Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту

(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)
(рівень вищої освіти)
Розробка системи викриття підробок
з використанням штучних нейронних мереж
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-2
Голдобин А. О
(прізвище, ініціали)

Спеціальність 122 – Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту (СШІ)
(повна назва освітньої програми)

Керівник проф. Терзіян В.Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2020 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2020 р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

студентові Голдобіну Арсенію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи викриття підробок з використанням штучних нейронних мереж

затверджена наказом по університету від 30 березня 2020 р. № 480Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 травня 2020 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження конволюційної нейронної мережі, існуючі рішення, дані з онлайн чемпіонату по викриттю DeepFake

4. Перелік питань, що потрібно опрацювати в роботі Аналіз проблеми DeepFake, аналіз алгоритмів створення підробок, аналіз алгоритмів викриття підробок, дослідження конволюційної нейронної мережі, розробка метода викриття підробок з використання конволюційної нейронної мережі, тестування та аналіз результатів.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Рисунок 1 – Процес створення deepfake за допомогою двох пар encoder-decoder, Рисунок 2 – Штучний нейрон, Рисунок 3 – Графічне зображення нейронної мережі, Рисунок 4 – Приклад розрахування конволюційного шару, Рисунок 5 – Результат обчислення активацій фільтрів, Рисунок 6 – Схема RNN, запропонована Елманом, Рисунок 7 – Розповсюдження сигналу у RNN, Рисунок 8 – Структура нейромережі автоасоціатора, Рисунок 9 – Архітектура мережі автоасоціатора, Рисунок 10 – Структура GAN, Рисунок 11 – Класифікація методів викриття підробок, Рисунок 12 – Двоступеневий процес для викриття модифікації, Рисунок 13 – DeepFake detection метод Guera і Delp, Рисунок 14 – Пайплайн Nguyen, Рисунок 15 – Приклади відеозаписів із першої частини, Рисунок 16 – Характеристики частини даних, Рисунок 17 - Архітектура базового рішення, Рисунок 18 – Приклад знаходження обличча з відеозапису, Рисунок 19 – Приклад витягнутого обличчя, Рисунок 20 – Хід навчання моделі, Рисунок 21 – Приклад поганого батчу

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Терзіян В. Я.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на атестаційну роботу	30.03.20	виконано
2	Аналіз предметної області і постановка завдання	31.03.20-12.04.20	виконано
3	Дослідження методів навчання	13.04.20-14.04.20	виконано
4	Створення імітаційної моделі	15.04.20-20.04.20	виконано
5	Тестування і відладка імітаційної моделі	20.04.20-28.04.20	виконано
6	Обробка і оформлення результатів	29.04.20-03.04.20	виконано
7	Оформлення пояснювальної записки	04.05.20-12.05.20	виконано
8	Нормоконтроль	15.05.20	виконано
9	Попередній захист	18.05.20	виконано
10	Захист перед ЕК	19.05.20	

Дата видачі завдання 30 березня 2020 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Терзіян В.Я.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 66 с., 21 рисунок, 1 таблиця, 18 джерел.

DEEPFAKE, ГЕНЕРАТИВНІ ЗМАГАЛЬНІ МЕРЕЖІ, ГЛИБИННЕ НАВЧАННЯ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ, НЕЙРОННІ МЕРЕЖІ

Об’єктом дослідження є підроблення відеозаписів за допомогою технологій DeepFake.

Предметом дослідження є підроблені відеозаписи, а також системи її викриття.

Метою роботи є розробка системи викриття підробок за допомогою штучних нейронних мереж з використанням мови програмування Python.

Методами дослідження є аналіз існуючих систем створення та розкриття підробок, а також аналіз літератури та електронних ресурсів.

В даній роботі розглянуті методи, за допомогою яких створюються підроблені відеозаписи з модифікацією чи заміною обличчя, а також методів, що викривають такі записи.

На основі проведеного аналізу був створений власний метод викриття підроблених відеофайлів, які також мають назву DeepFake.

РЕФЕРАТ

Пояснительная записка: 66 с., 21 рис., 1 табл., 2 прил., 18 источников.

DEEPFAKE, ГЕНЕРАТИВНЫЕ СОРЕВНОВАТЕЛЬНЫ СЕТИ, ГЛУБИННЕ ОБУЧЕНИЕ, ІНТЕЛЕКТУАЛЬНИЙ АНАЛИЗ ДАННИХ, НЕЙРОННЫЕ СЕТИ

Объект исследования – поддельные с помощью технологий DeepFake видео.

Предмет исследования – поддельные видеозаписи, а також системі их обнаружения.

Цель работы – разработка системы обнаружения подделок

Методы исследования – анализ существующих систем создания и обнаружения подделок, а также анализ литературы и электронных ресурсов.

В данной работе рассматриваются методы, с помощью которых создаются поддельные видеозаписи с модификацией или заменой лица, а также методы, которые обнаруживают такие записи.

На основании проведённого анализа был создан собственный метод обнаружения поддельных видеофайлов, которые также называют DeepFake.

ABSTRACT

Explanatory note: 66 p., 21 fig., 1 tabl., 2 ann., 18 sources.

DATA MINING, DEEPFAKE, DEEP LEARNING, GENERATIVE ADVERSERIAL NETWORKS, NEURAL NETWORKS

The object of study is fake video using DeepFake technology.

The subject of the research is fake videos, as well as systems for their detection.

Purpose of work – development of a fake detection system

Research methods – analysis of existing systems for fake creation and detection, as well as analysis of literature and electronic resources.

This paper discusses the methods by which fake videos are created with face modification or replacement, as well as methods that detect such videos.

Based on the analysis, we created our own method for detecting fake video files, also called DeepFake.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional Neural Network – конволюційна (або згорткова) нейронна мережа – нейрона мережа, яка складається з особливих шарів: згорткових та інших;

GAN – Generative adversarial network – Генеративні змагальні мережі;

ReLU – Rectified Linear Unit –випрямляч – активаційна функія, яка дорівнює собі при невід’ємних значеннях та 0 у інших;

RNN – Recurrent neural network – рекурентні нейронні мережі – нейрона мережа, у составі якої є шари, які зберігають стан, який залежить від усіх попередніх екземплярів.

ВСТУП

У сучасному світі розвиток та поширення технологій значно покращило рівень якості життя людей. Завдяки впровадженню ІТ-технологій чимала кількість галузей діяльності людини покращила свої показники продуктивності.

Використання методів штучного інтелекту призвело до другої хвилі розвитку: завдяки методам комп'ютерного зору стало можливим з великою ймовірністю визначати доброякісність пухлин без виростання дорогоцінних аналізів на початкових етапах обстежень; завдяки розвитку методів обробки природної мови пошукові системи навчилися пропонувати найбільш вірогідні варіанти запитів, перш ніж користувач завершив написання; значно покращилась також якість релевантності результатів пошуку: в більшості випадків пошук необхідної інформації закінчується на перегляді результатів на першій сторінці; у галузі послуг методи колоборативної фільтрації допомагають вибрати кіно, серіал, відео чи музику, які в більшості випадків сподобаються користувачу.

Перераховувати корисне застосування технологій штучного інтелекту можна безліч часу. Проте його застосування, як і будь-якої іншої технології, не обмежено гарними намірами. Однією з суперечливих технологій, яка зараз набирає попит поміж спеціалістів і міцно займає перші стрічки технологічних новин – це DeepFake. Назва методики походить від англійського deep learning – глибинне навчання і fake – підробка та каже само за себе. Це технологія, яка використовує методи глибинного навчання для підробки відеозаписів, шляхом зміни контенту: заміни лиця однієї людину на іншу або модифікації запису в підробкою аудіо- чи відеосигналів для створення враження, що людина казала або робила речі, чого не було в реальному житті.

Проте дана технологія дозволяє створювати також розважальні ролики, наприклад, створити відео в якому дає можливість переглянути, як

буде виглядати Термінатор в однойменному фільму, якщо його роль виконував не Арнольд Шварценеггер, а Сильвестр Сталлоне. Однак не слід забувати якої соціальної, політичної та економічної шкоди може завдати відеозапис, в якому високопоставлена людина буде казати речі, давати обіцянки або навіть погрожувати, чого ніколи не відбувалось насправді.

З метою викриття таких відеозаписів в галузі машинного навчання з’явилась нова задача DeepFake Detection – побудова таких моделей, які дозволяють відрізнити створені штучним шляхом відеозаписи від справжніх.

Таким чином, метою даної роботи є дослідження різних алгоритмів, які генерують підробки, дослідження реалізацій методів DeepFake Detection, а також спроба розробити власний метод викриття штучностворенних записів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Для того, щоб мати можливість відрізняти відео-підробки від оригінальних відеозаписів потрібно розуміти, як створюються підробки. Задля цього в рамках розділу буде показано що таке DeepFake, чому актуально їх викриття, які основні методи створення DeepFake існують, а також буде сформована постановка задачі.

1.1 Що таке DeepFakes, і чому їх викриття є актуальною задачею

DeepFake – це техніка, яка замінює обличчя людини на зображенні або відео на обличчя цільової людини шляхом накладання на першу людину перетворене зображення обличчя цільової людини, таким чином, що цільова людина робить ті самі дії або каже ті самі слова, які казала чи робила людина із відео джерела. Моделі глибинного навчання такі як автоенкодери або генеративні змагальні мережі широко застосовуються для вирішення різноманітних задач комп’ютерного зору. Таким самим чином ці моделі застосовуються в DeepFake задля вивчення виразів обличчя та дій людини джерела та синтезування таких самих зображень та відео вже цільової людини. DeepFake вимагають великого обсягу даних у вигляді фото та відео цільової людини для тренування моделей, щоб отримати фотореалістичні зображення та відео. Таким чином, публічні люди, такі як знаменитості або політики, у яких цих даних достатньо у відкритому доступі онлайн, є першими жертвами для цієї технології. Отримані відеозаписи, які називають за тим самим ім’ям – deepfakes, вже були застосовані для заміни облич відомої голлівудської акторки та акторки порнографічних фільмів вперше у 2017 році. Це є погрожуючим для мирової безпеки, коли методи DeepFake можуть бути застосовані задля створення підроблених з ціллю фальсифікації відео, у яких світові лідери

кажуть підроблені промови. Більш того DeepFake можуть бути застосовані для підвищення політичних або релігійних суперечок між країнами, обману населення та впливу на результати виборів, або створення хаосу у фінансових ринках, розповсюджуючи фальшиві новини [1].

Це підтверджує необхідність створення систем визначення підробок і актуальність теми.

1.2 Принципи роботи та основні моделі

DeepFakes стали популярними завдяки якості підроблених відеозаписів, а також простоті у використанні їхніх додатків для широкого кола користувачів з різними навичками роботи на комп’ютері від професіонала до початківця. Ці додатки здебільшого розробляються на основі методів глибокого навчання. Глибоке навчання добре відоме своєю здатністю представляти складні та об'ємні дані. Одним із варіантів глибоких мереж з такою можливістю є глибокі автокодери, які широко застосовуються для зменшення розмірності та стиснення зображення.

Першою спробою створення deepfakes був FakeApp, розроблений користувачем Reddit за допомогою структури автокодер-декодер. У цьому способі автокодер витягує приховані особливості зображень обличчя, а декодер використовується для реконструкції зображень обличчя. Для обміну граней між вихідними зображеннями та цільовими зображеннями необхідні дві пари кодер-декодер, де кожна пара використовується для тренування на наборі зображень, а параметри кодера розподіляються між двома мережевими парами. Іншими словами, дві пари мають однакову мережу кодера. Ця стратегія дозволяє загальному кодеру визначити схожість між двома наборами зображень обличчя, які є досить непростими, оскільки обличчя зазвичай мають подібні риси, такі як очі, ніс, положення рота.

На рисунку 1.1 показаний процес створення deepfake, коли набір функцій обличчя A пов'язаний з декодером B для реконструкції обличчя B від вихідного обличчя A. Цей підхід застосовується в кількох роботах, таких як DeepFaceLab, DFaker, DeepFake-tf.

Популярні інструменти за допомогою котрих створюються DeepFake зображення та відеозаписи та їх особливості, а також посилання до сторінки вмістилищем представлені в таблиці 1.1.

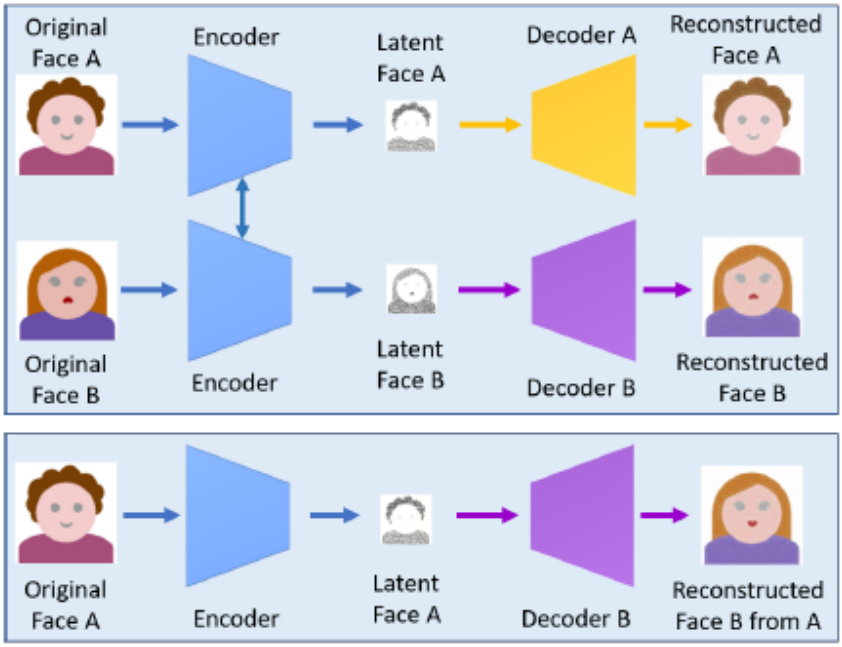

Рисунок 1.1 – Процес створення deepfake за допомогою двох пар encoder-decoder

Таблиця 1.1 – Приклади DeepFake інструментів

Інструмент	GitHub	Особливості
1	2	3
Faceswap	https://github.com/deepfakes/faceswap	Використовуються 2 пари кодера та декодера, загальні параметри кодера [2]

Продовження таблиці 1.1

1	2	3
Faceswap-GAN	https://github.com/shaoanlu/faceswap-GAN	Доповнені двома видами функцій втрат
DeepFaceLab	https://github.com/iperov/DeepFaceLab	Розширена версія FaceSwap, підтримує кілька облич
DFaker	https://github.com/dfaker/df	DSSIM – функція втрат, виконано задопомогою Keras

1.3 Постановка задачі

Метою даної роботи є розробка системи викриття підробок з використанням штучних нейронних мереж.

Для досягнення мети необхідно вирішити наступні задачі та виконати наступні дії:

- дослідити найбільш поширені інструменти створення Deepfakes;
- опрацювати існуючі методи викриття Deepfakes;
- визначити недоліки існуючих методів, якщо вони є;
- спроектувати систему викриття підробок враховуючи наявні джерела даних, недоліки та переваги існуючих методів;
- реалізувати основні компоненти системи, крім нейронних мереж;
- обрати структури використаних нейронних мереж;
- обрати гіперпараметри та метод навчання;
- навчити нейронні мережі, або використати існуючі мережі та виконати тонку настройку;
- оцінити результати за допомогою існуючих еталонних наборів даних;
- зробити висновки та запропонувати подальші поліпшення.

2 ТЕОРИТИЧНІ ДОСЛІДЖЕННЯ

Задача автоматичного викриття підробок є дуже складною, тому що по зображенню чи відеозапису, які для комп’ютера представлені у вигляді величезного набору нулів та одиниць потрібно вміти визначати, чи є це зображення чи відеозапис модифікованими саме за допомогою DeepFake моделей чи ні. Такі задачі можуть вирішувати методи машинного навчання, такі як нейронні мережі.

Як було зазначено у минулому розділі підробки є результатом роботи штучних нейронних мереж. А саме глибинних нейронних мереж з особливою архітектурою. У цьому розділі розповідається про нейроні мережі, компоненти із яких вони можуть складатися та архітектури й підходи, які використовуються в даній галузі як при генерації так само і при викритті deepfakes.

2.1 Штучні нейроні мережі

Штучні нейронні мережі або просто нейронна мережа – це математична модель, а також її програмні або апаратні реалізації, побудована в певному сенсі за образом і подобою мереж нервових клітин живого організму [3].

Штучну нейрона мережа складається штучних нейронів. Штучний нейрон є блоком, який має суматор, що складає підсилені вхідні сигнали з додатковим сигналом та подає на вихід результат активаційної функції над сумою, тобто приймає вектор вхідного сигналу, помножає його на вектор чисел, які мають назву «ваги», додає ще одно число, яке має назву «зміщення», а результат подає до деякої нелінійної функції. Ваги та зміщення є змінними параметрами нейронної мережі, що навчаються.

На рис. 2.1 x_k – елементи вектору вхідного сигналу, w_{kj} – ваги моделі та b_j – зміщення штучного нейрону.

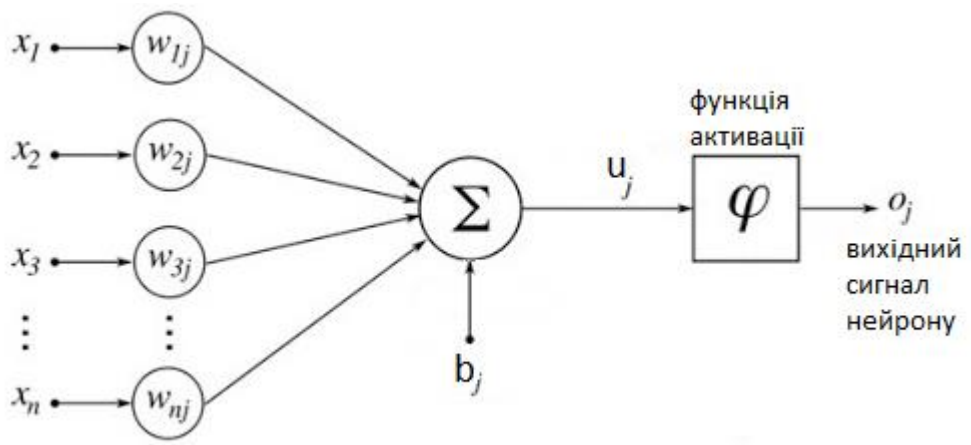

Рисунок 2.1 – Штучний нейрон

Ці блоки складаються в шари, які передають сигнал із входу нейронної мережі в її вихід, при цьому вихід нейронів першого одного є вхідним сигналом для наступного шару

На рисунку 2.2 графічно зображена нейрона мережа, де h_i – внутрішні чи приховані шари, кожна кулька в яких і є блоком – штучним нейроном, а і та о – вхідний та вихідний шари відповідно (є тими самими вхідними та вихідними сигналами мережі).

Для отримання результатів обчислення мережі, вхідний сигнал шар за шаром передається із вхідного шару до прихованих від першого до останнього, вихідним сигналом якого і буде вихідний сигнал мережи, який також називають її передбаченням. Цю процедуру обчислення називають пряме поширення. Навчають нейрону мережу, зокрема, методом зворотного поширення помилки, коли сигнал для зміни параметрів моделі розраховується у зворотному напрямку – від помилки вихідного слою до першого внутрішнього.

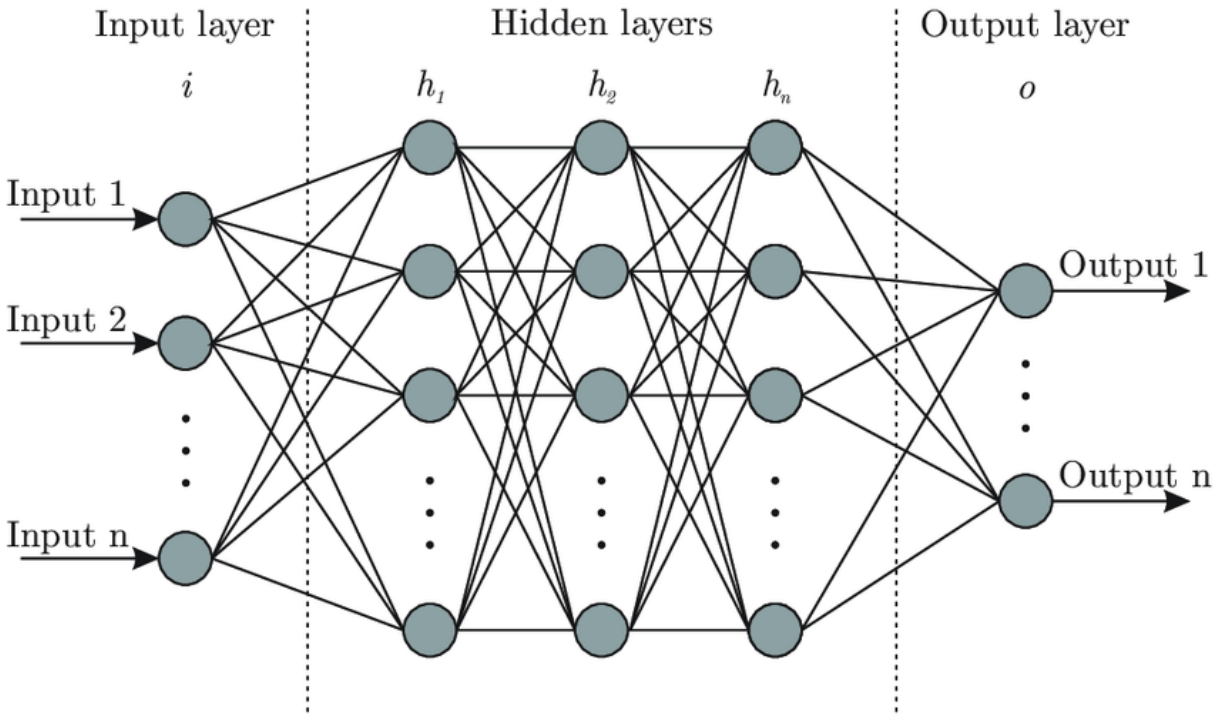

Рисунок 2.2 – Графічне зображення нейронної мережі

2.2 Архітектури нейронних мереж

Нейрона мережа, яка була описана у минулому пункті має назву повнозв’язна, тому що в неї зв’язані виходи усіх нейронів одного шару та всі входи нейронів наступного шару.

Такі нейромережі є дуже потужним інструментом при вирішенні низки задач, але вони мають свої недоліки. Насамперед для формування одного шару, у деяких випадків потрібно мати велику кількість параметрів.

Таким чином, для обробки фотографії, або одного кадру відео розміром 1920 на 1080 пікселів з 3 каналами кольору, потрібно мати 6220801 лише на 1 нейрон повнозв’язної нейронної мережі (формула 2.1). Та це є дійсно великою проблемою, тому що породжує так зване прокляття розмірності, яке полягає в тому, що для навчання нейронної мережі, кількість прикладів повинна бути значно більшою, ніж кількість параметрів, що навчаються.

$$|Pj| = (M+1)\times N, \quad (2.1)$$

де P_j – множина параметрів j-ого шару нейромережі;

M – кількість штучних нейронів минулого шару – j-1(або кількість вхідних сигналів у разі першого прихованого шару);

N – кількість штучних нейронів поточного шару – j (або кількість вихідних сигналів у разі останнього шару).

Також недоліком цієї мережі є те, що всі пікселі (які є входами нейронної мережі), у разі повнозв'язної мережі, є рівнозначними та не враховують відносне розташування, в той самий час, коли в обробці зображень це є вагомо.

Задля рішення цієї та низки інших проблем, були створені спеціальні архітектури нейронних мереж, які будуть розглянуті в поточному пункті.

2.2.1 Конволюційні нейронні мережі (CNN)

Конволюційні нейронні мережі дуже схожі на звичайні нейронні мережі: вони складаються з нейронів, які мають ваги та ухили, що навчаються. Кожен нейрон отримує деякі входи, виконує скалярний добуток і результат необов'язково передається на активаційну функцію. Вся мережа все виражає єдину диференційовану функцію оцінки: від неочищених пікселів зображення на одному кінці до оцінок класу на іншому. І вони все ще мають функцію втрат на останньому (повнозв'язному) шарі [4].

Однак, ця архітектура припускає, що входи – це зображення, що дозволяє нам кодувати певні властивості в архітектурі. Це робить функцію передачі більш ефективною для реалізації та значно скорочують кількість параметрів у мережі.

Конволюційні нейронні мережі користуються тим, що вхід складається з зображень і вони обмежують архітектуру більш розумним

чином. Зокрема, на відміну від звичайної нейронної мережі, шари такої мережі мають нейрони, розташовані в 3 вимірах: ширина, висота, глибина.

CNN складається з шарів, які перетворюють вхідний 3D тензор у вихідний 3D тензор з користуючись деякими диференційованими функціями, які можуть мати або не мати параметрів.

Для побудови архітектури CNN використовуються три основні типи шарів: конволюційний шар, шар пулу та повністю пов'язаний шар (рисунок 2.3).

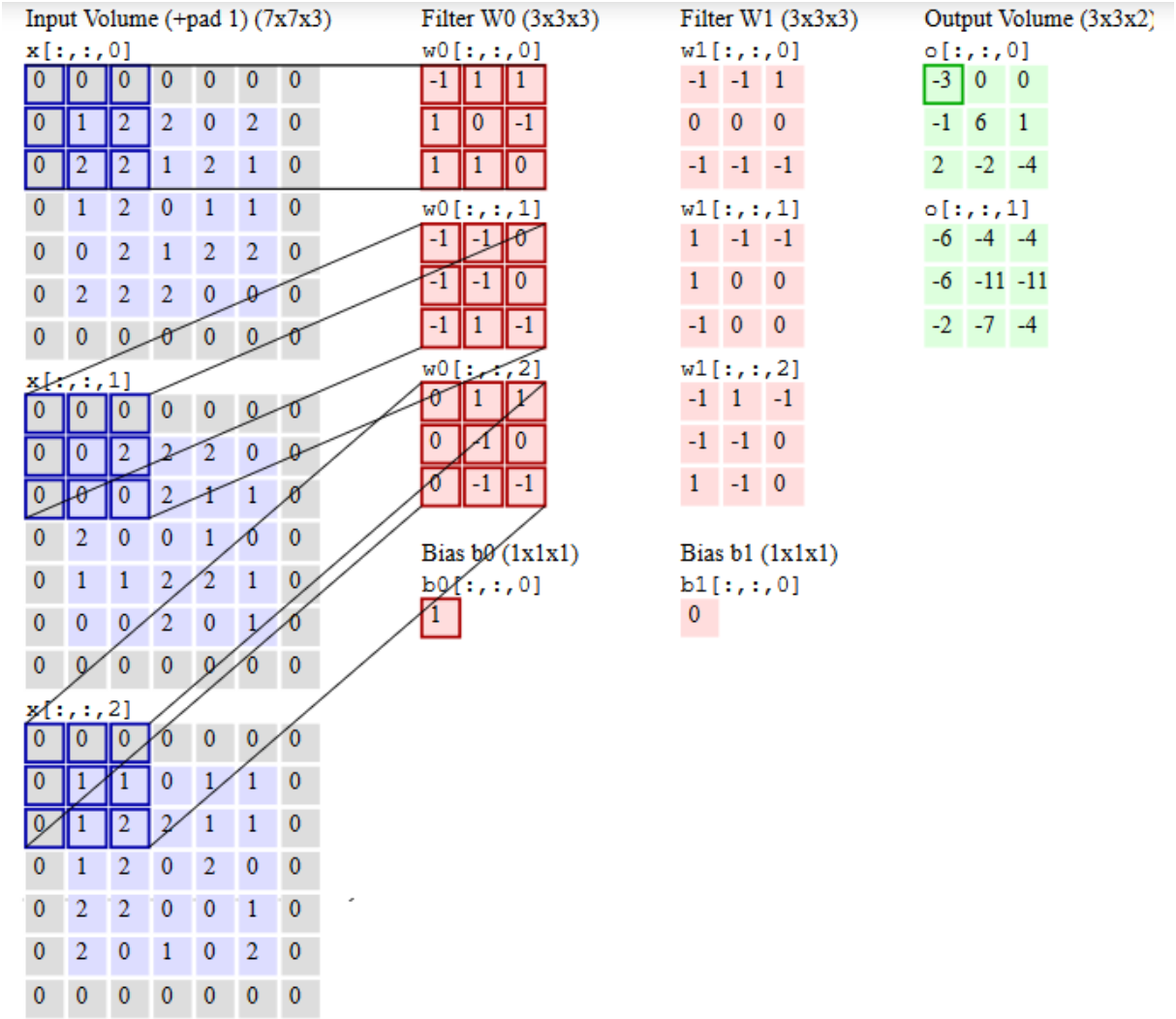

Рисунок 2.3 – Приклад розрахування конволюційного шару на даних розміру 5×5×3 з двома фільтрами 3×3, падінгом 1 та кроком 2

Конволюційний шар складаються з набору фільтрів, що вивчаються. Кожен фільтр невеликий просторово (по ширині та висоті), але поширюється на повну глибину вхідного тензору. Наприклад, фільтр першого шару може мати розмір 5x5x3 (тобто ширина та висота пікселів 5 пікселів і 3, оскільки зображення мають глибину 3, кольорові канали). Під час проходу вперед кожен фільтр просувується по ширині та висоті вхідного об'єму та обчислюється скалярний добуток між параметрами фільтра та входом у будь-якому положенні.

Коли фільтр просувається по ширині та висоті вхідного об'єму, створюється двовимірна карта активації, яка дає вихід цього фільтра у кожному просторовому положенні. Мережа навчає фільтри, які активуються, коли вони побачать певний тип візуальної функції, наприклад, край певної орієнтації або пляма якогось кольору на першому шарі, або, зрештою, цілі сотові або колісоподібні візерунки на вищих шарах мережі . Далі будується цілий набір фільтрів у кожному шарі, і кожен з них створить окрему двовимірну карту активації. Ці карти активації розташовуються по глибинному виміру і створюють вихідний тензор (рисунок 2.4).

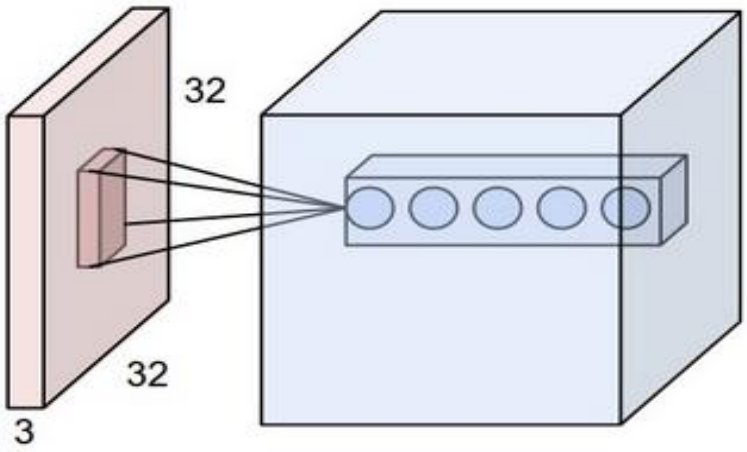

Рисунок 2.4 – Результат обчислення активацій фільтрів
на локальній частці входного сигналу

У роботі з багатомірними входами, такими як зображення, як було зазначено вище, підключати нейрони до всіх нейронів недоцільно. Замість цього з'єднується кожен нейрон лише з локальною областю вхідного обсягу. Просторовий ступінь цього зв’язку – це гіперпараметр, який називається сприйнятливим полем нейрона (це розмір фільтра). Ступінь з'єднання по осі глибини завжди дорівнює глибині вхідного об'єму.

Наступний найбільш поширений шар CNN є шар пулу. Цей шар вставляють між послідовними шарами Conv в архітектуру мережі. Його функція полягає в поступовому зменшенні просторових розмірів представлення для зменшення кількості параметрів і обчислень в мережі, що також контролює надмірну обробку. Шар пулу працює незалежно на кожному фрагменті глибини входу і змінює його просторово, використовуючи операцію MAX, MIN або AVG. Найпоширенішою формою є шар об'єднання з фільтрами розміром 2x2, нанесеним з кроком по 2, понизу на кожен зріз глибини на вході по 2 по ширині та висоті, відкидаючи 75% (у випадку активаційній функції максимуму та мінімуму) активацій. Розмір глибини залишається незмінним.

2.2.2 Рекурентна нейронна мережа (RNN)

На відміну від мереж, розглянутих раніше, рекурентні нейронні мережі сприймають не лише поточний приклад, який вони бачать, а й ті, що вони сприймали раніше. На рисунку 2.5 зображено приклад простої RNN, запропонованої Елманом, де вхідний сигнал «BTSXPE» внизу креслення представляє вхідний приклад у поточний момент, а «CONTEXT UNIT» являє вихід попереднього моменту [5].

Рішення періодичної сітки, досягнутої на етапі часу t-1, впливає на рішення, яке воно досягне через мить на етапі t. Таким чином, RNN мають два вхідних джерела: поточний та минулий – які поєднують для того, щоб визначити, як вони реагують на поточні дані.

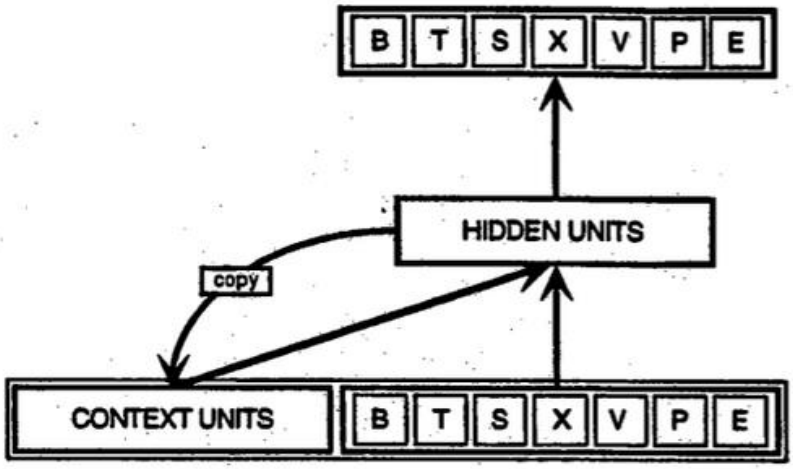

Рисунок 2.5 – Схема RNN, запропонована Елманом

RNN відрізняються від мереж зворотного зв'язку тим, що цикл зворотного зв'язку, пов'язаний з їхніми минулими рішеннями, включає свої власні результати мить як вхід. Так можливо сказати, що періодичні мережі мають пам'ять. Додавання пам’яті до нейронних мереж має наступну мету: інформація є в самій послідовності, а не окремих її частинах, і рекурентні мережі використовують її для виконання завдань, які мережі прямого розповсюдження не можуть виконати.

Ця послідовна інформація зберігається в прихованому стані RNN, та їй вдається пройти багато кроків, оскільки вона просувається вперед, щоб вплинути на обробку кожного нового прикладу. Таким чином, можливо знайти кореляцій між подіями, розділеними серед багатьох моментів часу, і ці кореляції називаються «довготривалими залежностями», оскільки наступна за часом подія залежить від функції, і є функцією однієї або декількох подій, що відбувалися раніше. Так можливо сказати, що в рекурентних мережах ваги поділяються упродовж часу.

Опишемо процес перенесення пам'яті вперед математично:

$$h_t = \varphi(Wx_t+Uh_{t-1}), \tag{2.2}$$

де прихований стан на етапі часу t – h_t, що є функцією від входу для кроку часу x_t, модифікованого ваговою матрицею W, доданої до прихованого стану попереднього кроку часу h_{t-1}, помноженого на його власну матрицю передачі прихованого стану U. Ваги матриці – це фільтри, які визначають, важливість інформації, отриманої при поточному вході та інформації, акумульованої в ході минулих кроків – прихованого стану. Помилка, яку вони генерують, повернеться за допомогою зворотного розповсюдження та буде використовуватися для регулювання ваги.

Сума активується функцією φ – або логістичною сигмоїдальною функцією, або гіперболічним тангенсом, залежно – що є стандартним інструментом для конденсації дуже великих або дуже малих значень у логістичний простір, а також для того, щоб зробити градієнти працездатними для зворотного розповсюдження.

Оскільки цей цикл зворотного зв’язку відбувається на кожному кроці серії, кожен прихований стан містить долю інформації не тільки попереднього прихованого стану, але і всіх тих, що передували h_{t-1} протягом того часу, поки пам'ять може зберігатися.

На рисунку 2.6 ми можемо побачити приклад прямого розповсюдженню сигналу RNN, де кожен x є вхідним прикладом, w – ваги, які фільтрують входи, a – активація прихованого шару (поєднання зваженого вводу та попереднього прихованого стану), а b – вихід прихованого шару після того, як вона активована, використовуючи лінійний або сигмоподібний блок.

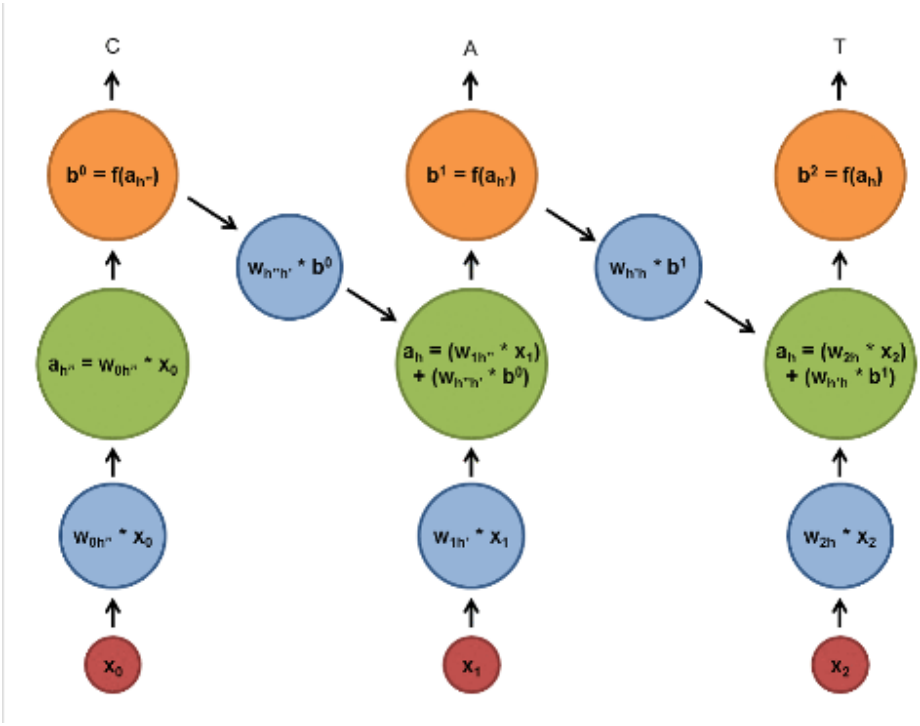

Рисунок 2.6 – Розповсюдження сигналу у RNN

2.2.3 Автоасоціатор

Автоасоціатори є специфічним типом нейронних мереж прямого розповсюдження, у яких вхід такий же, як і вихід. Вони стискають вхід до «коду» меншої розмірності і потім реконструюють вихід з цього подання. Код – це компактний «підсумок» або «стиснення» вхідного сигналу, який також називається представленням латентного простору [6].

Автоасоціатор складається з 3 компонентів: кодера (Encoder), коду (Code) та декодера (Decoder). Кодер стискає вхід і виробляє код, після чого декодер реконструює вхід лише за допомогою цього коду (рисунок 2.7).

Для будування автоасоціатора, необхідно визначити 3 речі: метод кодування, метод декодування та функція втрати для порівняння вихідного сигналу з цільовим.

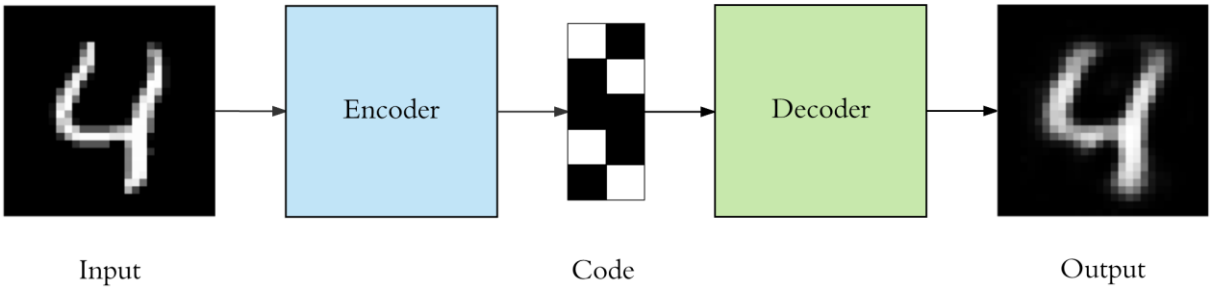

Рисунок 2.7 – Структура нейромережі автоасоціатора

Автоасоціатори мають кілько важливих властивостей:

- специфічні для даних: автоасоціатори здатні лише суттєво стискати дані, аналогічні тим, на яких вони пройшли навчання. Оскільки вони вивчають функції, характерні для даних навчальних даних, вони відрізняються від стандартного алгоритму стиснення даних. Тому не слід очікувати, що автоасоціатор, навчений стискати рукописні цифри, буде стискати фотографії пейзажу;
- значення функції втрат: вихідний не буде точно таким же, як вхід, це буде близьке, але деградоване подання;
- навчання без вчителя: для того, щоб навчити мережу, не потрібно шукати та розмічати вихідні дані, тому що у ролі цільового вихідного сигналу виступає оригінальний вхідний сигнал. Автоасоціатори вважаються технікою навчання без вчителя, оскільки їм не потрібні чіткі мітки для навчання. Але якщо бути точнішим, вони здійснюють самонавчання, оскільки вони генерують власні мітки з навчальних даних.

Давайте вивчимо деталі кодера, коду та декодера. І кодер, і декодер є повністю пов'язаними подаючими нейронними мережами, по суті, ANN, про які ми розглядали в частині 1. Код – це один шар обраної розмірності. Кількість вузлів у рівні коду (розмір коду) - це гіперпараметр, який ми встановили перед тренуванням автоенкодера.

На рисунку 2.8 показана більш детальна візуалізація автокодера. Спочатку вхід проходить через кодер, який є повнозв’язною

нейромережою, для отримання коду. Декодер, який має схожу структуру, потім виробляє вихід лише за допомогою коду. Мета – отримати вихід, ідентичний вхідному сигналу. Найчастіше архітектура декодера – це дзеркальне зображення кодера. Це не є вимогою, але це зазвичай так. Єдина вимога – розмірність вводу та виходу повинна бути однаковою. Внутрішни ж шари можливо настраювати різноманітними способами.

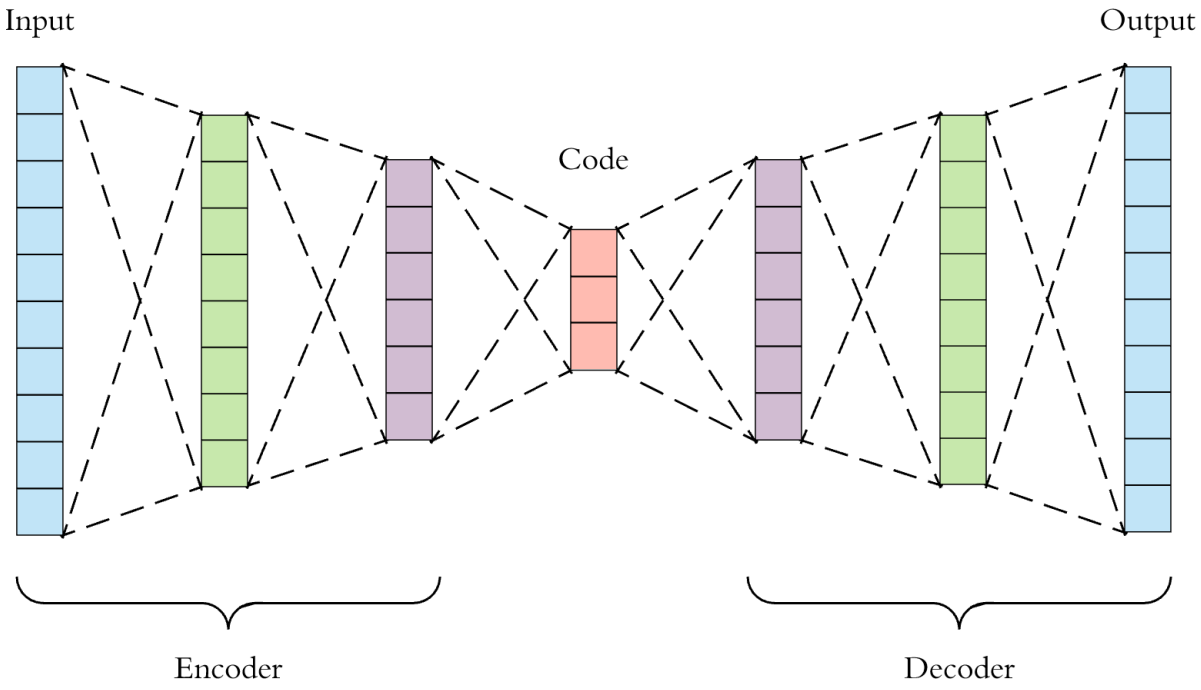

Рисунок 2.8 – Архітектура мережі автоасоціатора

Існує 4 гіперпараметри, які потрібно встановити перед тренуванням автоасоціатора:

- розмір коду: кількість вузлів у середньому шарі. Менший розмір призводить до більшої компресії;

- кількість шарів: мережа може бути настільки глибока, наскільки цього потребує задача. На рисунок 2.8, 2 шари і в кодері, і в декодері, не враховуючи вхід і вихід;

- кількість вузлів на шар: ця архітектура називається складеним автокодером, оскільки шари складаються один за одним. Кількість вузлів

на шар зменшується з кожним наступним шаром кодера і збільшується назад у декодері. Також декодер симетричний щодо кодера з точки зору структури шару. Як зазначалося вище, це не обов'язково, і ці параметни контрольовані розробником мережі;

- функція втрати: найпоширенішими функціями втрати є середня квадратична помилка (mse) та двійкова кросцентропія. Якщо вхідні значення знаходяться в діапазоні [0, 1], зазвичай використовується кросцентропія, у іншому разі використовується середня квадратична помилка.

Автоасоціатори навчаються так само, як нейромережі , за допомогою зворотного розповсюдження.

2.2.4 Генеративна змагальна мережа (GAN)

Генеративні змагальні мережі – це спеціальна архітектура, алгоритм якої використовує дві нейронні мережі, протиставляючи одну проти іншої (таким чином, створюючи «змагання»), з ціллю генерування нових, синтетичні екземпляри даних, які можуть вважатися за реальні дані. Вони широко використовуються для генерації зображень, відео та голосу. GAN були представлені у 2014 році у праці Гудфеллоу та інших дослідників університету в Монреалі [7]. Посилаючись на GANs, директор з досліджень AI у Facebook Ян Лекун назвав змагальний навчання найцікавішою ідеєю за останні 10 років в ML [8].

Потенціал GAN є величезним, оскільки вони можуть навчитися імітувати будь-який розподіл даних. Наприклад, за допомогою GAN можна навчити створювати світи, чудово схожі на наш у будь-якій галузі: образи, музика, промова, проза. Вони в певному сенсі є художниками-роботами, а їхній результат вражаючий. Але вони також можуть бути використані для генерування підробленого медіа-контенту, і це технологія, що лежить в основі DeepFake.

Щоб зрозуміти GAN, потрібно знати, як працюють генеративні алгоритми, і для цього показовим є протиставлення їх дискримінаційним алгоритмам. Дискримінаційні алгоритми намагаються класифікувати вхідні дані; тобто, враховуючи особливості екземпляра даних, вони прогнозують мітку або категорію, до якої належать ці дані.

Наприклад, враховуючи всі слова в електронній пошті (екземпляр даних), дискримінаційний алгоритм може передбачити, чи є це повідомлення спамом чи не спамом. спам є однією з міток, а мішок слів, зібраний з електронної пошти, є функціями, що складають вхідні дані. Коли ця проблема виражається математично, мітка називається y, а функції називаються x. Формулювання p(y|x) використовується для позначення «ймовірності y заданого x», що в цьому випадку перекладається як «ймовірність того, що повідомлення електронної пошти є спамом з урахуванням слів, які вони містять».

Так дискримінаційні алгоритми відображають функції міток. Вони стосуються виключно цього співвідношення. Один із способів думати про генеративні алгоритми – це те, що вони роблять навпаки. Замість того, щоб передбачити мітку з певними функціями, вони намагаються передбачити функції, що надаються певній мітці.

Питання, на яке намагається відповісти генеративний алгоритм, таке: якщо припустити, що цей електронний лист є спамом, наскільки ймовірні ці функції? У той час як дискримінаційні моделі дбають про співвідношення між y та x, генеративні моделі дбають про те, "як ви отримуєте x". Вони дозволяють зафіксувати p(x|y), ймовірність x при заданому y, або ймовірність ознак, заданих міткою або категорією. Таким чином дискримінативний алгоритм відрізняється від генеративного тим, що:

- дискримінаційні моделі вивчають межу між класам;
- генеративні моделі моделюють розподіл окремих класів.

На рисунку 2.9 можна побачити структуру генеративної змагальної мережі, яка складається з двох нейронних мереж. Одна нейронна мережа, яка називається генератором, генерує нові екземпляри імітуючи цільовий розподіл (синтезовані дані), а інша, дискримінатор, оцінює їх достовірності; тобто дискримінатор вирішує, чи належить кожен екземпляр даних, які він переглядає, фактичним набором навчальних даних.

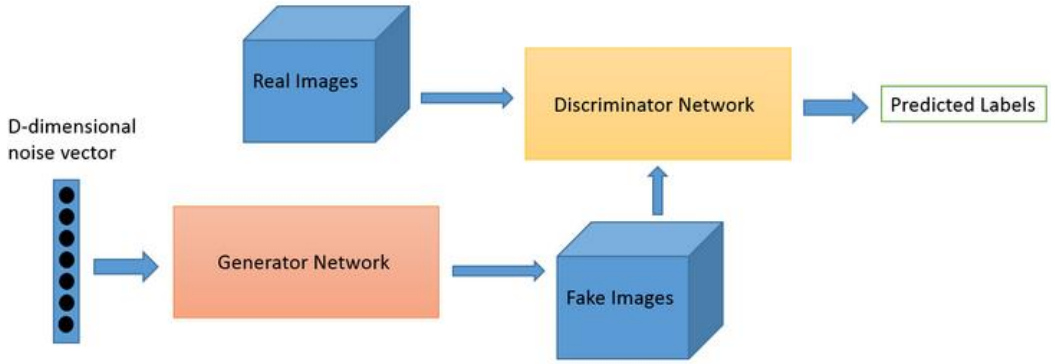

Рисунок 2.9 – Структура GAN

Мета дискримінатора, коли йому показують екземпляр із справжнього набору даних або синтезованого, – визнати ті, які є автентичними. Генератор в той же час створює нові, синтетичні сигнали, які він передає дискримінатору. Це робиться з надією, що вони теж будуть визнані справжніми, навіть якщо вони підроблені. Мета генератора – генерувати сигнали, які будуть визнані дискримінатором, як справжні. Мета дискримінатора – визначити зображення, що надходять від генератора, як підроблені.

Під час своєї роботи GAN іде наступними кроками:

- генератор приймає випадкові числа і повертає зображення;
- це сформоване зображення подається в дискримінатор поряд із потоком зображень, узятих із фактичног набору даних про основну правду;

- дискримінатор приймає як реальні, так і підроблені зображення та повертає ймовірності, число від 0 до 1, причому 1 представляє передбачення справжності, а 0 – фальшивку.

Таким чином утворюються 2 цикли зворотного зв'язку, по яким йде навчання мереж:

- дискримінатор перебуває у циклі зворотного зв’язку із основною істинністю зображень, які ми знаємо;

- генератор знаходиться в циклі зворотного зв'язку з дискримінатором.

Після вдалого навчання мереж для генерації сигналів, що імітують цільовий розподіл достатньо подати на генератор випадкові числа.

3 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Методи викриття deepfakes пропонувалися з того самого часу, як з’явилася перша загроза. З самого початку вони базувались на артефактах та невідповідностях, які виникали внаслідок недостатньо доброї якості синтезування підроблених відеозаписів. Останні ж методи, з іншого боку, застосовують глибинне навчання для автоматичного витягу помітних і відрізняючихся властивостей для викриття підробок.

DeepFake Detection прийнято вважати задачею бінарної класифікації, в якої необхідно визначити чи є даний відеозапис або зображення підробленим чи ні. Такий тип методів вимагає численну кількість даних обох реальних та підроблених відеозаписів для навчання нейронних мереж. Все більша кількість підроблених відео стає доступною, але вона все ще обмежена з точки зору встановлення орієнтиру для перевірки різних методів викриття. Щоб вирішити цю проблему, Коршунов та Марсель створили помітний набір даних із deepfakes, що складається з 620 відео на основі моделі GAN, використовуючи відкритий код Faceswap-GAN. Відео з загальнодоступної бази даних VidTIMIT використовувались для створення низько- та високоякісних відеозаписів, які можуть ефективно імітувати міміку, рухи у роті та моргання очей. Ці ж відео потім використовувались для тестування різних методів виявлення deepfakes. Результати тестів показують, що популярні на той час системи розпізнавання облич на основі VGG та Facenet не в змозі ефективно виявити deepfakes. Інші що базуються на синхронізації губ та показниках якості зображення з використанням SVM, показують дуже високий рівень помилок при застосуванні для виявлення відеофайлів із нового набору даних. Це викликає занепокоєння з приводу критичної потреби в майбутньому розробки більш надійних методів, які дозволяють відрізнити deepfakes [1]. У цьому розділі представлено опис методів викриття deepfakes, де вони групуються на категорії: методи викриття підроблених

зображень та методи викриття підроблених відео (рисунок 3.1). Останні розділені на дві групи: візуальні артефакти в рамках методів, що базуються на кадрах, та часові особливості на основі кадрів. Хоча більшість методів, заснованих на часових особливостях, використовують рекурентні методи глибинного навчання, методи використання візуальних артефактів у відеокадрі можуть бути реалізовані або глибинними, або не глибинними класифікаторами.

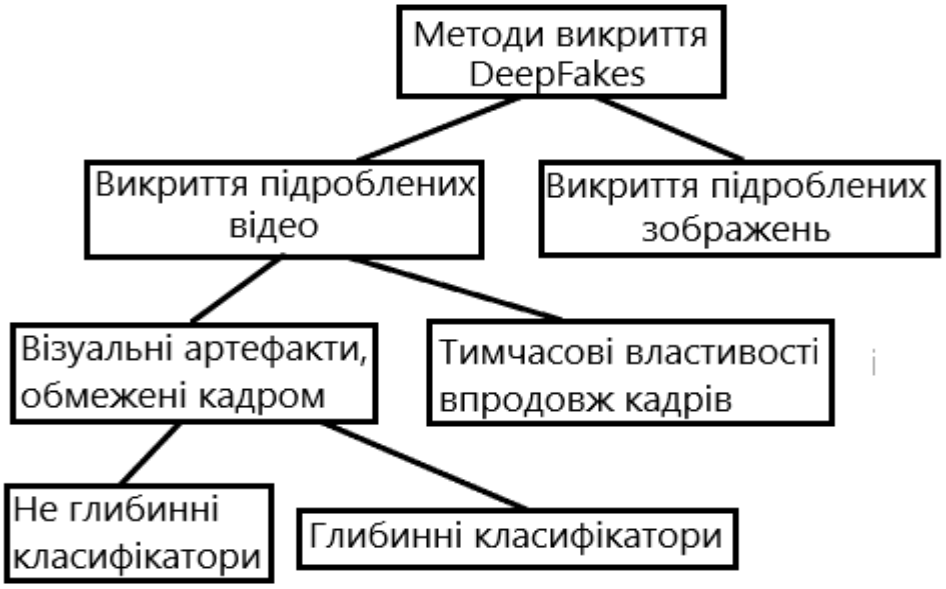

Рисунок 3.1 – Класифікація методів викриття підробок

3.1 Методи викриття підроблених зображень

FaceSwap має ряд переконливих застосувань у композицій відео, трансформації в портретах та особливо в захисті ідентичності, оскільки він може замінити обличчя на фотографіях на одне з колекції зображень. Однак це також одна з методик, яку застосовують для кібератак з проникненням в системи ідентифікації або автентифікації, щоб отримати незаконний доступ. Використання глибинних мереж, таких як CNN та GAN, зробило розміщені зображення складних моделей викриття, оскільки

можливо зберегти позу, вираз обличчя та освітлення фотографій. Чжан та інші використовував метод «мішок слів», щоб витягти набір компактних функцій і подав його в різні моделі класифікації, такі як SVM, Random Forest та Multi-Layer Perceptron для розрізнення підроблених зображень обличчя від справжніх. Серед зображень, породжених глибоким навчанням, синтезовані моделями GAN, ймовірно, найбільш важко виявити, оскільки вони реалістичні та якісні, ґрунтуючись на можливості GAN вивчати розподіл складних вхідних даних та генерувати нові результати з аналогічним розподілом вихідних даних.

Більшість робіт з виявлення зображень, генерованих за допомогою GAN не враховують можливості узагальнення моделей, хоча розробка GAN триває, і багато нових та більш якісних реалізацій GAN впроваджуються. Xuan та ін. використовували етап попередньої обробки зображень, такі як Гауссова розмитість і гаусовий шум, щоб видалити низькі рівні частих властивостей зображень GAN. Це статистично збільшує рівень схожості пікселів між реальними зображеннями та підробленими зображеннями та вимагає, щоб класичний детектор пізнавал більше внутрішніх та значущих особливостей, що є більш узагальненими, ніж попередні методи оцінювання зображень або мережі стегоаналізу зображень.

З іншого боку, Агарвал і Варшні розглядають на основі GAN виявлення deepfakes як проблему тестування гіпотез, коли статистична база була введена за допомогою інформаційно-теоретичного дослідження автентичності. Визначена мінімальна відстань між розповсюдженнями справжніх зображень та зображень, генерованих певним GAN, а саме помилка передбачення. Аналітичні результати показують, що ця відстань збільшується, коли GAN менш точний, і в цьому випадку легше виявити deepfakes. У разі вводу зображень з високою роздільною здатністю потрібен надзвичайно точний GAN для створення deepfakes, які важко виявити.

Останнім часом Hsu та ін. запровадив двофазний метод глибокого навчання для виявлення deepfakes зображень. Перша фаза – це екстрактор

властивостей, заснований на мережі загальних підроблених властивостей (CFFN). CFFN включає в себе кілька щільних шарів, включаючи різну кількість щільних нейронів для поліпшення представницьких можливостей для deepfakes зображень. Кількість щільних одиниць становить три або п’ять залежно від даних валідації обличчя чи загальних зображень, а кількість каналів у кожному блоці змінюється до кількох сотень. Дискримінаційні особливості між фальшивими та реальними зображеннями, тобто парною інформацією, витягуються в процесі навчання CFFN. Ці функції потім подаються у другу фазу, яка є невеликим CNN, з'єднаним з останнім згортковим шаром CFFN, щоб відрізнити оманливі зображення від справжніх.

Запропонований метод підтверджений як для виявлення підробленого обличчя, так і для підробленого зображення взагалі. З одного боку, набір даних для обличчя отриманий від CelebA, що містить 10 177 ідентичностей та 202 539 вирівняних зображень обличчя різних поз та безладного фону. П'ять варіантів GAN використовуються для генерування deepfakes розміром 64x64, включаючи глибокий згортковий GAN (DCGAN), Wasserstein GAN (WGAN), WGAN з градієнтним покаранням (WGAN-GP), GAN з найменшим квадратом та GAN прогресивного росту (PGGAN). Всього для перевірки запропонованого способу отримано 385,198 навчальних зображень та 10 000 тестових зображень як реальних, так і підроблених. З іншого боку, загальний набір даних витягується з ILSVRC12. Широкомасштабна тренувальна модель GAN для синтезу природного зображення високої якості (BIGGAN), GAN з самоувагою та GAN спектральної нормалізації використовується для створення підроблених зображень розміром 128x128. Навчальний набір складається з 600 000 підроблених і реальних зображень, а тестовий набір включає 10 000 зображень обох типів.

3.2 Методи викриття підроблених відеозаписів

Більшість методів виявлення зображень не можна використовувати для відеозаписів через сильну деградацію даних кадру після стиснення відео. Крім того, відеоролики мають тимчасові характеристики, які відрізняються між наборами кадрів і, таким чином, є складними для методів, розроблених для виявлення лише ще підроблених зображень. Цей підрозділ зосереджується на методах виявлення deepfake відеофайлів, і класифікує їх на дві групи: методи, що використовують тимчасові особливості, і ті, які досліджують візуальні артефакти в кадрах.

3.2.1 Тимчасові властивості впродовж кадрів

Виходячи із спостереження, що тимчасова узгодженість не забезпечується ефективно в процесі синтезу deepfakes, Sabir та ін. використали просторово-часові властивості відео потоків для виявлення підробок. Модифікація відео проводиться на кадрах за кадром, таким чином, що артефакти, отримані в результаті маніпуляцій над обличчям, вважаються тими, що проявляють себе як тимчасові артефакти з невідповідностями між кадрами.

Рекурентна згорткова модель (RCN) була запропонована на основі інтеграції згорткової мережі DenseNet та GRU для ефективного використання тимчасових розбіжностей між кадрів (рисунок 3.2). Запропонований метод тестували на наборі даних FaceForensics++, який включає 1000 відео, та показує перспективні результати.

Аналогічно, Guera і Delp підкреслили, що deepfakes відеоролики містять невідповідності внутри кадрів та часові невідповідності між кадрами.

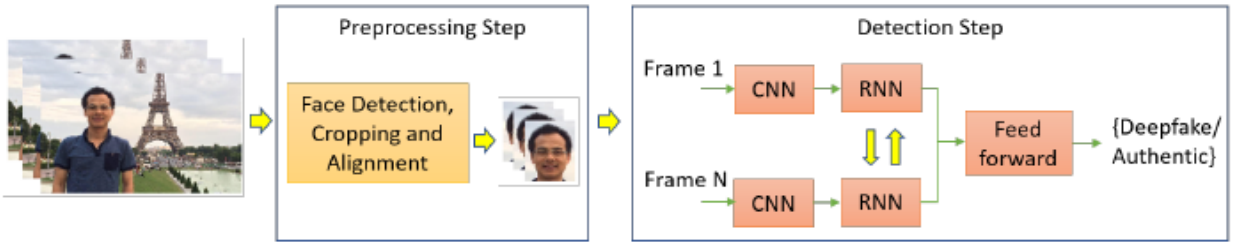

Рисунок 3.2 – Двоступеневий процес для викриття модифікації облич

Потім вони запропонували усвідомлений тимчасовим методом пайплайн, який використовує CNN та довгострокову короткострокову пам'ять (LSTM) для виявлення deepfakes відеофайлів. CNN використовується для отримання властивостей на рівні кадру, які потім подаються в LSTM для створення дескриптора часової послідовності. Після цього повністю підключена 5 мережа використовується для класифікації докторованих відеороликів від реальних на основі дескриптора послідовностей (рисунок 3.3).

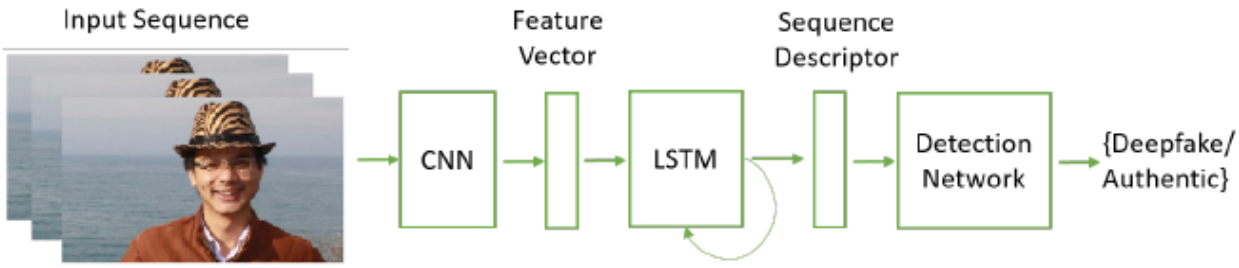

Рисунок 3.3 – DeepFake detection метод Guera і Delp

З іншого боку, використання фізіологічного сигналу – миготіння очей, для виявлення deepfakes ґрунтувалося на спостереженні, що людина у deepfakes має набагато рідше моргання, ніж у незмінених відео. Здорова доросла людина зазвичай моргає раз у десь від 2 до 10 секунд, а кожне моргання триватиме 0,1 та 0,4 секунди. Алгоритми Deepfake, однак, часто використовують зображення обличчя, доступні в Інтернеті для тренувань,

які зазвичай показують людей з відкритими очима, тобто дуже мало зображень, опублікованих в Інтернеті, показують людей із закритими очима. Таким чином, не маючи доступу до зображень людей, що блимають, алгоритми глибокого підробки не мають можливості генерувати підроблені обличчя, які можуть нормально моргати. Іншими словами, частота мерехтіння у флеш-фейках значно нижча, ніж у звичайних відео [11].

Щоб розмежувати реальні та підроблені відео, Li та ін. розкладали відео на кадри, з областю обличчя, а потім витягали області очей на основі шести орієнтирів очей. Після кількох етапів попередньої обробки, таких як вирівнювання граней, вилучення та масштабування обмежувальних границь орієнтирів очей для створення нових послідовностей кадрів, ці обрізані послідовності області очей розподіляються в довгострокові повторювані згорткові мережі (LRCN) для динамічного прогнозування стану. LRCN складається з екстрактора властивостей на основі CNN, навчання послідовності на основі довготривалої короткочасної пам'яті (LSTM) та прогнозування стану на основі повністю пов'язаного шару для прогнозування ймовірності відкритого та закритого стану очей. Мерехтіння очей демонструє сильні часові залежності, і тому реалізація LSTM допомагає ефективно фіксувати ці часові структури. Швидкість моргання обчислюється на основі результатів передбачення, коли блимання визначено, при оцінки вищої порогу 0,5 при тривалості менше 7 кадрів.

Цей метод оцінювався на основі даних, зібраних з Інтернету, що складається з 49 відеороликів для інтерв'ю та презентації та відповідних підроблених відеороликів, згенерованих алгоритмами. Результати експериментів вказують на перспективну ефективність запропонованого способу виявлення підроблених відеороликів, які можна вдосконалити, враховуючи динамічну схему блимання, наприклад. дуже часте моргання також може бути ознакою фальсифікації.

3.2.2 Візуальні артефакти у відеокадрі

Як можна помітити в попередньому підрозділі, методи, що використовують тимчасові шаблони у deepfake відеокадрах, здебільшого базуються на глибинних моделях мереж для виявлення deepfake відеофайлів. Цей підрозділ досліджує інший підхід, який зазвичай розбиває відео на кадри та досліджує візуальні артефакти в межах одного кадру, щоб отримати дискримінантні ознаки. Потім ці властивості поширюються або в глибинній, або в неглибинній класифікатор, щоб розрізняти підроблені віедо та не підроблені відео. Таким чином, ми групуємо методи у цьому підрозділі на основі типів класифікацій, тобто глибинні чи неглибинні.

Відео deepfake, як правило, створюються з обмеженою роздільною здатністю, яка вимагає афінного підходу до викривлення обличчя (тобто масштабування, обертання та інші), щоб відповідати конфігурації оригінальних зобюражень. Через невідповідність роздільної здатності між викривленою поверхнею обличчя та навколишнім контекстом цей процес залишає артефакти, які можна виявити за допомогою моделей CNN, таких як VGG16, ResNet50, ResNet101 та ResNet152. Запропонований метод глибинного навчання для виявлення deepfakes на основі артефактів, що спостерігаються під час кроку викривлення обличчя алгоритмами генерації глибоких фейків. Метод оцінюється на двох наборах даних глибокого підбору даних, а саме UADFV та DeepfakeTIMIT. Набір даних UADFV містить 49 реальних відео та 49 підроблених відео із загальною кількістю 32752 кадрів. Набір даних DeepfakeTIMIT включає набір відео низької якості розміром 64 x 64 та ще один набір відео високої якості розміром 128 x 128 із повністю 10537 незайманими зображеннями та 34 023 виготовленими зображеннями, витягнутими з 320 відео для кожного набору якості. Ефективність запропонованого способу порівнюється з іншими поширеними методами, такими як метод фальсифікацій обличчя

двопотоковий NN, HeadPose і два MesoNet виявлення глубоких фактів, тобто Meso-4 і MesoInception-4.

Перевага запропонованого способу полягає в тому, що він не повинен генерувати відео з deepfake як негативних прикладів перед навчанням моделей виявлення. Натомість негативні приклади генеруються динамічно, витягуючи область обличчя вихідного зображення та вирівнюючи його у декілька масштабів, перш ніж застосовувати розмиття Гаусса до масштабованого зображення випадкового вибору та повернення назад до вихідного зображення. Це зменшує велику кількість часу та обчислювальних ресурсів порівняно з іншими методами, які потребують заздалегідь генерованих фейків.

Нещодавно Nguyen та ін. запропонували використовувати капсульні мережі для виявлення маніпульованих зображень та відеозаписів. Мережа капсул спочатку розглядалася як засіб вирішення обмеженнь CNN, коли їх застосовували до зворотних графічних завдань, які спрямовані на пошук фізичних процесів, що використовуються для створення зображень світу. Нещодавно розроблена капсульна мережа, заснована на алгоритмі динамічної маршрутизації, демонструє її здатність описувати ієрархічні взаємозв'язки між частинами об'єкта. Ця розробка використовується як компонент у конвеєрі для виявлення виготовлених зображень та відео, як показано рисунку 3.4. Алгоритм динамічної маршрутизації розгортається для маршрутизації виходів трьох капсул до вихідних капсул через ряд ітерацій для розділення між підробленими і реальні зображеннями. Метод оцінюється за допомогою чотирьох наборів даних, що охоплюють широкий спектр кованих зображень та відео-атак. Вони включають відомий набір даних повторної атаки науково-дослідного інституту Idiap, набір даних про обмін deepfake обличчя, створений Afchar et al., Набір даних для реконструкції обличчя FaceForensics, що виробляється методом Face2Face, та повністю набір комп'ютерних наборів даних зображень породжені Ra. Запропонований метод дає найкращі показники порівняно з

конкуруючими методами у всіх цих наборах даних. Це показує потенціал мережі капсул у побудові загальної системи виявлення, яка може ефективно працювати для різних кованих зображень та відео-атак.

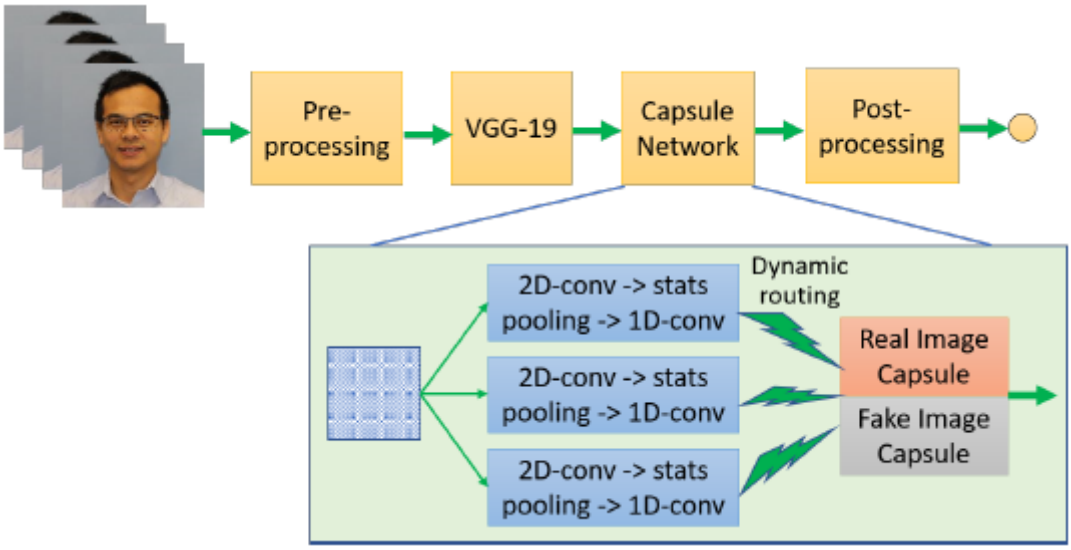

Рисунок 3.4 – Пайплайн Nguyen та ін.

Методи виявлення deepfakes здебільшого покладаються на артефакти чи невідповідність внутрішніх властивостей між фальшивими та реальними зображеннями чи відеозаписами. Yang та ін. запропонували методи виявлення шляхом спостереження за різницею між тривимірними позами голови, що включають орієнтацію та положення голови, які оцінюються на основі 68 орієнтирів обличчя центральної області обличчя. 3D-позиції голови оглядаються, оскільки в пайплайні генерації deepfake обличчя є недоліки. Витягнуті властивості подаються в клас SVM для отримання результатів викриття.

Експерименти на двох наборах даних показують велику ефективність запропонованого підходу проти конкуруючих методів. Перший набір даних, а саме UADFV, складається з 49 глибоких підроблених відеозаписів та відповідних реальних відео. Другий набір даних містить 241 реальне зображення та 252 підроблені зображення, що є підмножиною даних, що

використовуються у DARPA MediFor GAN Image/Video Challenge. Аналогічно було вивчено метод використання артефактів deepfakes та маніпуляцій обличчя на основі візуальних особливостей очей, зубів та обличчя. Візуальні артефакти виникають через відсутність глобальної послідовності, неправильної чи неточної оцінки освітлюваності, що падає, або неточної оцінки основної геометрії.

Для викриття deepfakes використовуються пропущені зображення та відсутні деталі в області очей та зубів, а також текстурні особливості, витягнуті з області обличчя на основі міток на обличчі. Відповідно, використовуються векторний властивостей очей, вектор властивостей зубів та властивостей, витягнуті з повноцінного зображення. Після вилучення функцій два класифікатори, включаючи логістичну регресію та малу нейронну мережу, використовуються для класифікації підробок із реальних відео.

Експерименти, проведені на наборі відеоданих, завантаженому з YouTube, показують найкращий результат 0,851 за площею під AROC. Пропонований спосіб, однак, має недолік, який вимагає зображень, що відповідають певним умовам, таким як відкриті очі або видимість зубів.

Використання аналізу нерівномірностей фотовідповіді (PRNU) було запропоновано для викриття deepfakes. PRNU – це схема шуму, що виникає із дефектів світлочутливих датчиків цифрової камери. PRNU відрізняється для кожної цифрової камери і часто розглядається як друк цифрових зображень. Аналіз широко застосовується у оцінюванні зображень, оскільки розміщене обличчя повинно змінити місцевий зразок PRNU в області обличчя відеокадрів. Відео конвертується у кадри, які обрізаються на потрібну область обличчя. Потім обрізані кадри послідовно розділяють на вісім груп, де для кожної групи обчислюється середня значення моделі PRNU. Нормалізовані показники перехресної кореляції розраховуються для порівняння моделей PRNU серед цих груп.

Автори створили тестовий набір даних, що складається з 10 автентичних відео та 16 модифікованих відео, де підроблені відео були створені з справжніх інструментом DeepFaceLab. Аналіз показує значну статистичну діаграму за середніми нормалізованими показниками перехресної кореляції між глибинними факами та справжніми. Таким чином, цей аналіз має потенціал у виявленні deepfakes, хоча потрібно буде перевірити більший набір даних.

Подивившись відео чи зображення з підозрою, користувачі зазвичай хочуть шукати його походження. Однак наразі немає можливості здійснити такий інструмент. Хасан і Салах запропонували використовувати блокчейн і смарт-контракти, щоб допомогти користувачам виявляти deepfake відео, виходячи з припущення, що відео справжні лише тоді, коли їх джерела відстежуються. Кожне відео пов'язане з розумним контрактом, який посилається на його батьківське відео, і кожне батьківське відео має посилання на свою дитину в ієрархічній структурі. Через цей ланцюг користувачі можуть надійно прослідкувати оригінальний смарт-контракт, пов'язаний із первозданним відео, навіть якщо це відео було скопійовано кілька разів. Важливим атрибутом розумного контракту є унікальні хеші міжпланетної файлової системи (IPFS), яка використовується для зберігання відео та його метаданих децентралізованим та вмістом адресованим. Ключові особливості та функціональні можливості контракту перевіряються на кілька загальних проблем безпеки, таких як розподілене відмову в послугах, перепрогравання та атака людей у середині, щоб запропонувати рішення відповідати вимогам безпеки. Цей підхід є загальним, і його можна поширити на інші типи цифрового контенту, такі як зображення, аудіо та рукописи.

4 ПРАКТИЧНЕ ДОСЛІДЖЕННЯ

4.1 Аналіз даних

Одною з важливіших компонент для успішної моделі є набор даних. В залежності від якості набору даних отримана модель може давати добрі чи погані результати на реальних даних, в залежності від того, чи є вибірка репрезентативною, тобто чи відражає вода приховані закономірності генеральної сукупності.

Для роботи було обрано найсвіжіший набір даних, який був підготовлений до чемпіонату з машинного навчання на ресурсі kaggle.com DeepFake Detection Challenge такими компаніями, як AWS, Facebook, Microsoft, та інші [14].

Цей набір був спеціально створений для цього чемпіонату, а саме для вирішення задачі викриття підроблених за допомогою DeepFake моделей відеозаписів, наступним чином. Організатори найняли різних людей для того, щоб ті створили оригінальні відеозаписи, получили ці відеозаписи та обробили за допомогою низки сучасних DeepFake моделей. Повний тренувальний датасет займає більш ніж 470 ГБ даних, його було розділено на 50 частин розміром близьким до 10 ГБ. Тестовий набір був схований організаторів для забезпечення справедливого оцінювання робіт.

Так як цілий набір даних має дуже великий розмір у рамках цієї роботи обробляється тільки його перша частина. Вона має характерну ознаку: майже на всіх відеозаписів та сама людина, тоді як у інших частинах цього датасету більша варіативність людей.

Відеозаписи набору були підібрані та підготовлені таким чином, щоб опрацювати якнайбільшу частину варіативних компонентів генеральної сукупності:

- була обрана велика кількість людей, які мають різну вік, стать та національність;

- люди стояли у різних позіціях відносно камери та мали різний кут поворота відносно камери;
- відеозаписи були відзняти при разних джерел світла, з разною освітленністю;
- мали місце відеозаписи з більш ніж однією людиною;
- були обрані різні моделі DeepFake, та інше.

На рисунку 4.1 позначені приклади відеозаписів із набору даних у вигляді світлин кадрів у провіднику.

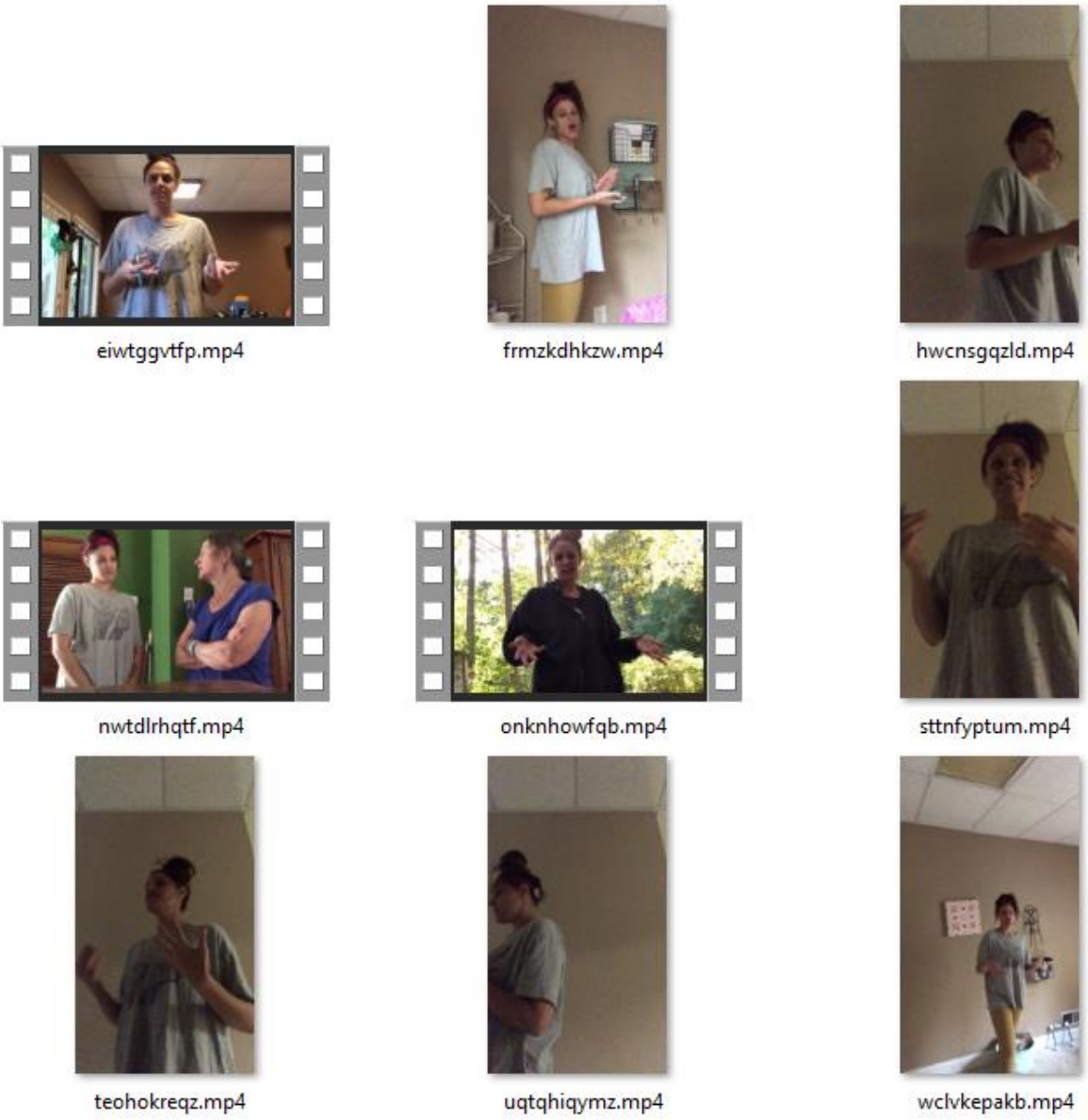

Рисунок 4.1 – Приклади відеозаписів із першої частини

Як ми можемо побачити, велика кількість варіативних компонент (що не стосуються людини, тому що, як було зазначено раніше, особливість цієї частини датасета саме у тому, що в більшості відеозаписів присутня та сама людина).

Також в наборі даних присутні файли метаданих, що включають в себе наступні дані:

- ім’я відеозапису;
- помітку, підроблена ця відеозапис, чи ні;
- у разі якщо це підробка – оригінальне відео з цієї ж вибірки.

На рисунку 4.2 зображені характеристики першої частини датасета. Як ми можемо побачити, вибірка дуже небалансована, причому більше підроблених даних, ніж оригінальних. Це є специфікою даної вибірки, тому що на один оригінальний екземпляр дається декілька (інколи навіть більше ніж 30) підроблених. Та в даної частині датасету оригінальних відеозаписів лише 6,4%.

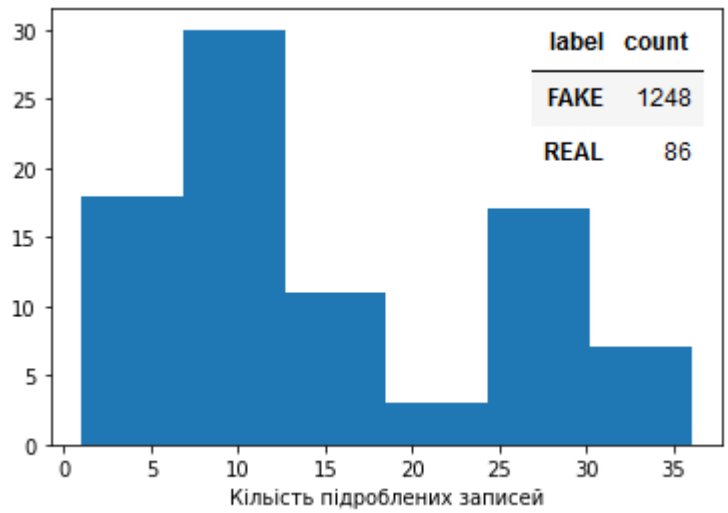

Рисунок 4.2 – Характеристики частини даних: співвідношення підроблених даних та реальних кількісно, та розподіл кількості підроблених екземплярів на один оригінальний відеозапис

Однак у реальному житті на цей час цифри протилежні та навіть з набагато більшим відривом. Це потрібно мати на увазі при навчанні моделі.

4.2 Проєктування системи

Створення підробок є дуже складною системою, яка складається багатьох елементів. Таким же складним є створення методів викриття підробок.

До найпростішої моделі можуть відноситися наступні елементи:

- обробник зображень – елемент, необхідний для зчитування, відображення зображень, або кадрів з відео, а також окремих їх частин;
- модуль попередньої обробки – для підготовки зображення або відео для вхідного формату нейромережі (або іншого класифікатору);
- саме класифікатор (нейрона мережа певної структури або інший класифікатор);
- великий обсяг розмічених даних для навчання моделі;
- дані для тестування (оцінки) моделі.

Для більш складних мереж можуть використовуватися на додаток до вказаних раніше:

- низка нейронних мереж або інших моделей для витягу конкретних частин зображення (обличчя, глаз, зубів, та інше) як вхідних даних для спеціалізованих класифікаторів, які можуть працюють разом із іншими моделями у ансамблі;
- доповнені розмічені дані для навчання відповідних мереж;
- окремий класифікатор звукової доріжки (для відеозаписів), який може виступати разом з аналізатором відеопотоку для контролю їх узгодженості;
- інші елементи.

В даній роботі пропонується рішення, основане на обробці зображень, тобто окремих кадрів відеозаписів.

Для створення якісної моделі машинного навчання, найчастіше можливо враховувати загальні знання домену, які значно спрощують задачу.

Якщо напряму без незайвих помислів розв’язувати задачу аналізу відеозаписів (в тому числі задачу викриття підробок), то можна використати багато ресурсів не прийти к жаданому результату.

Для задачі, що розглядається в цієї роботи, це знання, що обробляєма частина відеоролика є саме частина обличчя, таким чином, нема необхідності створювати мережу, яка буде обробляти цілий відеозапис у повному форматі.

У запропонованому методі використовується обмеження обробляємої частити наступним чином. У якості попередньої обробки, кожний відеозапис розділяється на кадри, в яких виділяються обличчя і відокремлюються від повного зображення.

Для цього пропонується використовувати модель MTCNN, яку можливо віднайти в відкритому доступі з параметрами, що були навчені. Її окремим плюсом є то, що вона може видавати на виході зображення обличчя у заданому розмірі, що спрощує подальшу обробку, так як усі виходи будуть в одному форматі незалежно від його положення та масштабу.

Наступним кроком стає навчання конволюційної нейронної мережі, яка реалізує класифікатор. В якості базової моделі було обрана наступна архітектура (рисунок 4.3):

- конволюційний шар з ядром 3 на 3, паддінгом 1 та кроком 1, таким чином не змінюючи розміру, без активаційної функції, з утворенням 8 фільтрів;
- другий конволюційний шар з такими самими ядрами, але з 16 фільтрами та функцією активації – ReLU, таким чином образующи менше параметрів на більше сприятливе поле (5 на 5 із допомогою двох згорткуваних шарів 3 на 3);
- шар батч-нормалізації, який повинен ускорювати навчання;

- шар пулу, с кроком та розміром 2, для зменшення зображеня в 2 рази;
- такі ж 4 шари але тепер з відповідно 32 фільтрами;
- ще один конволюційний шар 3 на 3, з тією самою активаційною функцією;
- спеціальний шар пулінгу (як в ResNet), який розраховує середнє значення кожного фільтру, в результаті чого утворюється 32 нейрона; дає можливість подавати зображення майже будь-якого розміру;
- повнозв'язний шар з 64 штучними нейронами, та активаційною функцією – тангенс гіперболічний;
- останній шар з 1 вихідним нейроном для отримання результату бінарної класифікації (без активаційної функції, для прискорення навчання).

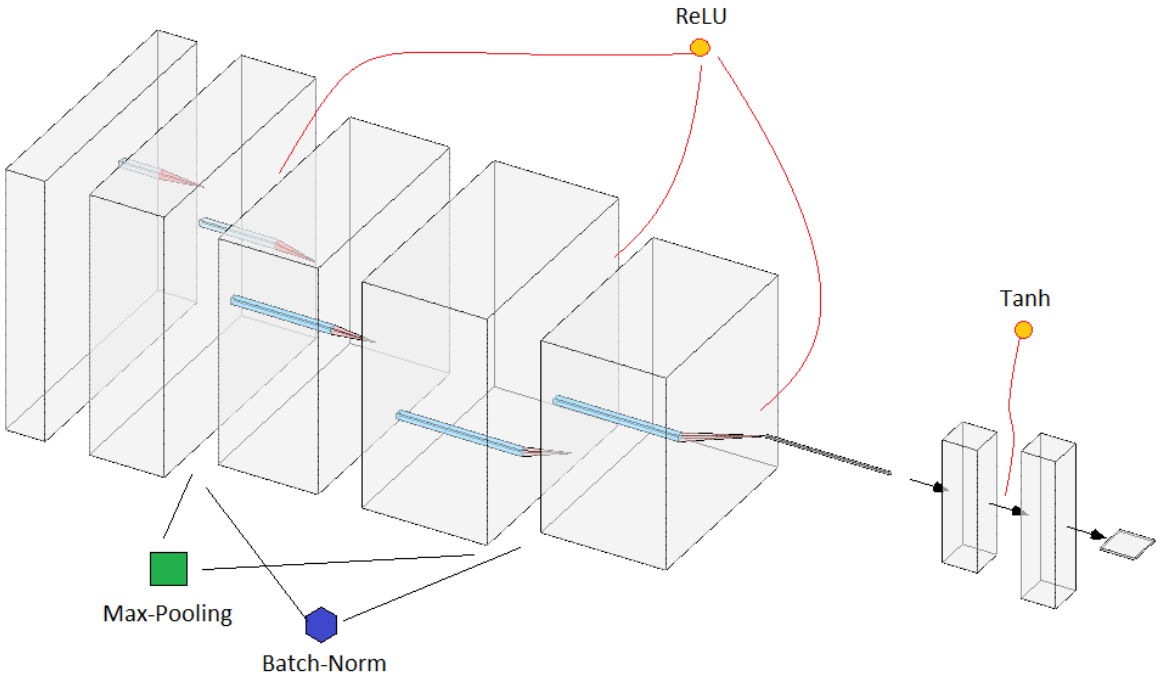

Рисунок 4.3 – Архітектура базового рішення

4.3 Програмне забезпечення

Для реалізації системи викриття підробок було прийнято рішення використовувати муву програмування Python 3.

Python – це легка у вивченні, потужна мова програмування. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Простий синтаксис і динамічний набір Python разом з інтерпретованою природою роблять його ідеальною мовою для створення сценаріїв та додатків швидкого розвитку у багатьох областях на більшості платформ.

Інтерпретатор Python та обширна стандартна бібліотека є у вільному доступі у вихідному чи двійковому вигляді для всіх основних платформ веб-сайту Python https://www.python.org/ та може розповсюджуватися вільно. Цей же сайт також містить дистрибутиви та покажчики на багато безкоштовних сторонніх модулів, програм та інструментів Python та додаткову документацію.

Інтерпретатор Python легко розширюється новими функціями та типами даних, реалізованими на C або C ++ (або інших мовах, які можна викликати з C). Python також підходить як мова розширення для налаштованих програм [15].

Останнім часом Python набрав велику популярність, із-за великої та швидко зростаючою кількості бібліотек, зокрема для штучного інтелекту, машинного навчання, нейромереж, обробки даних у вигляді таблиці, чи аудіо або зображень.

Таким чином, Python краще за всі мови програмування підходив під цей проєкт.

В роботі використовувались бібліотеки numpy для матричних обчислень та matplotlib для візуалізації.

Для створення моделей глибинного навчання на високому рівні було обрано фреймворк PyTorch. Це система машинного навчання з відкритим

кодом, що прискорює прототипування досліджень та розгортання виробництва.

Його ключові особливості:

- готовий до виробництва. Він має рішення для швидкого прототипування рішень та переходу на виробництво за допомогою TorchScript і TorchServe;

- розподілене навчання. Він має здатність масштабувати розподілене навчання та оптимізує ефективність в науково-дослідних і виробничих процесах за допомогою бекэнду torch.distributed;

- надійна екосистема. Багата екосистема інструментів та бібліотек розширює PyTorch та підтримує розвиток у комп'ютерному зорі, NLP тощо;

- хмарна підтримка. PyTorch добре підтримується основними хмарними платформами, забезпечуючи розвиток без тертя та просте масштабування [16].

Для роботи з зображеннями і відео було використано крос-платформну бібліотеку з відкритим кодом OpenCV. Вона відрізняється оптимізацією свого коду, та можливістю використовувати її безкоштовно задля комерційної діяльності [17].

4.4 Програмна реалізація та результати

В ході роботи було використано 2 середи розробки: Jupyter Notebook та PyCharm. За допомогою Jupyter Notebook велась розробка, та остаточні результати у вигляді скриптів-модулів були створені в PyCharm, їх можна побачити у додатку А.

Всі розрахунки велись у PyCharm. В той час, як дослідження і тести велись за допомогою Jupyter Notebook.

Як вже було зазначено у пункті з проєктуванням, система складається з двох основних масивних (не за рядками коду, а за функцією, що вони виконуюсь) модулів: preprocessing.py, model.py.

Система складалась з двох кроків:

- розкладання відеозапису на зображеня набір кадрів облич, розміром 200 на 200 пікселів, за допомогою MTCNN;

- класифікація отриманих зображетнь облич за допомогою розрабляємої нейроної мережі.

4.2.1 Витягання облич

Для витягання облич була використана модель MTCNN, яка була представлена у 2016 року Жангам та іншими, та портована з TensorFlow розробником timesler [18].

Ця бібліотека має назву facenet-pytorch та має здатність детектувати обличчя на зображенні, а також деякі опорні крапки (такі як контури губ, ніс та інше). На рисунку 4.4 показано приклад, як за допомогою цієї бібліотеці можна детектувати обличчя.

Рисунок 4.4 – Приклад знаходження обличча з відеозапису із другої частини вибірки даних

На основі цієї бібліотеці був створений код для витягання облич. Основну функцію цього коду можна побачити в додатку А в файлі preprocessing.py, з назвою «extract_store_faces».

Ця функція приймає 3 основних параметри: ім’я файлу, розташування файлу, та цільове розташування.

За допомогою бібліотеки OpenCV, було проведено зчитування кожного кадру відеозапису, а далі за допомогою MTCNN були витягнути обличчя, як на рисунку 4.5:

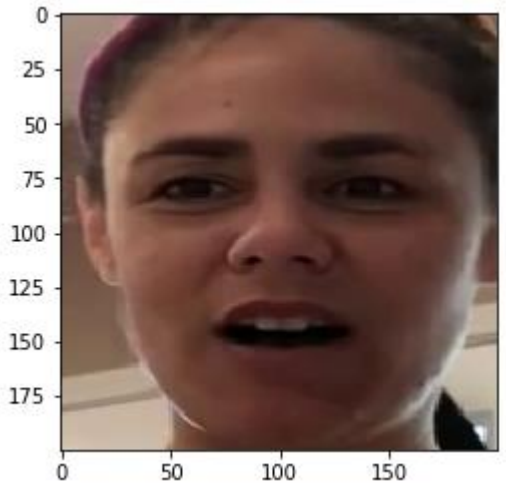

Рисунок 4.5 – Приклад витягнутого обличчя з першої частини датасету

Таким же чином були витягнуті всі обличчя, які було можливо витягнути за допомогою наданої моделі із кожного файлу, та навіть більше. Інколи нейромережа вважала, що плями на об’єктах, або відблиски сонця також вважалися за обличчя. Однак якщо на кадрі було виявлено декілька обличь, то вони зберігалися за порядком зменшення впевненості мережі, тому було можливо в нашому випадку брати тільки перше обличчя для кожного кадру.

Таким чином, за допомогою цього класу було сформовано близько 400 тисяч зображень із обличчями, які загалом були у стислому форматі важили більш ніж 4 ГБ.

4.4.2 Створення класифікатору

В другому модулі model.py в додатку Б, ми можемо побачити 2 основних класи: FakeDataset та DeepFakeDetectior.

Так як дані зберігаються на жорсткому диску, тому що не має можливості вмістити усе в оперативну пам’ять, тим більше пам’ять відеокарти, на якої були усі основні обчислення, було потрібно створити клас для доступу к тренувальній вибірці.

В ньому були основні приховані функції підготовки даних з метафайлів та зображень для подачі в нейрону мережу.

Клас DeepFakeDetectior відповідає моделі описаної в пункті 4.1, побудованій для фреймворку pytorch.

Також там написана функція для навчання будь-якої моделі с вибраними датасету написаного класу. Всі параметри навчання та метод навчання можна побачити в функції «train» в тому самому файлі.

Код був написаний так, щоб було можливо вчити модель та перевіряти на валідаційній виборці, навіть якщо вона не може бути поміщена пам’ять цілком.

На рисунку 4.6 ми можемо побачити графік зміни функції втрат для тренувальної та валідаційної вибірках.

Як ми можемо побачити навчання не протікає вдало. Після першої епохи суттєвих змін не відбувається.

Це показує, наскільки складно обучити модель для задачі викриття підробок, при умові незбалансованої вибірці.

Отримана модель мала високий показник точності, однак це було досягнуто за рахунок дуже великого дисбалансу класів у обох тренувальній та валідаційній вибірках.

В підтвердження того на рисунку 4.7 можна побачити батч, у якому усі дані були підроблені.

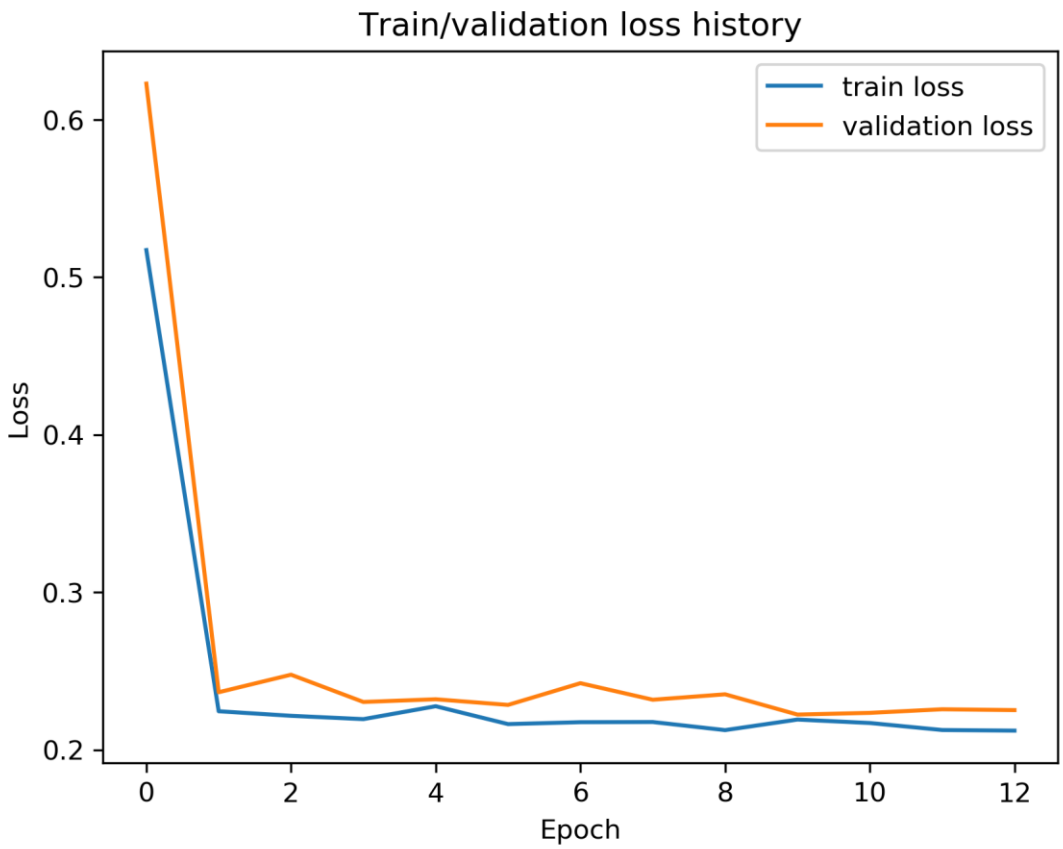

Рисунок 4.6 – Хід навчання моделі

```
In [24]:  1 y_train[start-batch_size:start]

Out[24]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
                0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
                0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
                0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

Рисунок 4.7 – Приклад поганого батчу

Таким чином, можна зробити висновок, що для бінарної класифікації сучасних підробок недостатньо створити просту не дуже глибоку модель з нуля, що навчалась на незбалансованій вибірці, без додаткової аугментації. Це все й стало основними недоліками обраного підходу.

ВИСНОВКИ

В ході атестаційній роботи магістра було розглянуто проблеми, пов’язані з виникненням таких речей як DeepFake, було проведене дослідження предметної галузі, та були розглянуті історичні досягнення по викриттю підробок.

С кожним роком системи штучного інтелекту приносять великий вклад в розвиток суспільства та економіки. Однак, в той самий час, розвинуті технології починають використовувати для поганих намірів.

Так, роботи, які «навчили штучний інтелект мріяти», а саме генеративні змагальні мережі, почали використовувати задля модифікації відеозаписів, тим самим компрометуючи цільових осіб, які брали участь в відеозаписі, та змусивши людей «казати» та «робити» речі, які насправді вони ніколи могли не робити чи казати.

Поява таких відеозаписів, яким не можна довіряти, ставить крапку на епосі, коли можливо було «вірити своїм очам».

Для подолання цього спочатку Google, а потім AWS, Facebook, Microsoft та інші створили свої набори даних для розробки алгоритмів, що будуть викривати підробки. Останні створили чемпіонат для поширення свідомості та залучення світових спеціалістів з машинного навчання.

В даній роботі були проаналізовані дані з чемпіонату. Та була створена дещо спрощена модель детектора. У зв’язку з обмеженістю ресурсів була здійснена спроба по вирішенню легшої проблеми: на датасеті с однією людиною визначити підроблені записи.

Ця спроба показала, що навіть спрощена задача є дуже складною задачею машинного навчання і те, як сильно просунулися моделі створення DeepFakes.

ПЕРЕЛІК ПОСИЛАНЬ

1. Deep Learning for Deepfakes Creation and Detection. URL : https://arxiv.org/abs/1909.11573 (дата звернення: 13.04.2020).

2. Faceswap. URL : https://github.com/deepfakes/faceswap (дата звернення: 15.04.2020).

3. Искусственная нейронная сеть. URL : http://www.machinelearning.ru/wiki/index.php?title=Искусственная_нейронная_сеть (дата звернення: 13.04.2020).

4. Convolutional Neural Networks. URL : https://cs231n.github.io/convolutional-networks/ (дата звернення: 14.04.2020).

5. A Beginner's Guide to LSTMs and Recurrent Neural Networks. URL : https://pathmind.com/wiki/lstm (дата звернення: 15.04.2020).

6. Applied Deep Learning. Part 3: Autoencoders. URL : https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798 (дата звернення: 16.04.2020).

7. Goodfellow, Ian J. et al. (2014). «Generative Adversarial Nets». In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. NIPS’14. Cambridge, MA, USA: MIT Press, pp. 2672–2680. URL: http://dl.acm. org/citation.cfm?id=2969033.2969125. (дата звернення: 10.04.2020)

8. Generative Adversarial Network. URL : https://pathmind.com/wiki/generative-adversarial-network-gan (дата звернення: 18.04.2020).

9. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. URL : https://ieeexplore.ieee.org/document/7803544 (дата звернення: 15.03.2020).

10. Guo, Y., Jiao, L., Wang, S., Wang, S., and Liu, F. Fuzzy sparse autoencoder framework for single image per person face recognition. URL : https://ieeexplore.ieee.org/document/8017477 (дата звернення: 18.03.2020).

11. Lyu, S. Detecting deepfake videos in the blink of an eye. URL : http://theconversation.com/detecting-deepfake-videos-in-the-blink-of-an-eye-101072 (дата звернення: 21.03.2020).

12. Bloomberg. How faking videos became easy and why that’s so scary. URL : https://fortune.com/2018/09/11/deep-fakes-obama-video/ (дата звернення: 22.03.2020).

13. Chesney, R., and Citron, D. Deepfakes and the new disinformation war: The coming age of post-truth geopolitics. URL : https://www.foreignaffairs.com/articles/world/2018-12-11/deepfakes-and-new-disinformation-war (дата звернення: 25.03.2020).

14. Deepfake Detection Challenge. URL : https://www.kaggle.com/c/deepfake-detection-challenge/overview (дата звернення: 13.02.2020).

15. Python Documentation URL : https://docs.python.org/3/tutorial/index.html (дата звернення: 07.05.2020).

16. PyTorch. URL : https://pytorch.org/ (дата звернення: 09.05.2020).

17. OpenCV. URL : https://opencv.org/ (дата звернення: 08.05.2020).

18. Facenet-pytorch. URL: https://github.com/timesler/facenet-pytorch (дата звернення: 07.05.2020).