analysis, especially in the context of detecting misinformation, the use of machine learning algorithms for classifying textual data becomes a key tool for developers and analysts. The diversity of classification methods [12], such as logistic regression, support vector machines (SVM), random forest, gradient boosting, k-nearest neighbors (KNN), decision tree, XGBoost, and AdaBoost, provides a wide range of approaches for data analysis and classification. Each of these algorithms has its unique advantages and limitations, making them more or less suitable for specific types of data and analytical tasks. The main goal is to select the optimal classifier that best fits the specificity of the task and the data we are working with.

Logistic regression [13] is a classification method used to predict the probability of two possible outcomes based on one or more independent variables. It transforms the linear combination of input data into probability using the logistic function. The advantages of logistic regression include simplicity in interpreting results, but it may be limited when analyzing complex relationships between variables and requires an assumption of linearity in relationships. Mathematically, logistic regression models the probability $P(Y=1)$ as a function of X, where Y is the dependent variable, and X is the set of independent variables. The probability is described by the equation:

$$P(Y = 1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\ldots+\beta_k X_k)}}, \quad (1)$$

where e is the base of the natural logarithm, β_0 is the constant term (intercept), and $\beta_1,\ldots, \beta_k$ are the coefficients of the independent variables. This equation allows us to estimate the probability that an observation belongs to class 1, depending on the values of the independent variables.

Support Vector Machine (SVM) algorithm [14] finds a hyperplane in a multidimensional space that best separates different data classes, maximizing the distance between the closest data points (support vectors) of different classes. The advantages of SVM include high accuracy in classification tasks, especially on relatively small datasets, and flexibility through the use of various kernel functions. However, its drawbacks include high computational resource

requirements for large datasets and complexity in interpreting the model. Mathematically, SVM seeks to solve the optimization problem: minimize $\frac{1}{2} || w ||^2$ subject to the constraints that

$$y_i (w \cdot x_i + b) \geq 1 \text{ to all } I, \quad (2)$$

where w is the weight vector of the hyperplane, b is the bias, x_i are the feature vectors, and y_i are the class labels.

The Random Forest algorithm [15] creates an ensemble of decision trees, training each tree on randomly selected subsets of the training dataset and features, which ensures high accuracy and model universality. The advantages of Random Forest include its ability to efficiently handle large datasets with high feature dimensionality and its lower tendency to overfit compared to individual decision trees. However, its drawbacks include relatively high computational resource requirements and the complexity of interpreting the model due to the large number of trees. The mathematical interpretation of Random Forest is based on the principle of "wisdom of the crowd," where the final model decision is determined by voting among the trees for classification tasks or averaging the outputs for regression tasks. More precisely, for classification:

$$Y = mode\{y_1, y_2, \ldots, y_n\}, \quad (3)$$

and for regression:

$$Y = \frac{1}{n} \sum_{i=1}^{n} y_i, \quad (4)$$

where y_i is the prediction of each tree, and Y is the final prediction of the ensemble.

Gradient Boosting [16] is an ensemble machine learning method that improves predictions by sequentially training weak models, typically decision trees, to minimize a loss function. It is characterized by high prediction accuracy and flexibility in parameter tuning, but it can be prone to overfitting if not properly configured and requires more time and computational resources for training compared to other algorithms. Mathematically, Gradient Boosting performs optimization by adaptively reducing the difference between actual and predicted values using gradient descent, where the model update in the m-th iteration is defined as:

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x), \quad (5)$$

where $F_{m-1}(x)$ is the prediction at the previous step, $h_m(x)$ is the weak classifier, and α_m is the learning rate.

The k-Nearest Neighbors (KNN) algorithm [17] classifies objects based on the nearest training examples in the feature space, where "k" indicates the number of neighbors considered to determine the class of a new object. The advantages of KNN include ease of implementation and its ability to effectively work with multi-class datasets. However, it requires significant computational resources to store training data and determine neighbors in large datasets, and it is sensitive to irrelevant features and data scaling. Mathematically, the classification of an object x in the KNN algorithm is determined by the majority vote of its neighbors, where each neighbor is weighted according to the inverse distance to x, typically using Euclidean distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{m} (x_j - x_{ij})^2}, \quad (6)$$

where x is the point for classification, x_i is a point from the training dataset, and j varies from 1 to m, the number of features. The class of x is determined based on the most frequent class among the k nearest neighbors.

The Decision Tree algorithm [18] builds a predictive model in the form of a tree-like structure by partitioning the dataset into smaller subsets while simultaneously developing the associated decision tree. The advantages of decision trees include ease of interpretation, the ability to handle both numerical and categorical data, and no requirement for data normalization. However, decision trees are prone to overfitting, especially with deep trees, and can be unstable, meaning small changes in data can result in significantly different decision trees. Mathematically, decision trees use the concept of information gain or reduction in uncertainty (entropy) to select the attribute that best splits the dataset into subsets according to the target variable. The information gain for attribute A is calculated as:

$$IG(T, A) = Entropy(T) - \sum_{v \in Values(A)} \frac{|T_v|}{|T|} Entropy(T_v), \quad (7)$$

where T is the training set, $Values(A)$ is the set of all possible values of attribute A, T_v is the subset of T for which attribute A has the value v, and $Entropy(T)$ is the entropy of the training set T.

XGBoost (eXtreme Gradient Boosting) [19] is a highly efficient implementation of the gradient boosting algorithm, which optimizes both linear models and decision trees. The algorithm stands out for its high execution speed, ability to efficiently scale to large datasets, and built-in support for regularization, helping to mitigate overfitting. However, XGBoost can be challenging to tune due to the large number of hyperparameters and requires more time for training compared to simpler models. Mathematically, XGBoost minimizes losses using gradient descent, where the objective function includes both the loss function L and a penalty for model complexity Ω, adding regularization:

$$Obj = \sum_i L(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad (8)$$

where y_i are the true labels, $\hat{y}_i$ are the predicted labels, f_k are the functions representing individual trees, and $\Omega(f_k)$ includes terms for regularization, such as the number of leaves and the sum of squares of node weights, thereby reducing the risk of overfitting.

AdaBoost (Adaptive Boosting) [20] is a machine learning algorithm that combines multiple weak classifiers to create a strong classifier, using an iterative approach to correct the errors of previous classifiers by assigning greater weight to observations that are harder to classify. The advantage of AdaBoost is its ability to improve prediction accuracy, ease of implementation, and automatic correction of underperforming classifiers. However, it can be prone to overfitting in the presence of outliers or highly noisy data and requires careful tuning of the number of iterations. Mathematically, AdaBoost adapts the weights of training observations, w_i, by increasing the weights of incorrectly classified observations. At each iteration step t, a classifier h_t is selected to minimize the weighted sum of errors. The final classifier is determined as a weighted sum of these classifiers.

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)), \quad (9)$$

where α_t is the weight assigned to classifier h_t, which depends on its accuracy.

Conclusions drawn from the description of the used classifiers underscore the importance of adapting the model to the specificity of the dataset and the analytical task. The effectiveness of each method depends on the size and quality of the data, the complexity of relationships in the dataset, as well as specific requirements for accuracy and interpretation of results. In the context of disinformation analysis, the choice between the simplicity of interpreting logistic regression and the high accuracy but complexity of tuning XGBoost or SVM may determine the success or failure in identifying fake news. Thus, careful selection and tuning of classifiers are critically important for developing effective tools to combat disinformation in the context of the Russian-Ukrainian war.

3.3 Evaluation metrics

For evaluating the effectiveness of models in classification tasks, key metrics such as precision, recall, F1-score, accuracy, and confusion matrix are utilized [21]. These metrics allow for a deeper analysis of the model's performance, identifying potential weaknesses, and optimizing the model for better results.

Precision is defined as the ratio of the number of true positive results to the total number of results classified as positive by the model. Mathematically, it is represented as:

$$P = \frac{TP}{TP+FP}, \quad (10)$$

where TP — true positives, and FP — false positives.

Recall measures the model's ability to identify all actual positive cases in the dataset. It is defined as the ratio of the number of correctly identified positive results to the sum of correctly identified positive results and instances that are actually positive but were missed by the model. The formula for calculation is:

$$R = \frac{TP}{TP+FN}, \quad (11)$$

where FN — false negatives.

The F1 Score is the harmonic mean between precision and recall, providing a balance between these two metrics. This is particularly useful in situations where class imbalances may cause biases in one metric over the other. F1 is defined as:

$$F1 = 2 \cdot \frac{P \cdot R}{P+R}. \quad (12)$$

Accuracy measures the percentage of cases correctly classified by the model and is defined by the formula:

$$A = \frac{TP+TN}{TP+TN+FP+F}, \quad (13)$$

where TN — true negatives.

The Confusion Matrix provides a visualization of classification results by representing the counts of TP, TN, FP, and FN in the form of a matrix. This allows us not only to determine the accuracy of the model but also to understand the types of errors made by the model.

3.4 Model training

The training procedure (Figure 4) on the training dataset is a fundamental step in the development of effective machine learning algorithms for text data classification. In this context, the use of datasets "news_data.csv" and "data_set_4.csv" for training and testing models, respectively, provides a valuable foundation for misinformation analysis. The initialization of the procedure begins with the import and preprocessing of data, including the removal of records without textual content, ensuring data cleanliness for subsequent processing stages.

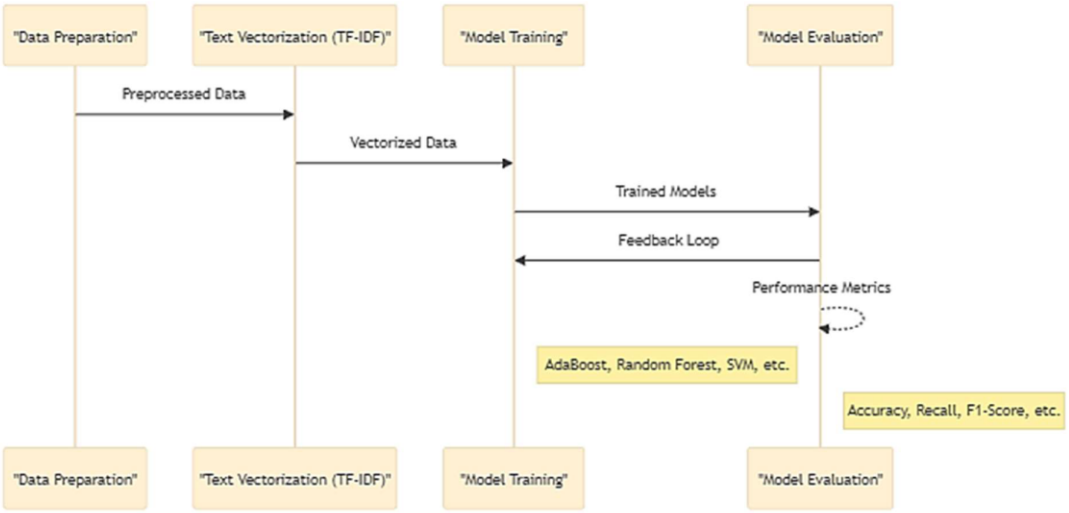


Figure 4: Model Training Process

Text vectorization using TF-IDF (Term Frequency-Inverse Document Frequency) is a key step in data preparation, as it transforms textual data into a numerical format, making them suitable for processing by machine learning models. This method considers not only the frequency of words in the text but also their uniqueness through the inverse frequency of documents, allowing the model to better identify important features in the text.

After preparing the vectorized training and testing data, the next stage involves training different models on the training dataset. This process requires the application of machine learning algorithms to the training dataset to form a model capable of effectively classifying text based on learned features. Each model adjusts its parameters to minimize errors on the training set while simultaneously ensuring the ability to generalize to new data.

Evaluation of the effectiveness of each trained model on the test dataset is crucial for determining its suitability and efficiency in classification tasks. The use of evaluation metrics such as accuracy, recall, F1-score, and overall accuracy allows for deep analysis and comparison of model performance. The evaluation results can be visualized for better understanding of model effectiveness, and the best-performing model can be selected for further use or saved for future analysis.

Thus, the model training procedure on the training dataset using pre-processed and vectorized text data is fundamental for developing reliable machine learning tools capable of effectively classifying and analyzing text for misinformation.

4. Research results

Further, we will discuss the implementation and evaluation of the effectiveness of various machine learning models in the task of classifying misinformation data in the Ukrainian media space, contextualized in the context of Russia's full-scale intervention. By analyzing a wide range of algorithms, we aim to identify the most effective methods for accurate detection and differentiation of misinformation incidents. This analysis is based on a comprehensive comparison of overall classification accuracy as well as specific metrics such as precision, recall, and F1-score for each class. The implementation was done in the Python programming language utilizing libraries for text analysis.

The logistic regression model showed (Figure 5) an overall classification accuracy of 86.4% on the test sample of 10,735 examples, demonstrating high effectiveness in identifying true values with an F1-score of 0.92 for the "True" class. However, the model was less accurate in identifying the "False" class, with a prediction accuracy of 0.96 and a low recall score of 0.44, indicating a significant number of false negatives, as reflected in the confusion matrix with 1,407 misclassified examples.

Model: logistic_regression

	precision	recall	f1-score	support
False	0.96	0.44	0.60	2498
True	0.85	0.99	0.92	8237
accuracy			0.86	10735
macro avg	0.90	0.72	0.76	10735
weighted avg	0.88	0.86	0.84	10735

Accuracy: 0.864182580344667

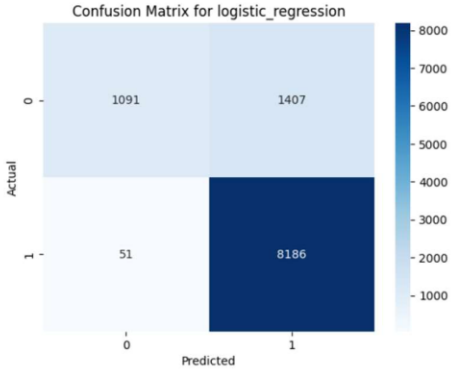


Figure 5: LogisticRegression Results

The SVM model demonstrated (Figure 6) high overall classification accuracy of 93.6% for the test sample, indicating its effectiveness in distinguishing between the "True" and "False" classes. Specific accuracy and recall metrics for the "False" class were 0.97 and 0.75, respectively, demonstrating the model's ability to identify negative cases well, albeit with some errors. At the same time, high metrics for the "True" class with precision of 0.93 and recall of 0.99 indicate minimal false negative classifications, confirmed by a low number of errors in the confusion matrix (618 for "False" and 68 for "True").

Model: svm

	precision	recall	f1-score	support
False	0.97	0.75	0.85	2498
True	0.93	0.99	0.96	8237
accuracy			0.94	10735
macro avg	0.95	0.87	0.90	10735
weighted avg	0.94	0.94	0.93	10735

Accuracy: 0.9360968793665579

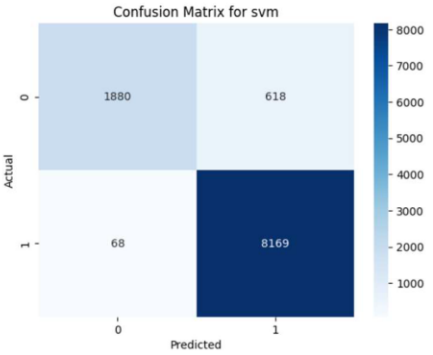


Figure 6: SVM Results

The Random Forest model demonstrated (Figure 7) high accuracy in classification with an overall accuracy of 95.3% on the test dataset, indicating the model's high effectiveness in recognizing both classes. For the "False" class, the model showed high accuracy (0.98) and a relatively high recall of 0.81, indicating the model's ability to effectively identify negative cases with a moderate number of errors. Conversely, extremely high accuracy (0.95) and almost perfect

recall (1.00) for the "True" class highlight the minimal number of false negative results, corroborated by the low number of errors in the confusion matrix.

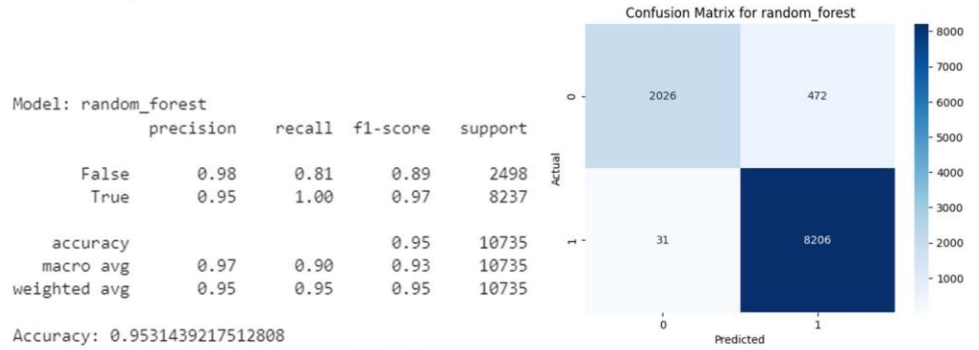


Figure 7: RandomForestClassifier Results

The Gradient Boosting model achieved (Figure 8) an overall classification accuracy of 87.3% on the test dataset, indicating the model's good ability to distinguish between the "True" and "False" classes. The model's precision for the "False" class was high (0.94), but the recall was only 0.48, indicating a significant number of type II errors, where negative cases are often misclassified as positive. Conversely, for the "True" class, the model showed impressive classification ability with a precision of 0.86 and a recall of 0.99, demonstrating its high effectiveness in identifying true positive cases with minimal errors.

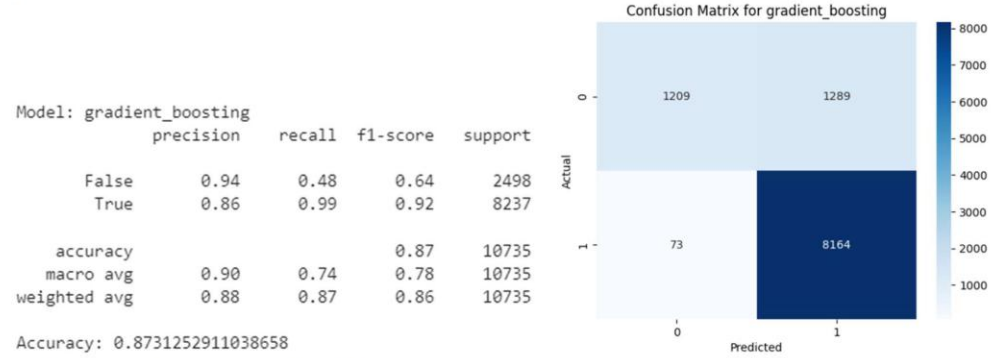


Figure 8: Gradient Boosting Results

The KNN (K-Nearest Neighbors) model demonstrated (Figure 9) an overall classification accuracy of 87.6% on the test dataset, highlighting its ability to effectively classify data. Although the model showed high precision (0.91) for the "False" class, the recall was only 0.51, indicating difficulties in identifying all negative cases. Conversely, for the "True" class, the model exhibited excellent precision (0.87) and a high recall (0.99), indicating its ability to identify positive cases with high confidence, albeit with some false positive errors, as seen in the confusion matrix.

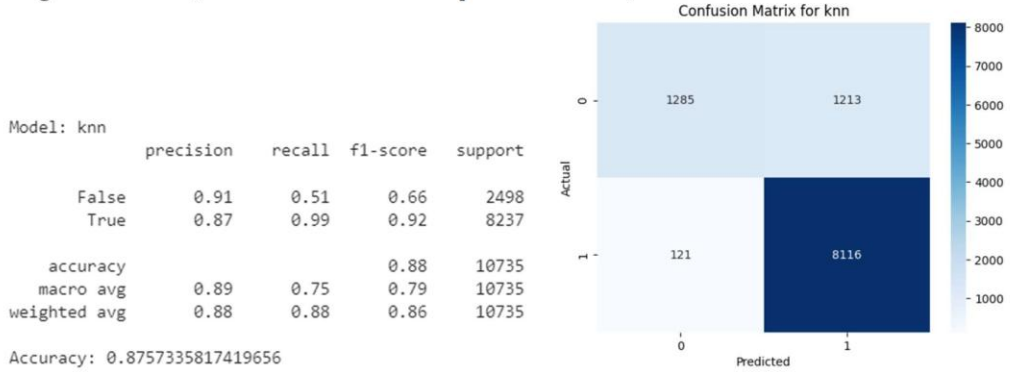


Figure 9: KNN Results

The Decision Tree model achieved (Figure 10) a high level of classification accuracy, 91.4%, on the test dataset, confirming its effectiveness in classifying data into "True" and "False" classes. For the "False" class, the model showed relatively high precision (0.81) and recall (0.83), indicating a balanced ability to identify negative cases with a relatively small number of errors. Meanwhile, for the "True" class, the model provided impressive precision (0.95) and recall (0.94), demonstrating its strong capabilities in identifying positive cases with minimal false negatives, as reflected in the confusion matrix.

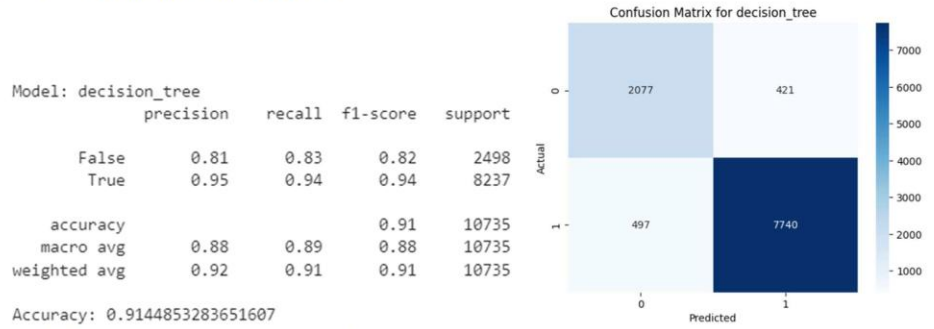


Figure 10: DecisionTreeClassifier Results

The XGBoost model demonstrated (Figure 11) an overall accuracy of 89.2% on the test dataset, indicating its ability to effectively classify the given dataset. The precision and recall metrics for the "False" class were 0.89 and 0.61, respectively, indicating the model's higher ability to correctly identify negative cases, albeit with some errors. Meanwhile, for the "True" class, the model showed high precision and recall (both 0.89 and 0.98), demonstrating excellent ability to accurately identify positive cases with minimal errors, as reflected in its high F1-score.

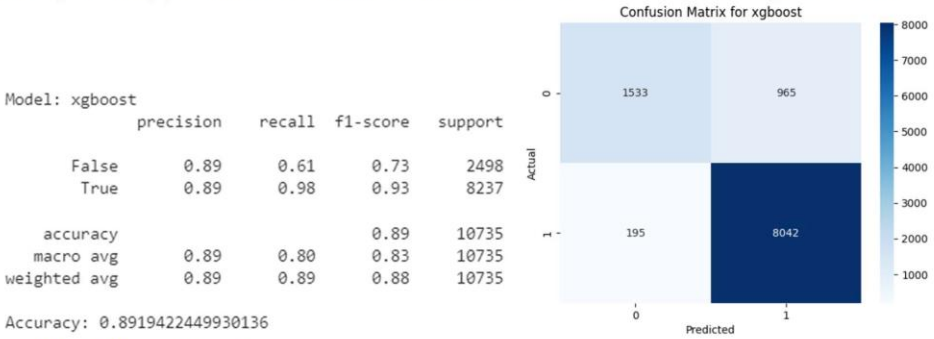


Figure 11: XGBoost Results

The AdaBoost model exhibited (Figure 12) an overall classification accuracy of 86.9% on the test dataset, indicating its effectiveness in recognizing data, albeit with some limitations. For the "False" class, the model achieved a precision of 0.88 with a recall of 0.50, indicating a relatively low ability to identify all negative cases. On the other hand, high precision (0.87) and recall (0.98) for the "True" class underscore the model's strong ability to detect positive cases, albeit with few errors, as reflected in the confusion matrix.

Model: adaboost

	precision	recall	f1-score	support
False	0.88	0.50	0.64	2498
True	0.87	0.98	0.92	8237
accuracy			0.87	10735
macro avg	0.88	0.74	0.78	10735
weighted avg	0.87	0.87	0.86	10735

Accuracy: 0.8692128551467163

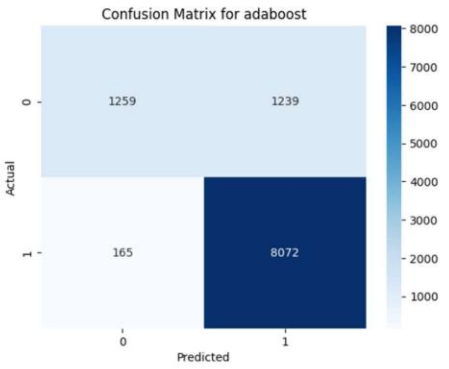


Figure 12: AdaBoostClassifier Results

Therefore, analyzing the results of applying various machine learning models to classify disinformation data in the Ukrainian media space after the onset of Russia's full-scale invasion, it can be noted that the Random Forest model proved to be the most effective with an accuracy of 95.3%, highlighting its ability to accurately detect and differentiate disinformation incidents. At the same time, models such as AdaBoost and logistic regression showed lower overall accuracy, which may indicate their limitations in identifying subtle cases of disinformation or more subjective aspects of information operations. This underscores the importance of choosing the appropriate model for the task of analyzing disinformation, where Random Forest may be more suitable for deep understanding and detecting complex patterns of disinformation in the context of information warfare.

5. Conclusion

The analysis of the effectiveness of various machine learning models applied to the task of classifying news headlines for misinformation in the Ukrainian media space demonstrates significant variations in accuracy, recall, and F1-score among the models. The considered algorithms, including logistic regression, SVM, random forest, gradient boosting, KNN, decision tree, XGBoost, and AdaBoost, showed different levels of performance in addressing the task.

The random forest model emerged as the most effective, achieving an overall accuracy of 95.3%, indicating its high capability to recognize and distinguish true and false messages. This is supported by high precision scores for both the "False" class (0.98) and the "True" class (0.95), as well as significant recall scores for both classes (0.81 for "False" and 1.00 for "True"), demonstrating its effectiveness in minimizing both false positives and false negatives.

These results underscore the importance of choosing the appropriate model for a specific misinformation analysis task. The model choice not only affects the overall classification accuracy but also the model's ability to minimize false positives or false negatives, which is crucial for developing effective tools to combat misinformation. Particularly, the random forest model, which demonstrated the best performance, can be recommended as the optimal choice for similar tasks, providing a high level of accuracy and the ability to effectively distinguish between true and false messages.

Further scientific research in the field of identification and analysis of misinformation in the Ukrainian media space requires deeper development and improvement of machine learning algorithms, with a particular focus on enhancing their ability to recognize subtle and complex forms of misinformation. The results of our study show that the random forest model, with an accuracy of 95.3%, proved to be the most effective, but there is potential for improvement, especially in accurately distinguishing between true and false messages. In the future, researchers may focus on developing hybrid models that combine the advantages of multiple algorithms, including deep learning and neural networks, to ensure greater adaptability and accuracy in different informational contexts. Additionally, an important direction will be the development of methods that allow models to better understand the semantic context and emotional tone of texts, which can significantly improve their ability to identify hidden

misinformation. Implementing such approaches will require not only technological innovations but also a deeper understanding of linguistic nuances and cultural-historical contexts on which misinformation is based.

References

[1] Ashford, J.R. (2024). Detecting Anti-vaccine Content on Twitter using Multiple Message-Based Network Representations. arXiv preprint arXiv:2402.18335.

[2] Donabauer, G., & Kruschwitz, U. (2024). Challenges in Pre-Training Graph Neural Networks for Context-Based Fake News Detection: An Evaluation of Current Strategies and Resource Limitations. arXiv preprint arXiv:2402.18179.

[3] Kaur, S., & Ranjan, S. (2024). Comparative Analysis of Supervised and Unsupervised Machine Learning Algorithms for Fake News Detection: Performance, Efficiency, and Robustness. ResearchGate.

[4] Vahdat-Nejad, H., Akbari, M. G., Salmani, F., Azizi, F., & Nili-Sani, H. R. (2023). Russia-Ukraine war: Modeling and Clustering the Sentiments Trends of Various Countries. arXiv preprint arXiv:2301.00604.

[5] Haq, E. U., Tyson, G., Lee, L. H., Braud, T., & Hui, P. (2022). Twitter dataset for 2022 russo-ukrainian crisis. arXiv preprint arXiv:2203.02955.

[6] Zia, H. B., Haq, E. U., Castro, I., Hui, P., & Tyson, G. (2023). An Analysis of Twitter Discourse on the War Between Russia and Ukraine. arXiv preprint arXiv:2306.11390.

[7] Daria, S. (2023). When a Language Question Is at Stake. A Revisited Approach to Label Sensitive Content. arXiv preprint arXiv:2311.10514.

[8] Racek, D., Davidson, B. I., Thurner, P. W., & Kauermann, G. (2023). The Politics of Language Choice: How the Russian-Ukrainian War Influences Ukrainians' Language Use on Twitter. arXiv preprint arXiv:2305.02770.

[9] Zhu, Y., Haq, E. U., Lee, L. H., Tyson, G., & Hui, P. (2022). A reddit dataset for the russo-ukrainian conflict in 2022. arXiv preprint arXiv:2206.05107.

[10] Padalko, H., Chomko, V., & Chumachenko, D. (2023). Misinformation Detection in Political News using BERT Model. Proceedings of the 3rd International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2023) Waterloo, Canada, November 20-22, 2023. 117-127

[11] Ukrainian news. (n.d.-a). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/datasets/zepopo/ukrainian-fake-and-true-news/data?select=news_data.csv

[12] Lipyanina, H., Maksymovych, V., Sachenko, A., Lendyuk, T., Fomenko, A., & Kit, I. (2020, August). Assessing the investment risk of virtual IT company based on machine learning. In International Conference on Data Stream Mining and Processing (pp. 167-187). Cham: Springer International Publishing.

[13] Lipyanina, H., Sachenko, S., Lendyuk, T., Brych, V., Yatskiv, V., & Osolinskiy, O. (2021, January). Method of detecting a fictitious company on the machine learning base. In International Conference on Computer Science, Engineering and Education Applications (pp. 138-146). Cham: Springer International Publishing.

[14] Kalogridis, I. (2024). Robust and adaptive functional logistic regression. Computational Statistics & Data Analysis, 192, 107905.

[15] Çakir, M., Yilmaz, M., Oral, M. A., Kazanci, H. Ö., & Oral, O. (2023). Accuracy assessment of RFerns, NB, SVM, and kNN machine learning classifiers in aquaculture. Journal of King Saud University-Science, 35(6), 102754.

[16] Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 237, 121549.

[17] Wang, C., Xiao, W., & Liu, J. (2023). Developing an improved extreme gradient boosting model for predicting the international roughness index of rigid pavement. Construction and Building Materials, 408, 133523.

[18] Çakir, M., Yilmaz, M., Oral, M. A., Kazanci, H. Ö., & Oral, O. (2023). Accuracy assessment of RFerns, NB, SVM, and kNN machine learning classifiers in aquaculture. Journal of King Saud University-Science, 35(6), 102754.
[19] Gao, B., Zhou, Q., & Deng, Y. (2024). HIE-EDT: Hierarchical interval estimation-based evidential decision tree. Pattern Recognition, 146, 110040.
[20] Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., Dhawan, P., & Righetti, M. (2024). Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023). Environmental Modelling & Software, 105971.
[21] Xing, H. J., Liu, W. T., & Wang, X. Z. (2024). Bounded exponential loss function based AdaBoost ensemble of OCSVMs. Pattern Recognition, 148, 110191.