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)

Дослідження методів класифікації зображень, отриманих з використанням технології DeepFake
(тема)

Виконав:
студент 2 курсу, групи СШІм-18-2

Білоцерковський В.В.
(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 доц. Чала Л.Е.
(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 та інформаційні технології
(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)
«____»_____20____р.

# ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Білоцерковському Владиславу Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів класифікації зображень, отриманих з використанням технології DeepFake
затверджена наказом по університету від 30.03.2020 р. № 480Ст

2. Термін подання студентом роботи до екзаменаційної комісії 19 травня 2020 р.
3. Вихідні дані до роботи Точність класифікації описаних моделей на різнорідних даних, що містять дані, згенерованих алгоритмом DeepFake

4. Перелік питань, що потрібно опрацювати в роботі
Вступ; 1 Нейронні мережі; 2 Опис технології DeepFake; 3 Методи виявлення підроблених зображень; 4 Практичні дослідження методів виявлення підроблених зображень; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

_____
_____
_____
_____
_____

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 |
| Аналіз предметної галузі | доц. Чала Л.Е. | | 13.04.2020 |
| Методи і алгоритми обробки зображень | доц. Чала Л.Е. | | 21.04.2020 |
| Експериментальне моделювання | доц. Чала Л.Е. | | 30.04.2020 |
| Розробка моделі виявлення DeepFake | доц. Чала Л.Е. | | 07.05.2020 |

**КАЛЕНДАРНИЙ ПЛАН**

| № | Назва етапів роботи | Терміни виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Аналіз літератури за темою дослідження | 11.04.2020 | виконано |
| 2 | Аналіз згорткових мереж | 13.04.2020 | виконано |
| 3 | Аналіз похідних блоків алгоритму DeepFake | 15.04.2020 | виконано |
| 4 | Опис алгоритму DeepFake | 18.04.2020 | виконано |
| 5 | Дослідження існуючих методів розпізнання DeepFake | 24.04.2020 | виконано |
| 6 | Пошук та підготовка необхідного набору даних | 26.04.2020 | виконано |
| 7 | Робота з програмним забезпеченням | 30.04.2020 | виконано |
| 8 | Проведення дослідження та отримання результатів | 03.05.2020 | виконано |
| 9 | Оформлення пояснювальної записки | 05.04.2020 | виконано |
| 10 | Оформлення графічної частини | 14.04.2020 | виконано |
| 11 | Захист атестаційної роботи | 20.05.2020 | |

Дата видачі завдання ____30 березня_2020 р.

Студент _____
(підпис)

Керівник роботи _____ _____доц. Чала Л.Е._____
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ

Пояснювальна записка містить: 76 сторінки, 34 рисунків, 21 використаних джерел, 1 додаток.

НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ МЕРЕЖІ, AUTOENCODER, GAN, DEEPFAKE, EFFICIENTNET, ATTENTION, DCDF

Об'єкт дослідження – процес створення моделі для розпізнавання DeepFake.

Предмет дослідження – математичні, програмні засоби і методи створення моделей для розпізнавання підробних зображень.

Мета роботи – дослідження існуючих способів розпізнавання контенту, згенерованого за допомогою технології DeepFake.

Методи дослідження – методи класифікації зображень, методи оцінки роботи згортальних мереж, методи препроцессінгу зображень.

# РЕФЕРАТ

Пояснительная записка содержит: 76 страниц, 34 рисунка, 21 использованный источник, 1 приложение.

НЕЙРОННЫЕ СЕТИ, СВЕРТОЧНЫЕ СЕТИ, AUTOENCODER, GAN, DEEPFAKE, EFFICIENTNET, ATTENTION, DCDF

Объект исследования – процесс создания модели для распознавания DeepFake.

Предмет исследования – математические, программные средства и методы распознавания поддельных изображений.

Цель работы – исследование существующих способов распознавания контента, сгенерированного благодаря технологии DeepFake.

Методы исследования – методы классификации изображений, методы оценки работы сверточных сетей, методы препроцессинга изображений.

# ABSTRACT

The explanatory note contains: 76 pages, 34 pictures, 21 sources used, 1 application.

NEURAL NETWORKS, CONVOLUTINAL NETWORKS, AUTOENCODER, GAN, DEEPFAKE, EFFICIENTNET, ATTENTION, DCDF

The object of research – a process of creation of model for recognition of DeepFake.

The subject of research – mathematical, programmatic facilities and methods due to that there will be a possible achievement of research aim.

The purpose of the work – research of existent methods of recognition of the content generated due to technology of DeepFake.

Research methods – methods classification images methods, methods of estimation of work of convolutional networks, preprocessing images methods.

**ЗМІСТ**

Вступ ..... 8
1 Розвиток технологій завдяки яким з’явилися алгоритми генерації підробних зображень ..... 10
1.1 Спосіб дії нейронних мереж ..... 10
1.2 Згорткові нейронні мережі ..... 16
1.3 Autoencoder ..... 25
1.4 Постановка задач дослідження……...256
2 Опис технології DeepFake ..... 28
2.1 Генеративно-змагальні нейронні мережі ..... 28
2.2 Технологія DeepFake ..... 33
2.3 Модель fsgan ..... 39
3 Методи виявлення підроблених зображень ..... 41
3.1 Рекурентні нейронні мережі для виявлення підроблених зображень ... 41
3.2 Виявлення підробних зображень по біологічним сигналам ..... 42
3.3 Mesonet компактна нейромережа для виявлення підроблених відео .... 45
3.4 Виявлення артефактів деформації обличчя ..... 47
3.5 Виявлення зображень згенерованих за допомогою DeepFake використовуючи підхід парного навчання ..... 49
4 Практичні дослідження методів виявлення підроблених зображень ..... 54
4.1 Виявлення підроблених зображень за допомогою ансамблю згортальних мереж ..... 54
4.2 Експерименти виявлення підроблених зображень з використання ансамблю згортальних класифікаторів ..... 58
4.3 Результати інших досліджень на вибірці dfdc ..... 66
Висновки ..... 69
Перелік джерел посилання ..... 70
Додаток А Приклади розподілу даних ..... 72
Додаток В Відомість атестаційної роботи……..……...76

# ВСТУП

DeepFake (випливає з «глибокого навчання» і «підробки») − це техніка, яка може накладати зображення обличчя, змінюючи одну людину замість іншої на відео чи зображенні, що дозволяє створити нове відео, на якому одна людина говорить або робить те, що робила інша людина з похідного зображення чи відео. Моделі глибокого навчання, такі як Autoencoder та генеративні змагальні мережі широко застосовується в області комп'ютерного зору для вирішення різних проблем. Ці моделі також використовували алгоритми DeepFake для вивчення міміки, рухів людини та їх синтезу для перетворення обличчя іншої людини, що робить аналогічні вирази обличчя та рухи. Для алгоритма DeepFake, зазвичай, потрібна велика кількість зображень і відеоданих, щоб натренувати модель для створення фотореалістичних зображень та відео. Оскільки громадські діячі, відомі люди та політики, можуть мати велику кількість зображень і відео, доступних в Інтернеті, вони є найпоширенішою мішенню для застосування алгоритму DeepFake.

Внаслідок того, що проблема, викликана негативним застосуванням DeepFake по відношенню до людини, потреба у використанні все менших ресурсів (мінімальної кількості необхідних наборів даних для навчання моделі). Проблема, що описана в даній роботі, є актуальною на момент дослідження.

На даний момент існують численні методи запропоновані для виявлення DeepFake. Більшість з них базується також на глибокому навчанні, і, таким чином, в битві між ними з'явилися негативні та позитивні методи глибокого навчання.

Визначення актуальності теми дозволяє сформулювати досить конкретну мету роботи, яка обумовлена спробою дослідити ситуацію розвитку існуючих технологій виявлення DeepFake та на їх основі створити узагальнену модель розпізнавання DeepFake, яка буде давати більш точний

результат в роботі із складними підробними зображеннями та відео. Таким чином, у цій роботі представлені описи методів створення, а також виявлення DeepFake.

Об’єктом цього дослідження є процес створення моделі для розпізнавання DeepFake.

Предметом дослідження є математичні, програмні засоби і методи, завдяки яким буде можливе досягнення мети дослідження.

В основу виконаної роботи покладена наступна гіпотеза: застосовуючи найбільш сучасну архітектуру згортальних мереж, а також Attention механізми, можливо створити узагальнену модель, що показує достатні результати на різнорідних даних.

Таким чином атестаційна робота складається з: вступу, чотирьох частин, висновків, переліку джерел посилання та одного додатку.

# 1 РОЗВИТОК ТЕХНОЛОГІЙ, ЗАВДЯКИ ЯКИМ З'ЯВИЛИСЯ АЛГОРИТМИ ГЕНЕРАЦІЇ ПІДРОБНИХ ЗОБРАЖЕНЬ

Областю машинного навчання, що найбільш розвивається, є глибокі нейромережі. Саме завдяки розвитку цього напряму стала з'являтися велика кількість алгоритмів побудованих на глибокому навчанні. Усі основні алгоритми для генерації підробних зображень використовують для цього нейромережі.

Штучні нейронні мережі – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок людини. Такі системи навчаються виконувати різні задачі (покроково покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під ці задачі. Наприклад, у розпізнаванні зображень нейронні мережі можуть навчатися ідентифікувати зображення, які містять собак, аналізуючи приклади зображень, розподілені як «собака» і «не собака», і використовуючи результати для ідентифікації собак в інших зображеннях. Вони роблять це без додаткового знання про собак, наприклад, що вони мають хутро, хвости та вуха. Натомість, вони розвивають свій власний набір доречних характеристик з навчального матеріалу, який вони оброблюють.

## 1.1 Спосіб дії нейронних мереж

Первинною метою підходу штучних нейронних мереж було розв'язання задач тим же способом, як це робив би людський мозок. З часом увага зосередилася на відповідності певним розумовим здібностям, ведучи до відхилень від біології. Штучні мережі використовували в ряді різноманітних задач, включно з розпізнаванням мовлення, комп'ютерним

баченням, машинним перекладом, грою в настільні та відеоігри, соціально-мережевим фільтруванням, та медичним діагностуванням.

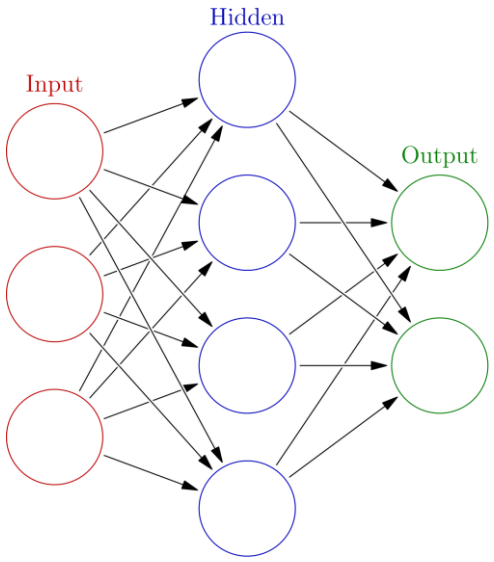


Рисунок 1.1 – Структура нейронної мережи

Нейронні мережі містять в собі нейрони, які складаються зі збудження, що залежить від дискретного параметру часу; можливого порогу, що залишається незмінним, якщо його не змінить функція навчання; функції збудження, яка обчислює нове збудження в заданий час; та функції виходу, яка обчислює вихід з активації.

Нейрон входу слугує інтерфейсом входу для всієї мережі. Аналогічно, нейрон виходу не має наступників, і, відтак, слугує інтерфейсом виходу для всієї мережі.

Мережа складається зі з'єднань, кожне з яких передає вихід нейрону $i$ до входу нейрону $j$. В цьому сенсі $i$ є попередником $j$, а $j$ є наступником $i$. Кожному з'єднанню призначено вагу $w$.

Також нейронна мережа містить в собі функцію поширення, яка обчислює вхід до нейрону $j$ з виходів нейронів-попередників, і зазвичай має вигляд:

$$p_j(\mathrm{t}) = \sum_{i=0}^{n} o_i(t) w_{ij}, \tag{1.1}$$

Правило навчання – це правило або алгоритм, який змінює параметри нейронної мережі за якимось алгоритмом, щоб заданий вхід до мережі видавав придатний вихід. Цей процес навчання зазвичай полягає в зміні ваг та порогів змінних мережі.

Нейромережеві моделі можна розглядати як прості математичні моделі, що визначають функцію $f: X \to Y$, або розподіл над $X$, або над $X$ та $Y$. Іноді моделі тісно пов'язують з певним правилом навчання. Поширене використання фрази «модель ШНМ» насправді є визначенням класу таких функцій (де складові цього класу отримують варіюванням параметрів, вагу з'єднань, або особливостей архітектури, таких як, число нейронів або їх зв'язність) [1].

З математичної точки зору, нейромережеву функцію *f(x)* визначають як композицію інших функцій $g_i(x)$, які можуть бути розкладені далі на інші функції. Це може зручно представляти як мережеву структуру, де стрілки зображують залежність між функціями. Широко застосованим способом компонування є нелінійна зважена сума, де $f(x) = \mathrm{K}(\sum_i w_i g_i(x))$ (що часто називають функцією збудження) є визначеною наперед функцією, такою як, гіперболічний тангенс, або нормована експоненційна функція, або сигмоїдна функція, або пряма функція. Важливою характеристикою функції збудження є те, що вона забезпечує плавний перехід при зміні значень входу, тобто, невелика зміна входу призводить до невеликої зміни виходу.

Функціональний підхід – це вхід $x$ перетворювано на тривимірний вектор $h$, який відтак перетворювано на двовимірний вектор $g$, який нарешті перетворювано на $f$. Цей підхід найчастіше зустрічається в контексті оптимізації.

Ймовірнісний підхід – це: випадкова змінна $F = f(G)$ залежить від випадкової змінної $G = g(H)$, яка залежить від $H = h(X)$, яка залежить від випадкової змінної $X$. Цей підхід найчастіше зустрічається в контексті моделей графів.

Ці два підходи є здебільшого рівнозначними. В кожному з випадків, для цієї конкретної архітектури, складові окремих шарів не залежать одна від одної (наприклад, складові $g$ є незалежними одна від одної за заданого їхнього входу $h$). Це природно створює міру паралелізму в реалізації. Зазвичай такі мережі називають мережами прямого поширення, оскільки їх граф є спрямованим ациклічним графом. Мережі з циклами зазвичай називають рекурентними. Такі мережі зазвичай зображують у спосіб, де $f$ показана, як залежна від самої себе. Проте, не показано часову залежність, яка мається на увазі.

Найбільше важливою ознакою нейронних мереж є їх здатність до навчання. Для заданої конкретної задачі для розв'язання та класу функцій $F$ навчання означає використання набору спостережень для знаходження $f^* \in F$, яка розв'язує цю задачу в певному оптимальному сенсі.

Це тягне за собою визначення такої функції витрат $C: F \to R$, що, для оптимального розв'язку $f^* = C(f^*) \leq C(f) \forall f \in F$ – тобто, жоден розв'язок не має витрат, менших за витрати оптимального розв'язку.

Функція витрат $C$ є важливим поняттям у навчанні, оскільки вона є мірою того, наскільки далеким є певний розв'язок від оптимального розв'язку задачі, яку потрібно розв'язати. Алгоритми навчання здійснюють пошук простором розв'язків, щоб знайти функцію, яка має найменші можливі витрати.

Для застосувань, де розв'язок залежить від даних, витрати обов'язково мають бути функцією спостережень, бо інакше модель не матиме зв'язку з даними. Їх часто визначають як статистику, для якої може бути зроблено лише наближення.

Коли $N \to \infty$, застосовує різновид інтерактивного машинного навчання, в якому витрати знижуються з кожним побаченим зразком. І, хоча інтерактивне машинне навчання часто застосовують за незмінного $D$ найкориснішим воно є у випадку, коли цей розподіл повільно змінюється з часом. В нейромережевих методах різновиди інтерактивного машинного навчання часто застосовують для великих наборів даних.

Коли можливо визначити функцію витрат спрямовану на конкретний випадок, використовують конкретні витрати (функцію витрат), або через те, що вони мають бажані властивості (наприклад, як опуклість), або через те, що вони природно виникають з певного формулювання задачі (наприклад, у ймовірнісному формулюванні як обернені витрати можна використовувати апостеріорну ймовірність моделі). Функція витрат залежить від задачі.

Глибинне навчання може бути натреновано за допомогою стандартного алгоритму зворотного поширення. Зворотне поширення – це метод обчислення градієнту функції втрат, які пов’язані з заданим станом, по відношенню до ваг в штучній нейронній мережі.

Уточнення ваг зворотного поширення можливо здійснювати за допомогою стохастичного градієнтного спуску із застосуванням наступного рівняння:

$$w_{ij}(t+1) = w_{ij}(t) + n\frac{\partial C}{\partial w_{ij}} + \varepsilon(t), \qquad (1.2)$$

де $n$ – темп навчання;

C – функція витрат;

ε(t) – стохастичний член.

Вибір функції витрат залежить від таких чинників як: тип навчання (кероване, спонтанне, з підкріпленням) та функції збудження. Наприклад, при здійсненні керованого навчання на задачі багатокласової класифікації

поширеними варіантами вибору функції збудження та функції витрат є нормована експоненційна функція та функція перехресної ентропії відповідно. Нормалізовану експоненційну функцію визначають як:

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}, \tag{1.3}$$

де $p_j$ – ймовірність класу (вихід вузла $j$);

$x_j$ та $x_k$ – загальний вхідний сигнал вузлів $j$ та $k$ одного й того ж рівня відповідно.

Перехресну ентропію визначають як:

$$C = -\sum_j d_j \log p_j, \tag{1.4}$$

де $d_j$ – цільова ймовірність для вузла виходу;

$j$, $p_j$ – вихід ймовірності для j після застосування функції збудження.

Це можливо використовувати для виведення обмежувальних меж об’єкта у вигляді двійкової маски. Їх також використовують для багатомасштабної регресії для підвищення точності визначення положення. Регресія на основі ШНМ може навчатися ознак, що охоплюють геометричну інформацію, на додачу до того, що вони слугують добрим класифікатором. Вони усувають вимогу явного моделювання частин та їх взаємозв’язків. Це допомагає розширити розмаїття об’єктів, яких можна навчитися. Модель складається з декількох шарів, кожен з яких має випрямляч як функцію збудження для нелінійного перетворення. Деякі шари є згортковими, тоді як деякі є повнозв’язними. Кожен згортковий шар має додаткове максимізаційне агрегування. Мережу тренують для зведення до мінімуму похибки L2 для передбачування маски, що пробігає весь тренувальний набір, що містить обмежувальні коробки, представлені як маски.

До альтернатив зворотному поширенню належать машини екстремального навчання, «безпоширні» мережі, тренування без пошуку з вертанням, «безвагові» мережі та не-конективістські нейронні мережі.

Існує три основні парадигми навчання, кожна з яких відповідає певній навчальній задачі. Ними є кероване навчання, спонтанне навчання та навчання з підкріпленням.

Кероване навчання використовує набір прикладів пар $(x, y), x \in X, y \in Y$, і має на меті пошук функції $f: X \to Y$ в дозволеному класі функцій, яка відповідає цим прикладам. Іншими словами, потрібно вивести відображення, на яке натякають ці дані; функцію витрат пов'язано з невідповідністю між нашим відображенням та даними, і вона неявно містить знання про предметну область.

Широко вживаними витратами є середньоквадратична похибка, яка намагається звести до мінімуму усереднену квадратичну похибку між виходом мережі, *f(x)*, та цільовим значення у над усіма прикладами пар. Зведення до мінімуму цих витрат за допомогою градієнтного спуску для класу нейронних мереж, званого багатошаровими перцептронами (БШП), дає алгоритм зворотного поширення для тренування нейронних мереж.

Задачами, що відносяться до парадигми керованого навчання, є розпізнавання образів (відоме, як класифікація) та регресія (відома, як наближення функцій). Парадигма керованого навчання є застосовною також і до послідовностей даних (наприклад, до розпізнавання писання вручну, мовлення та жестів).

## 1.2 Згорткові нейронні мережі

Аналіз зображень є однією з найбільш витратних по кількості операцій задачею. Якщо подати одне трьох канальне зображення з невеликим розміром 800 на 600 знадобилася б нейронна мережа, на вхідному рівні якої було б 144 000 нейронів. Тому була створена спеціальна

архітектура нейронних мереж для роботи із зображеннями, без якої будь-яка машинна підробка зображень була б неможливою.

Згорткові нейронні мережі – це архітектура штучних нейронних мереж, запропонована Я. Лекуном у 1988 році і націлена на ефективне з боку потужностей та точності розпізнавання зображень, входить до складу технологій глибокого навчання. Використовує деякі особливості зорової кори, в якій були відкриті так звані прості клітини, що реагують на прямі лінії під різними кутами, і складні клітини, реакція яких пов’язана з активацією певного набору простих клітин. Таким чином, ідея згорткових нейронних мереж полягає в чергуванні двох основних шарів convolution layers і subsampling layers. Структура цієї мережі односпрямована, принципово багатошарова. Для навчання використовуються стандартні методи, такі як метод зворотного поширення помилки [3].

Архітектура отримала свою назву через наявність операції згортки, суть якої в тому, що кожен фрагмент зображення множиться на матрицю (ядро) згортки поелементно, а результат підсумовується і записується в аналогічну позицію вихідного зображення.

Зазвичай, робота згорткової нейронної мережі інтерпретується як перехід від конкретних особливостей зображення до більш абстрактних, і так продовжується аж до виділення понять високого рівня. При цьому мережа самоналаштовується і виробляє необхідну ієрархію абстрактних ознак, фільтруючи незначні деталі і виділяючи істотне.

У стандартному перцептроні, який представляє собою повнозв’язну нейронну мережу, кожен нейрон пов’язаний з усіма нейронами попереднього шару, причому кожний зв’язок має свій персональний ваговий коефіцієнт. В операції згортки у згортковій нейронній мережі використовується лише обмежена матриця ваг невеликого розміру, яка «рухається» по всьому оброблюваному шару (на самому початку – безпосередньо по вхідному зображенню), формуючи після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією.

Тобто для різних нейронів вихідного шару використовуються одна і та ж матриця ваг, яку також називають ядром згортки. Цю матрицю інтерпретують, як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом. Тоді наступний шар, що вийшов в результаті операції згортки такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак. В згортковій нейронної мережі набір ваг не є одним, а є цілою гамою, що кодує елементи зображення (наприклад, лінії і дуги під різними кутами). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки. Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальною (багато незалежних карт ознак на одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір матриці), а на невелику відстань. Так, наприклад, при розмірності матриці ваг 5×5 її зрушують на один або два нейрона (пікселя) замість п’яти, щоб не «переступити» потрібну ознаку.

Розглянемо типову структуру згорткової нейронної мережі більш детально. Мережа складається з великої кількості шарів. Після початкового шару (вхідного зображення) сигнал проходить серію згорткових шарів, в яких чергується власне згортка і субдискретизація. Чергування шарів дозволяє складати «карти ознак» з карт ознак, на кожному наступному шарі карта зменшується в розмірі, але збільшується кількість каналів. На практиці це означає здатність розпізнавання складних ієрархій ознак. Зазвичай, після проходження декількох шарів, карта ознак вироджується в вектор або навіть скаляр, але таких карт ознак стають набагато більше. На виході згорткових шарів мережі додатково встановлюють кілька шарів повнозв’язної нейронної мережі, на вхід якої подаються кінцеві карти ознак.

Шар згортки – це основний блок згорткової нейронної мережі. Він включає в себе для кожного каналу свій фільтр, ядро згортки якого обробляє попередній шар за фрагментами (підсумовуючи результати матричного добутку для кожного фрагмента). Вагові коефіцієнти ядра згортки (невеликої матриці) невідомі і встановлюються в процесі навчання.

Особливістю згорткового шару є порівняно невелика кількість параметрів, яка встановлюється при навчанні. Так, наприклад, якщо вихідне зображення має розмірність 100×100 пікселів по трьом каналам (це значить 30000 вхідних нейронів), а згортковий шар використовує фільтри с ядром 3×3 пікселя з виходом на 6 каналів, тоді в процесі навчання визначається тільки 9 ваг ядра, однак за всіма сполученнями каналів, тобто 9×3×6=162, в такому випадку даний шар вимагає знаходження тільки 162 параметрів, що істотно менше кількості шуканих параметрів повнозв’язної нейронної мережі. Двовимірна згорткова нейронна мережа показана на рис. 1.2.

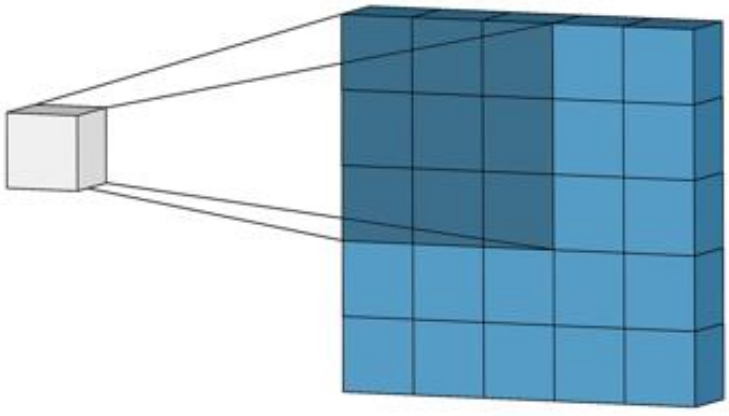

Рисунок 1.2 – Двовимірна згортка

Двовимірна згортка – це досить проста операція: починається з ядра, яке являє собою матрицю ваг [4]. Це ядро «ковзає» над двовимірним зображенням, поелементно виконуючи операцію множення з тією

частиною вхідних даних, над якою воно зараз знаходиться, і потім підсумовує всі отримані значення в один вихідний піксель.

Ядро повторює цю процедуру з кожною локацією, над якою воно «ковзає», перетворюючи двовимірну матрицю в іншу все ще двовимірну матрицю ознак. Ознаки на виході є зваженими сумами (де ваги є значеннями самого ядра) ознак на вході, розташованих приблизно в тому ж місці, що і вихідний піксель на вхідному шарі. Операція згортки продемонстрована на рис. 1.3.

Незалежно від того, чи потрапляє вхідна ознака в «приблизно те ж місце», він визначається в залежності від того, знаходиться він в зоні ядра, що створює вихідні дані, чи ні. Це означає, що розмір ядра згорткової нейронної мережі визначає кількість ознак, які будуть об’єднані для отримання нової ознаки на виході.

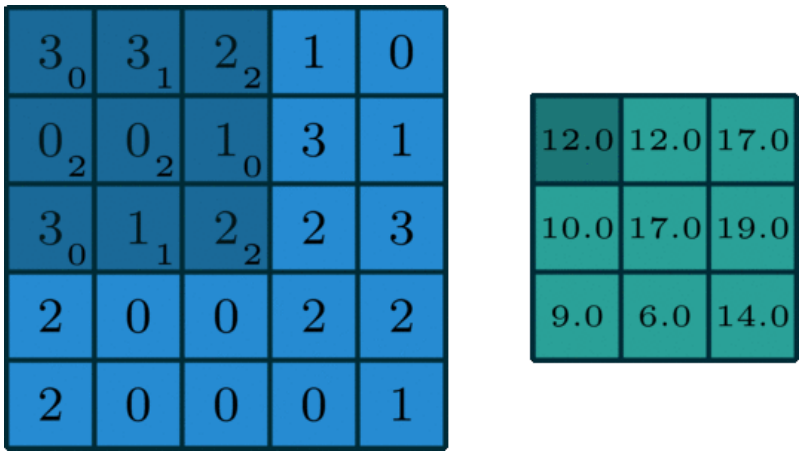


Рисунок 1.3 – Операція згортки

У прикладі, наведеному вище, ми маємо 5×5=25 ознак на вході і 3×3=9 ознак на виході. Для стандартного шару (standard fully connected layer) ми б мали вагову матрицю 25×9 =225 параметрів, а кожна вихідна ознака була би зваженою сумою всіх ознак на вході. Згортка дозволяє зробити таку операцію з усього 9-ма параметрами, адже кожна ознака на

виході отримується аналізом не кожної ознаки на вході, а тільки однією вхідною, що знаходиться в «приблизно в тому ж місці».

Після кількох проходжень згортки зображення і ущільнення за допомогою пулінгу система перебудовується від конкретної сітки пікселів з високою роздільною здатністю до більш абстрактних карт ознак, як правило, на кожному наступному шарі збільшується число каналів і зменшується розмірність зображення в кожному каналі. Зрештою залишається великий набір каналів, що зберігають невелику кількість даних (навіть один параметр), які інтерпретуються як абстрактні поняття, що виявлені з вихідного зображення.

Ці дані об’єднуються і передаються на звичайну повнозв’язну нейронну мережу, яка теж може складатися з декількох шарів. При цьому повнозв’язні шари вже втрачають просторову структуру пікселів і мають порівняно невелику розмірність (по відношенню до кількості пікселів вихідного зображення) [5].

На практиці більшість вхідних зображень мають 3 канали, і чим глибше ви в мережі, тим більше це число. Потрібно думати о каналах, в загальному, як про «погляди» на зображення в цілому, надаючи більшого значення одним аспектам і менше іншим.

У більшості випадків ми маємо справу з зображеннями RGB з трьома каналами. Приклад багатоканального зображення продемонстрований на рис. 1.4.

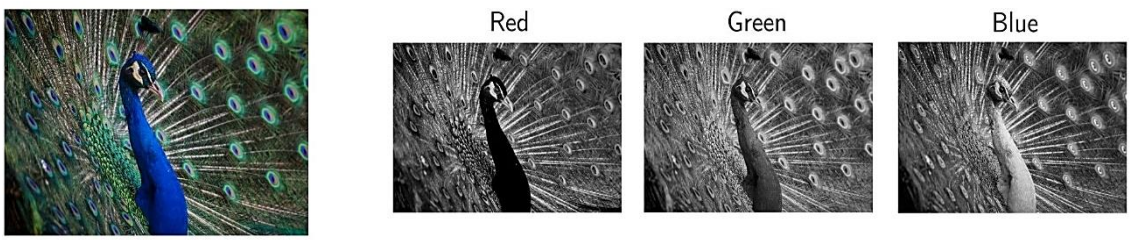


Рисунок 1.4 – Багатоканальне зображення

Ключові відмінності між термінами стають потрібні, коли у випадку з одним каналом, де терміни «фільтр» і «ядро» взаємозамінні, в загальному випадку вони різні. Кожен фільтр насправді являє собою колекцію ядер, причому для кожного окремого вхідного каналу для цього шару є одне ядро, де кожне ядро унікальне. Кожен фільтр в згортковому шарі створює тільки один вихідний канал і роблять вони це так: кожне з ядер фільтра «ковзає» по їх відповідним вхідним каналам, створюючи оброблену версію кожного з них. Приклад продемонстровано на рис. 1.5.

Деякі ядра можуть мати більшу вагу, ніж інші, для того щоб приділяти більше уваги певним вхідним каналам (наприклад, фільтр може мати червоний канал ядра з більшою вагою, ніж інші канали, і, тим самим, більше реагувати на відмінності в образах з червоного каналу).

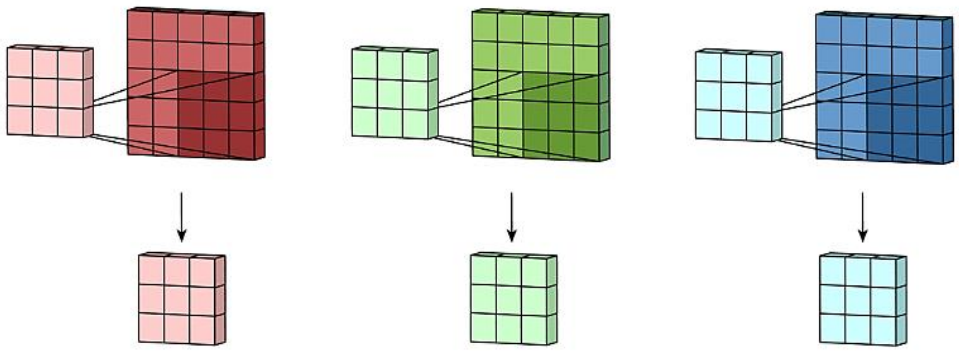

Рисунок 1.5 – Операція згортки на триканальному зображенні

Наступним кроком кожна з оброблених в каналі версій підсумовується разом для формування одного каналу. Ядра кожного фільтра генерують одну версію кожного каналу, а фільтр в цілому створює один загальний вихідний канал. Приклад продемонстровано на рис. 1.6. Принцип роботи зміщення полягає в тому, що кожен вихідний файл має

свій термін зміщення. Зсув додається до вихідного каналу для створення кінцевого вихідного каналу.

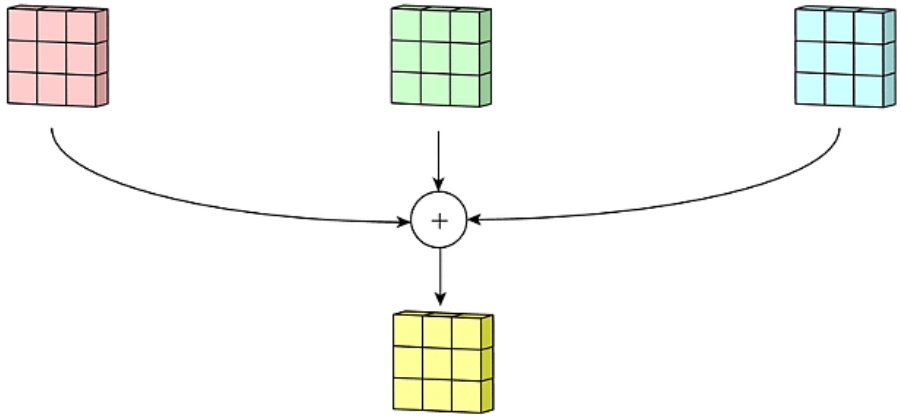

Рисунок 1.6 – Об’єднання операцій по всім каналам

Результат для будь-якої кількості фільтрів ідентичний: кожен фільтр обробляє вхід зі своїм відмінним від інших набором ядер і скалярним зміщенням за описаним вище процесом, створюючи один вихідний канал. Потім вони об’єднуються разом для отримання загального виходу, причому кількість вихідних каналів дорівнює числу фільтрів. При цьому зазвичай застосовується нелінійність перед передачею входу іншого шару згортки, який потім повторює цей процес.

Навіть з описаною механікою роботи згорткового шару, все ще складно пов’язати це з нейронною мережею прямого поширення, і це все ще не пояснює, чому згортки масштабуються та працюють набагато краще з зображеннями. Припустимо, що у нас є вхід 4×4, і ми хочемо перетворити його в сітку 2×2. Якби ми використовували feed-forward network, ми б переробили вхід 4×4 в вектор довжиною 16 і передали його через щільно пов’язаний шар з 16 входами і 4 виходами. Можна було б візуалізувати вагову матрицю *W* для шару по зображеному на рис. 1.7 [6].

І хоча згорткові операції з ядрами можуть спочатку здатися трохи дивними, це як і раніше лінійні перетворення з еквівалентною матрицею переходу.

$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & w_{1,5} & w_{1,6} & w_{1,7} & w_{1,8} & w_{1,9} & w_{1,10} & w_{1,11} & w_{1,12} & w_{1,13} & w_{1,14} & w_{1,15} & w_{1,16} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & w_{2,5} & w_{2,6} & w_{2,7} & w_{2,8} & w_{2,9} & w_{2,10} & w_{2,11} & w_{2,12} & w_{2,13} & w_{2,14} & w_{2,15} & w_{2,16} \\ w_{3,1} & w_{3,2} & w_{3,3} & w_{3,4} & w_{3,5} & w_{3,6} & w_{3,7} & w_{3,8} & w_{3,9} & w_{3,10} & w_{3,11} & w_{3,12} & w_{3,13} & w_{3,14} & w_{3,15} & w_{3,16} \\ w_{4,1} & w_{4,2} & w_{4,3} & w_{4,4} & w_{4,5} & w_{4,6} & w_{4,7} & w_{4,8} & w_{4,9} & w_{4,10} & w_{4,11} & w_{4,12} & w_{4,13} & w_{4,14} & w_{4,15} & w_{4,16} \end{bmatrix}$$

Рисунок 1.7 – Вагова матриця

Якби ми використовували ядро *K* розміру 3 на видозміненому входом розміру 4×4, щоб отримати вихід 2×2, еквівалентна матриця показана на рис. 1.8. В той час, як наведена матриця є еквівалентною матрицею переходу, фактична операція зазвичай реалізується, як інше матричне множення.

$$\begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{1,1} & k_{1,2} & k_{1,3} & 0 & k_{2,1} & k_{2,2} & k_{2,3} & 0 & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix}$$

Рисунок 1.8 – Матриця переходу

Згортка, в цілому, все ще є лінійним перетворенням, але в той же час вона також представляє собою зовсім інший вид перетворення. Для матриці з 64 елементами існує всього 9 параметрів, які повторно використовуються кілька разів. Кожен вихідний вузол отримує тільки певну кількість входів (ті, що знаходяться всередині ядра). Немає ніякої взаємодії з іншими входами, так як вага для них дорівнює 0.

Можна представити згорткові операції, як більш пріоритетні для вагових матриць. В даному контексті, під пріоритетом мається на увазі зумовлені мережеві параметри. Наприклад, коли використовується попередньо оброблена модель для класифікації зображень, також використовуються попередні параметри мережі, такі, як пріоритет образів для остаточного щільно пов'язаного шару. Обидва вони ефективні, в порівнянні з їх альтернативами.

Transfer learning є ефективнішим в порівнянні з випадковою ініціалізацією, тому що потрібно лише оптимізувати параметри кінцевого повністю пов'язаного шару, а це означає, що можна мати кращі результати всього лише з декількома десятками зображень в класі.

## 1.3 Autoencoder

Autoencoder – нейронна мережа, яка копіює вхідні дані на вихід. По архітектурі схожий на персептрон. Autoencoder стискає вхідні дані для представлення їх в latent-space (прихований простір), а потім відновлюють з цього представлення output (вихідні дані).

Мета – отримати на вихідному шарі відгук, найбільш близький до вхідного [7]. Відмінна особливість Autoencoder – кількість нейронів на вході і на виході співпадає. Autoencoder складається з двох частин: Encoder: відповідає за стискування входу в latent-space. Представлений функцією кодування $h=f(x)$. Dencoder: призначений для відновлення введення з latent-space. Представлений функцією декодування $h=f(x)$. Таким чином, автокодер описують, як $g(f(x))=r$, де $r$ співпадає з первинним $x$ на вході. Приклад Autoencoder показано на рис. 1.9.

Завдяки цій технології стало можливим подавати одне зображення на вхід нейронної мережі, змінювати його і отримувати на виході зображення тих же розмірів але вже з внесеними в нього змінами.

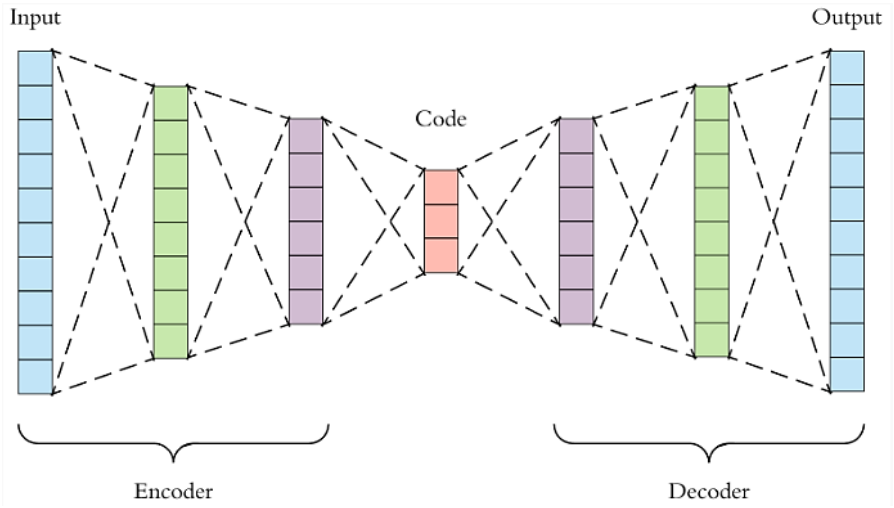


Рисунок 1.9 – Приклад роботи Autoencoder

Якби єдиним завданням Autoencoder було копіювання вхідних даних на вихід, вони були б даремні. Дослідники розраховують на те, що приховане представлення $h$ матиме корисні властивості. Цього домагаються створенням обмежень для завдання копіювання. Один із способів отримати корисні результати від Autoencoder − обмежити $h$ розмірами меншими, ніж $x$. Де $D$ в цьому випадку автокодування називається неповним. Архітектура Autoencoder може бути як прості нейронні мережі, так і глибокі рекурентні, LSTM наприклад [8], або згорткові. Все залежить від вирішуваної задачі.

## 1.4 Постановка задач дослідження

Визначення актуальності теми дозволяє сформулювати досить конкретну мету роботи, яка обумовлена спробою дослідити ситуацію розвитку існуючих технологій виявлення DeepFake та на їх основі створити узагальнену модель розпізнавання DeepFake, яка буде давати більш точний результат в роботі із складними підробними зображеннями та відео. Таким

чином, у цій роботі представлені описи методів створення, а також виявлення DeepFake.

Об’єктом цього дослідження є процес створення моделі для розпізнавання DeepFake.

Предметом дослідження є математичні, програмні засоби і методи, завдяки яким буде можливе досягнення мети дослідження.

В основу виконаної роботи покладена наступна гіпотеза: застосовуючи найбільш сучасну архітектуру згортальних мереж, а також Attention механізми, можливо створити узагальнену модель, що показує достатні результати на різнорідних даних.

Виходячи з поставленої мети, сформульовані наступні завдання:

− аналіз блоків і принципів роботи DeepFake та подібних алгоритмів;

− опис існуючих методів розпізнавання DeepFake та аналіз їх основних переваг і недоліків;

− сворення нового алгоритму на основі останніх тенденцій розвитку згортальних нейронних мереж;

− пошук відповідного набору даних для перевірки роботи моделі;

− аналіз отриманих результатів і рекомендації щодо їх покращення.

# 2 ОПИС ТЕХНОЛОГІЇ DEEPFAKE

## 2.1 Генеративно-змагальні нейронні мережі

Генеративно-змагальні мережі (GAN) розвиваються дуже швидко для вирішення завдань навчання як з учителем, так і для без нього. Цей метод був запропонований в 2014 році Я. Гудфеллоу з Університету Монреаля [9]. Основна ідея – це навчання пари мереж в постійному змаганні один з одним. Популярною аналогією може послужити конкуренція між фальшивомонетником і банкіром. Шахрай, який виступає генератором (*G*) в GAN, хоче створити підробні купюри, які не можна буде відрізнити від справжніх. Банкір, у свою чергу, що є дискримінатором (*D*), намагається розібрати підробки і справжні гроші один від одного. Протягом декількох років цей метод знаходить своє застосування в завданнях семантичної сегментації зображень, аналізу медичної інформації, розпізнаванні матеріалів та аналізі часових рядів.

Важливо, що генератор ніколи не отримує реальні дані, на вхід подається тільки випадковий вектор (джерело ентропії, іноді інтерпретується як простір прихованих змінних, latent-space). Єдиний спосіб навчання для нього– це взаємодія з дискримінатором. Дискримінатор же отримує на вхід або створені генератором дані, або об’єкт реальної навчальної вибірки.

Помилка навчання дискримінатора розраховується на знанні того, звідки прийшли дані. В процесі навчання (рис. 2.1) генератор навчається розподілу початкової вибірки і починає створювати дані усе більш близькі до реальних в той час, як дискримінатор стає усе більш точним в розпізнаванні підробки від оригіналу.

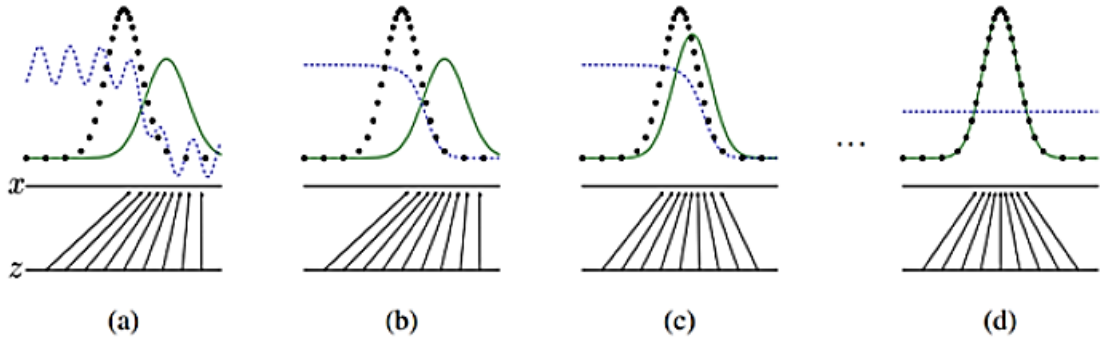


Рисунок 2.1 – Процес навчання GAN

GAN мережі навчаються одночасно, оновлюючи розподіл дискримінатора (*D*, синя переривчаста лінія), який показує ймовірність приналежності розподілу генератора (*G*, зелена лінія) до розподілу реальних даних (чорна точкова лінія). Нижня горизонтальна лінія – це область визначення, з якої вибирається *z* (в цьому випадку рівноймовірно). Горизонтальна лінія вища – область визначення для *x*. Стрілки показують зіставлення *x=G(z)*. У останній ітерації (*d*) дискримінатор не може відрізнити реальні дані від згенерованих.

Мережі, які лежать в основі генератора і дискримінатора, зазвичай є багатошаровими мережами, що складаються із згортальних і повнозв'язних шарів (рис. 2.2). *D* і *G* повинні відрізнятися, так що не обов'язково, щоб вони були повністю дзеркальними. Оскільки завдання генератора полягає в зіставленні простору прихованих змінних в простір даних, то можна це записати у вигляді:

$$G: G(z) \to R^{|x|}, \tag{2.1}$$

де z $\in R^{|x|}$ – вибірка з простору прихованих змінних;

x $\in R^{|x|}$ – справжні дані.

У разі дискримінатора потрібна функція зіставлення даних у ймовірність їх приналежності до істинних:

$$D: D(x) \to (0,1), \tag{2.3}$$

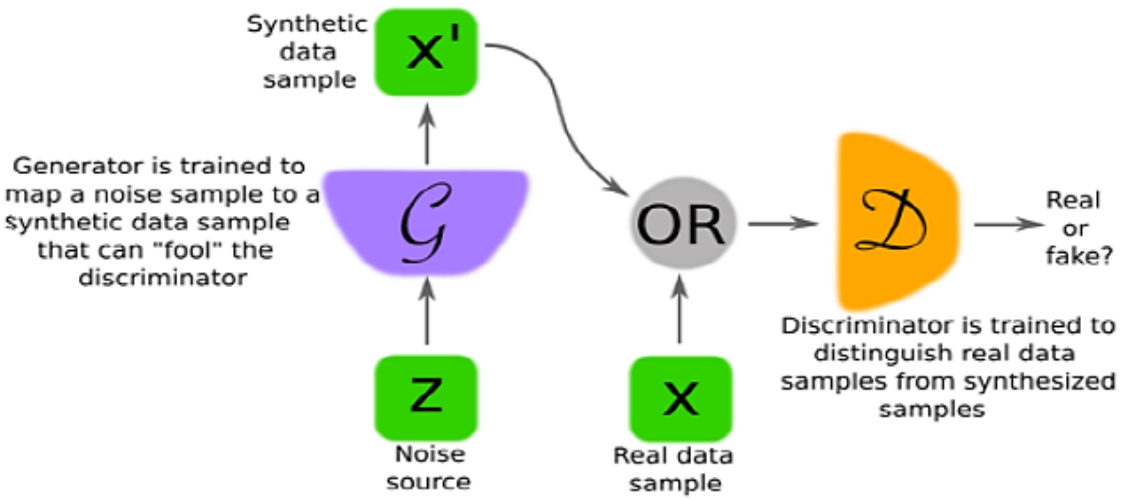


Рисунок 2.2 – Схема GAN

На рис. 2.2: *X* – реальний об’єкт предметної області;

*Z* − джерело випадкового шуму;

*G* − нейронна мережа генератора;

*X`* − штучно згенеровані дані, мімікруючі реальні;

*OR* − випадковий вибір одного з двох входів;

*D* − нейронна мережа дискримінатора.

GAN не перша генеративна модель, проте Гудфеллоу із співавторами запропонували оригінальний метод змагального навчання таких моделей, коли дві мережі «змагаються» в рішенні протилежних завдань.

Дослідники виділяють декілька варіацій первинної ідеї генеративно-змагальних мереж:

− повнозв’язні GAN − спочатку запропонована архітектура, де генератором і дискримінатором служать багатошарові мережі прямого поширення;

− згорткові GAN (deep convolutional GAN, DCGAN) − використовують багатошарові згорткові мережі [10]. Є логічним розвитком ідеї GAN – застосування до завдань синтезу зображень. Недоліком є довший процес навчання моделі;

− умовні GAN (Conditional GAN, CGAN) − архітектура, в якій як генератору, так і дискримінатору на вхід додатково подається вектор, що вказує на клас об’єкту (рис. 2.3). Такі мережі можуть генерувати умовний розподіл вибірки, з вказівкою конкретного класу. Таким чином, вони використовуються для моделювання мультимодальних розподілів;

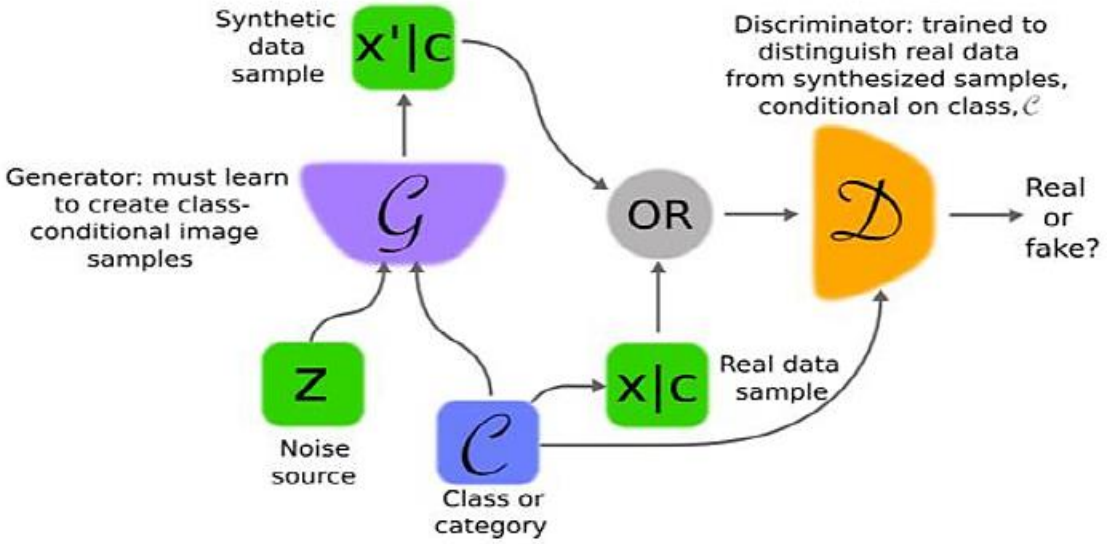


Рисунок 2.3 – Схема CGAN

– змагальні Autoencoders (Adversarial Autoencoders, AAE) – це мережі, що складаються з двох частин − Encoder і Decoder, які навчаються детерміністичному відображенню з простору даних в простір прихованих змінних (зазвичай набагато меншій розмірності) і зворотному

відображенню [11]. Змагальне навчання використовується для оптимізації, схожої з варіаційними автокодувальниками і зазвичай використовується для надання простору прихованих змінних осмисленої організації в термінах предметної області;

– GAN з моделями виведення (ALI, BiGAN) – розширюють функціональність GAN механізмом виведення прихованих змінних з об’єкту предметної області. Звичайні GAN можуть згенерувати правдоподібний об’єкт з випадкового вектору в просторі прихованих змінних (latent-space) [12]. Було запропоновано дві архітектури змагального навчання: однонаправлена (Adversarial Learning Inference) і двонаправлена GAN (Bidirectional GAN).

Пошук нових сфер застосування генеративно-змагальних мереж є в даний момент дуже активною областю дослідження. Вони вже добре себе зарекомендували в таких областях, як класифікація зображень, генерація зображень по текстовому опису, перетворення зображень, підвищення роздільної здатності. В наш час генеративно-змагальні мережі залишаються методикою машинного навчання, що активно розвивається.

Проте, за час свого існування, генеративно-змагальні мережі виявили істотні проблеми у використанні, такі як колапс моделі (ситуація, при якій генератор видає один об’єкт при будь-яких значеннях входів), нестабільність навчання, відсутність загальноприйнятих підходів до оцінки ефективності.

Штучні нейронні мережі довели свою ефективність в рішенні багатьох завдань, проте ця ефективність великою мірою залежить від уміння підбирати правильну архітектуру мережі під кожне конкретне завдання аналізу даних.

При проектуванні архітектури мережі експерт стикається з необхідністю прийняти безліч рішень, як кількісних, так і якісних, від реалізації яких безпосередньо залежить продуктивність результуючої моделі. Серед них такі як: вибір кількості шарів, кількості нейронів в

кожному шарі, вибір функції активації, використання рекурентних і згортальних шарів. В процесі такого проектування завданням експерта є балансування між варіативною моделлю і схильністю до перенавчання. Існує ще і зовнішні чинники, які необхідно брати в розрахунок: обчислювальна потужність, що використовується, часові рамки рішення задачі тощо.

Простір можливих нейронних мереж (рис. 2.4), в якому ведеться пошук, різноманітний. При проєктуванні нейронних мереж фахівці користуються певними евристичними правилами і інструментами діагностики, проте повноцінною методологією такий пошук назвати складно, це швидше творчий процес.

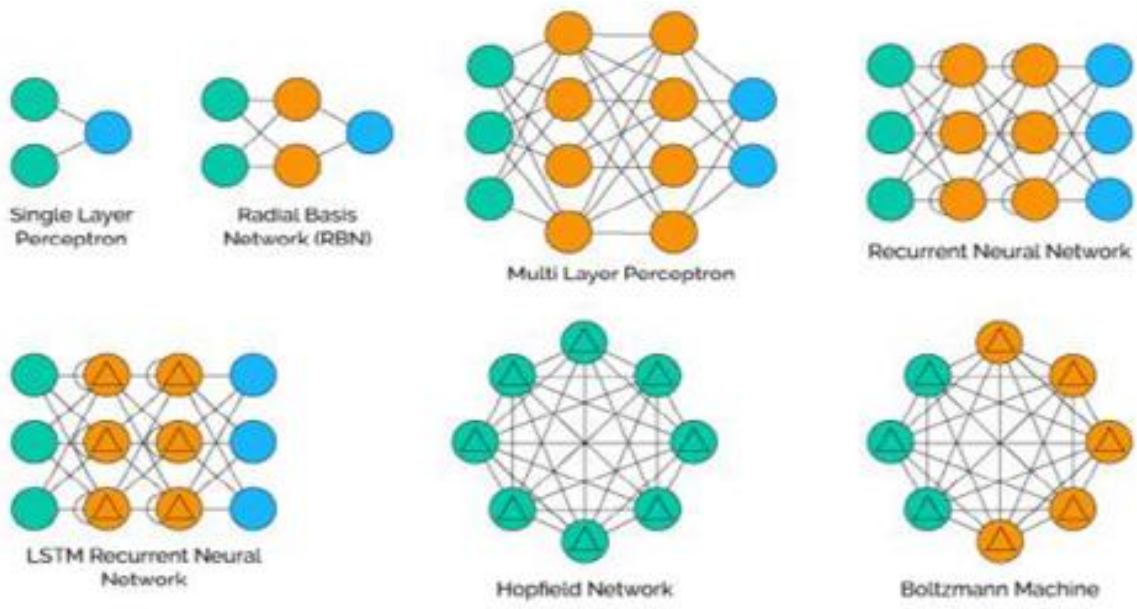


Рисунок 2.4 – Приклади поширених архітектур нейронних мереж

## 2.2 Технологія DeepFake

У грудні 2017 року, користувач під псевдонімом «DeepFakes» опублікував на форумі Reddit велику кількість дуже реалістичних відео для дорослих, де за допомогою глибинного навчання, застосував алгоритм,

який дозволяє підставляти обличчя людей замість інших. Перше DeepFake відео демонструвало обличчя актора з порноіндустрії, яке замінили на обличчя іншої людини.

У подальші декілька тижнів, інтернет-суспільство обговорювало потенційну небезпеку нової технології. Заміна чужого обличчя на зображеннях або відео було можливо і до цього, проте для реалістичності підміни були потрібні фахівці і години роботи. Завдяки технології DeepFake будь-хто з потужною відеокартою і зібраними даними з інтернету, може створити подібні відео за хвилини.

Ці форми фальсифікації створюють загрозу порушенню конфіденційності й ідентичності та впливають на багато аспектів життя людини. Стає можливим зловживання DeepFake для того, щоб викликати політичну чи релігійну напругу між країнами, обдурити громадськість, вплинути на результати виборчих кампаній, створення хаосу на фінансових ринках та, створювання підроблених новин. DeepFake можна використовувати створюючи підроблені супутникові знімки Землі, які містять об’єкти, які насправді не існують на зображенні для того, щоб заплутати військових аналітиків.

На сьогоднішній день застосовувати цей алгоритм дуже просто, адже для цього потрібна невелика кількість фотографій чи короткий відеоролик. Все менше і менше ресурсів потрібно для створення переконливих DeepFake знімків. Останнім досягненням у цій галузі стає можливість створити DeepFake не лише із нерухомого зображення. Наприклад, голосовий DeepFake був використаний для того, щоб заплутати генерального директора і, як результат, шахраї викрали $243000, а недавній випуск програмного забезпечення під назвою DeepNude показує більш тривожні загрози, оскільки він може перетворити людину на порноактора.

Існує також й позитивне використання DeepFake, таке як створення голосів тих, хто втратив свій або оновлення епізодів фільмів без їх повторної зйомки. Однак кількість шкідливих програм DeepFake значною

мірою перевищує позитивні. Розвиток прогресивних глибоких мереж та наявність великої кількісті даних зробила підроблені зображення та відео майже однаковими для людини і навіть до складних комп'ютерних алгоритмів.

Принцип роботи Autoencoder складає алгоритм, що лежить в основі DeepFake, глибокої нейронної мережі, яка навчається тому, що отримавши зображення на вході, стискає його до маленьких розмірів, після чого декодує його назад, до початкового формату [13]. Принцип роботи Autoencoder можна побачити на рис. 2.5.

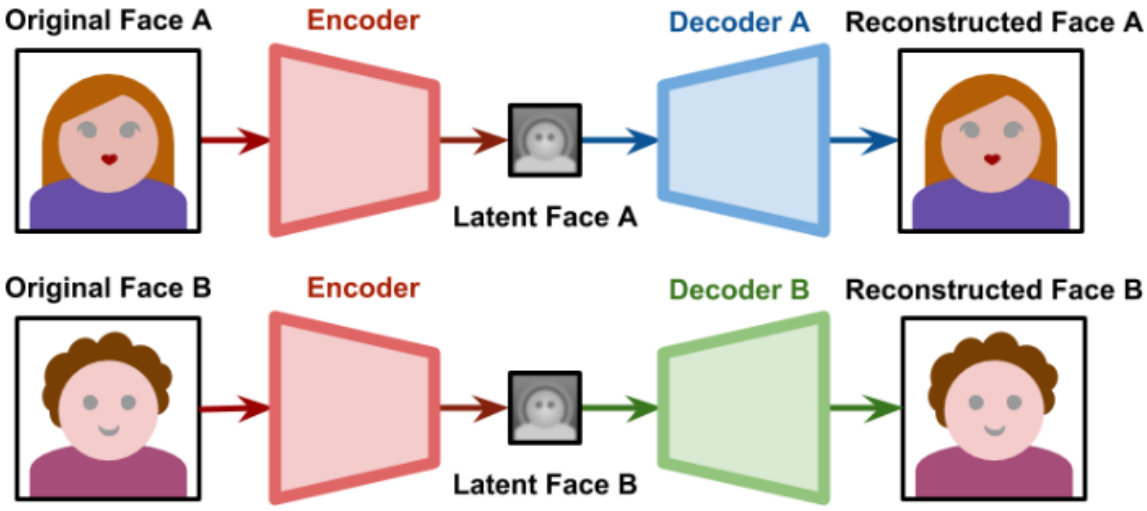


Рисунок 2.5 – Принцип роботи Autoencoder

Якщо навчити декілька різних Autoencoder, вони будуть не сумісні друг з другом. Базовий вектор кожного з них складатиметься з ознак, які кожна з мереж визнала важливими. Але якщо навчити два однакових Autoencoder, на різних обличчях то base-vector-и складатимуться з однакового набору ознак з різними значеннями.

Технологія DeepFake використовує два Autoencoder з однаковими Encoder, але різними Decoder частинами, що дозволяє мати однаково побудований базовий вектор для різних зображень, що і робить можливим

підстанову осіб надалі. Упродовж фази навчання, дві мережі відпрацьовують окремо. Decoder A навчається тільки на обличчях A, Decoder B навчається тільки на обличчях B. Проте усі базові вектори, отримані за допомогою однієї і тієї ж архітектури Encoder. Це означає, що Encoder визначає загальні ознаки в обох особах.

Коли процес навчання закінчений, можливо узяти базовий вектор суб'єкта A і передати в Decoder суб'єкта B. Decoder B спробує відтворити суб'єкт B за інформацією отриманої від суб'єкта A. Приклад генерації DeepFake показаний на рис. 2.6.

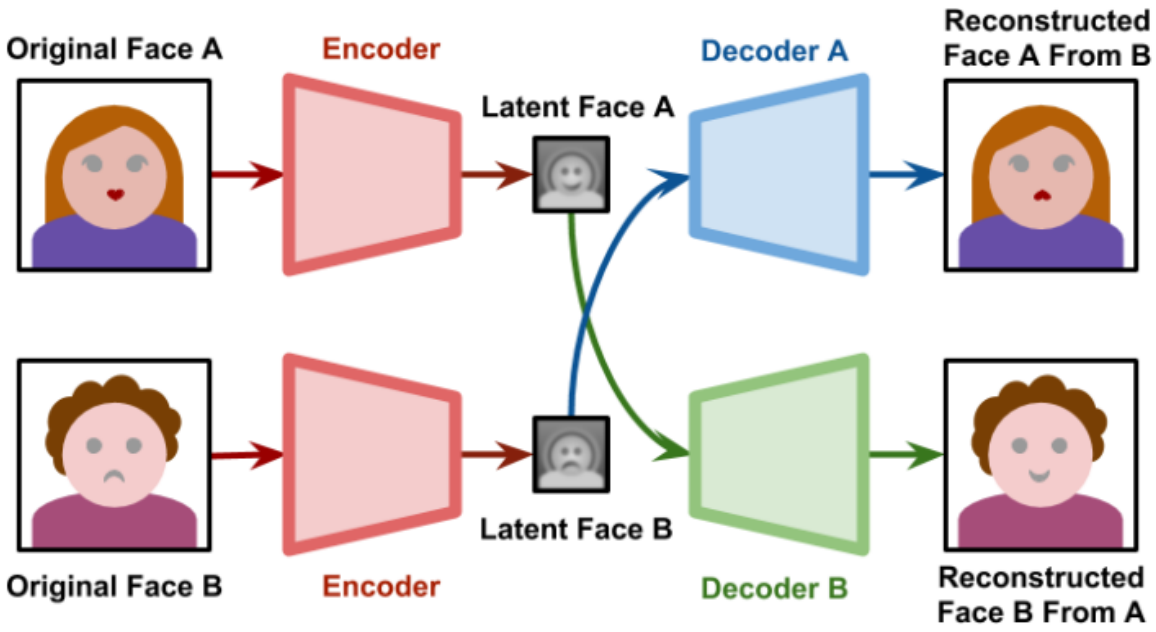


Рисунок 2.6 – Генерація DeepFake

Якщо мережа досить добре узагальнює те, що робить або передає особу, базовий вектор буде складатися з виразів обличчя, їх розташуванням та нахилом.

Хоча і даний підхід здається дуже дієвим, проте він має велику кількість обмежень. Він застосовується тільки за наявності великої кількості зображень людини, на чиє обличчя ми прагнемо замінити оригінал. Для отримання видимих результатів, треба від 200 до 2000

зображень. Кількість зображень залежить від того, наскільки незвичайні обличчя. Так само зображення мають бути з різних ракурсів і містити якомога більший спектр емоцій.

Навчання таких моделей дуже витратно: щоб навчити модель замінювати обличчя людини A на обличчя людини B, знадобиться близько 72 годин навчання на останніх моделях відеокарт, а це приблизно 45 доларів.

Щоб згенерувати однохвилинне відео і замінити на ньому обличчя, треба близько 18 хвилин на одній відеокарті. Архітектуру DeepFake, можна оптимізувати багатьма способами, проте найбільш поширеними з них є комбінування моделей з GAN архітектурою.

Генератори вчаться створювати найбільш реалістичну картинку, дискримінатори − визначати, яка з них згенерована, а яка оригінальна. У міру того як генератори навчаються обманювати дискримінатор, зображення виходить усе більш реалістичним.

Таким чином, Encoder і Decoder відповідають за перенесення зображення, а дискримінатор від генеративних мереж – за покращення результату. За цією логікою працює архітектура Face Swap GAN, створена японським розробником Shaoanlu [14]. Цей підхід продемонстрований на рис. 2.7.

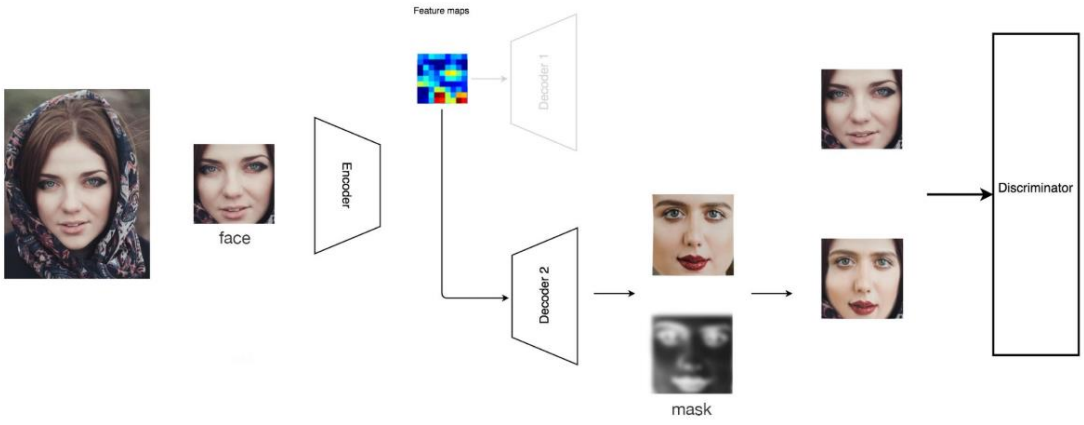


Рисунок 2.7 – Застосування GAN для покращення результатів DeepFake

Ще один підхід – архітектура з використанням декількох генеративно-змагальних мереж. Кожна мережа відповідає за свою операцію, що зводить до мінімуму кількість етапів із застосуванням ручної праці.

Щоб навчити таку нейромережу, вимагається декілька діб і потужний кластер відеокарт. Незважаючи на це, такий підхід є найбільш перспективним, тому що дає кращий результат. Одне з рішень, створених за цією технологією, – FSGAN, яке було опубліковано ізраїльським дослідником Ю. Ниркиним.

Додавши до даного підходу рекурентні мережі можна ще більше покращити якість DeepFake. Рекурентні мережі дозволяють отримувати базовий вектор ґрунтуючись не лише на одному зображенні, а на декількох зображеннях. Базовий вектор генерується на поточному зображенні, а також зображеннях з відео створених за декілька секунд до і після початкового зображення. Цей підхід придатний, якщо є відео людини. Таким чином, переходи з одного стану і положення обличчя в інше, стають набагато реалістичніше. Так Туляков у своїй роботі MoCoGAN, продемонстрував найновітніший підхід в генерації DeepFake відео [15]. Не так давно, в роботі «First Order Motion Model for Image Animation», автори представили модель, яка тільки по одному зображенню, здатна генерувати дуже реалістичні DeepFake відео.

DeepFake технологію можна використовувати не лише для генерування зображень, але і голосу. При такому підході, часто беруть справжнє відео і змінюють в ньому слова і рух губ. Технологія синхрон. губ з’явилася в університеті Вашингтону. Використовуючи рекурентну мережу LSTM, аудіо запис *x* трансформується в послідовність з 18 опорних точок у на губах. Враховуючи витягнуту текстуру губ і рота, алгоритм генерує нове положення і підставляє його в оригінальне відео. Цей підхід продемонстрований на рис. 2.8.Алгоритм проходить по усьому відео і шукає найбільш схоже положення губ із згенерованим варіантом.

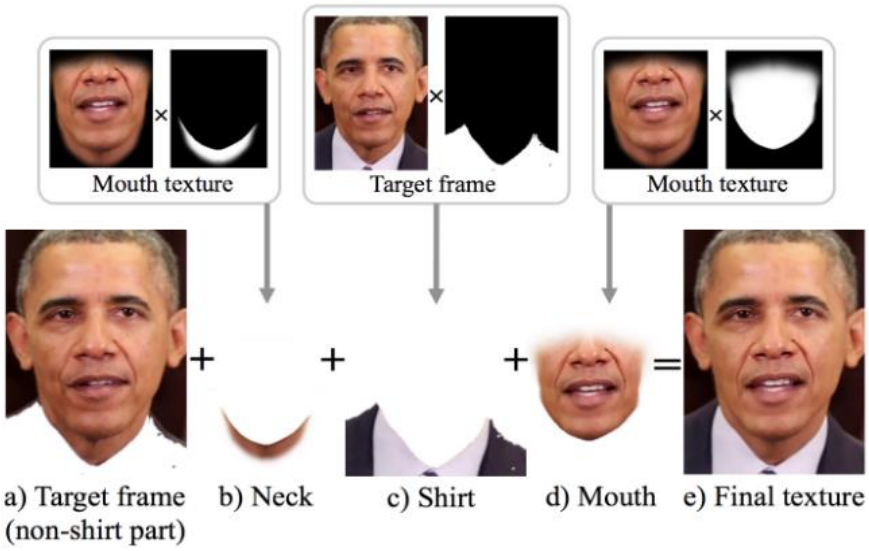


Рисунок 2.8 – Принцип роботи «lip-sync» алгоритму

Далі знайдений фрагмент поєднується з згенерованим використовуючи медіанну функцію. Наступним кроком, відбувається згладжування зображення, щоб прибрати розмиті частини і зробити його реалістичнішим.

## 2.3 Модель FSGAN

FSGAN розшифровується як Face Swapping GAN. Ця модель являється найновітнішою для генерації підроблених зображень і відео. Головна відмінність цього підходу полягає в тому, що він може застосовуватися до осіб без необхідності попереднього навчання на них [14]. У моделі FSGAN одна нейромережа вчиться підганяти обличчя донора під параметри цільового відео (поворот голови, нахил убік або вперед), друга переносить риси обличчя, а третя робить image blending (злиття зображень), щоб картинка була реалістичнішою, без ушкоджень або артефактів. Особливістю цього алгоритму, являється наявність рекурентної нейронної мережі для реконструкції осіб, яка коригує як позу, так і різні вирази обличчя.

Ця мережа може бути використана на зображеннях і відео. Для відео використовується підхід безперервної інтерполяції представлень обличчя на основі реконструкції, тріангуляція Делоне і барицентричні координати. Закриті ділянки обличчя обробляються окремою мережею. У кінці використовується мережа для об'єднання двох осіб, яка зберігає цільовий колір шкіри і особливості освітлення. На рис. 2.9. продемонстрований описаний підхід.

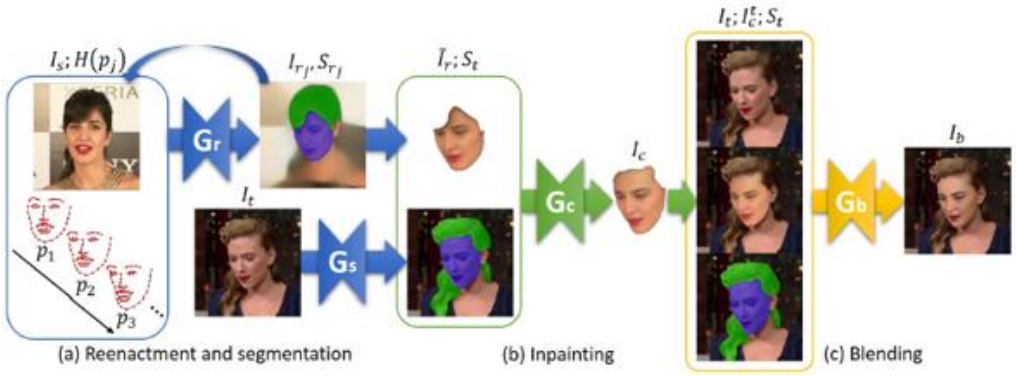


Рисунок 2.9 – Принцип роботи FSGAN

Нехай $I_s$ це початкове зображення обличчя $F_s$, а $I_t$ підсумкове обличчя $F_t$. Алгоритм створює нове зображення, використовуючи $I_t$, намагаючись замінити $F_t$ на $F_s$, обличчя зберегши його положення і вираз. FSGAN складається з трьох компонентів. Перший це генератор інсценування $G_r$ і згортальна мережа для сегментації ділянок обличчя $G_s$. $G_r$ витягаючи опорні точки обличчя $F_t$ генерує зображення $I_r$, таке, що $F_r$ відображає $F_s$ з таким же положенням і виразом як у $F_t$. Він обчислює маску сегментації $S_r$ обличчя $F_r$. Компонент $G_s$ сегментує особу і волосся обличчя $F_t$. На виділеному зображенні $I_r$ можуть бути відсутніми деякі частини обличчя, як продемонстровано на рис. 2.9, етап b. Далі застосовується inpainting мережа $G_c$, використовуючи сегментацію $S_t$, вона виявляє упущені пікселі. Остання частина моделі об'єднує обличчя $F_c$ з початковим зображенням $I_t$ для отримання кінцевого результату.

# 3 МЕТОДИ ВИЯВЛЕННЯ ПІДРОБЛЕНИХ ЗОБРАЖЕНЬ

## 3.1 Рекурентні нейронні мережі для виявлення підроблених зображень

Для виявлення підроблених зображень використовується згортальна рекурентна мережа LSTM. Ця модель складається з двох основних блоків: згортальна мережа для витягання елементів зображення і LSTM для аналізу частини послідовності зображень [16]. Архітектура моделі представлена на рис. 3.1.

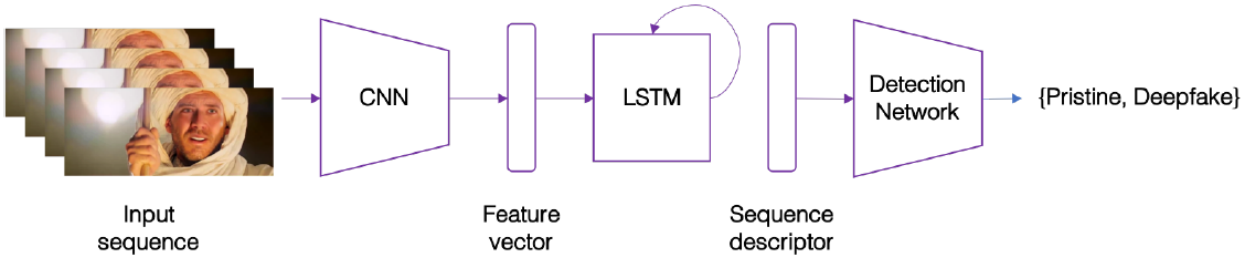


Рисунок 3.1 – Архітектура рекурентної моделі

Отримуючи послідовність тестових зображень, модель витягає набір функцій для кожного кадру за допомогою згортальної мережі. Після цього, функції декількох послідовних кадрів об’єднуються і передаються LSTM для аналізу. Рекурентна мережа видає ймовірність для послідовності, правдива вона або ні.

Для згортальної мережі була узята архітектура InceptionV3 з прибраним fully-connected шаром у кінці архітектури.

LSTM для обробки послідовностей. Як вхідні дані використовується послідовність вектором ознак витягнутих згортальною мережею з поданих на вхід кадрів, і згортальна мережа з двома вузлами для визначення вірогідності чи є відео частиною згенерованого відео або ж оригінального.

Для рекурсивної обробки послідовності зображень використовується LSTM розмірністю у 2048, з Dropout шаром з ймовірністю у 0,5. У кінці мережі використовується softmax функція для визначення ймовірності підробності послідовності. LSTM мережа є проміжним блоком системи і не вимагає додаткового навчання.

Для демонстрації точності моделі було узято близько 300 відео, згенерованих за допомогою DeepFake. Автори моделі перевіряли точність, використовуючи 20, 40 і 80 кадрів підряд з відео. Результати продемонстровані на рис. 3.2.

| **Model** | **Training acc. (%)** | **Validation acc. (%)** | **Test acc. (%)** |
|---|---|---|---|
| Conv-LSTM, 20 frames | 99.5 | 96.9 | 96.7 |
| Conv-LSTM, 40 frames | 99.3 | 97.1 | 97.1 |
| Conv-LSTM, 80 frames | 99.7 | 97.2 | 97.1 |

Рисунок 3.2 – Точність рекурентної моделі

Як видно модель демонструє дуже непогані результати пророцтва. Проте у цієї моделі є безліч недоліків. По-перше, вона може працювати тільки з послідовностями підроблених відео. По-друге, експерименти проводилися до появи моделі FSGAN, яка здатна генерувати підробки набагато більшої складності. Так само ця модель погано працює з відео, на яких обличчя було змінено частково.

### 3.2 Виявлення підробних зображень за біологічними сигналами

При використання цього підходу, було зроблено спостереження, що хоча GAN-подібні моделі здатні генерувати фотореалістичні візуальні і

геометричні сигнали, які виходять за рамки можливостей виявити їх людським оком, біологічні сигнали, приховані природою, як і раніше складно відтворити. Біологічні сигнали є невід'ємною частиною відео із зображенням осіб, яку у свою чергу є сферою застосування DeepFake технології. Крім того, відео, на відміну від зображень, містять ще один рівень складності, для створення підробок: узгодженість в часі. Часова узгодженість є ключем для виявлення підробного контенту.

Модель обробляє вхідне відео, збираючи сегменти відео з різними частинами обличчя, визначаючи декілька областей інтересів на кожному обличчі. Далі з областей інтересів витягається декілька біологічних сигналів. Модель формулює рішення, досліджуючи витягнуті сигнали, перетворюючи їх різними способами (вимір тривалості, частоти і тривалості і частоти разом) і спостерігаючи кореляцію між ними. Цей підхід називається витягання парного контексту. Далі відбувається об'єднання спостережень парного контексту із спостереженнями з літератури, щоб розробити узагальнений класифікатор достовірності, працюючий з багатовимірним простором ознак [17].

Наступним кроком, витягнуті сигнали перетворяться в нові сигнальні карти спостережуваних сегментів, для того, щоб натренувати простий згортальний класифікатор DeepFake. Принцип роботи моделі продемонстрований на рис. 3.3.

Модель витягає шість сигналів, які є комбінацією G каналів PPG (стійкі до стискування артефакти) і каналів на основі колірного тону (стійкі до освітлення артефакти). Ці артефакти витягаються з правої і лівої щоки, а також з центральної ділянки обличчя. Кожному сигналу присвоюється перша буква каналу, за яким йде ділянка обличчя в якості нижнього індексу. Як результат, шість витягнутих сигналів називаються відповідно: $G_l, G_r, G_m, C_l, C_r, C_m$.

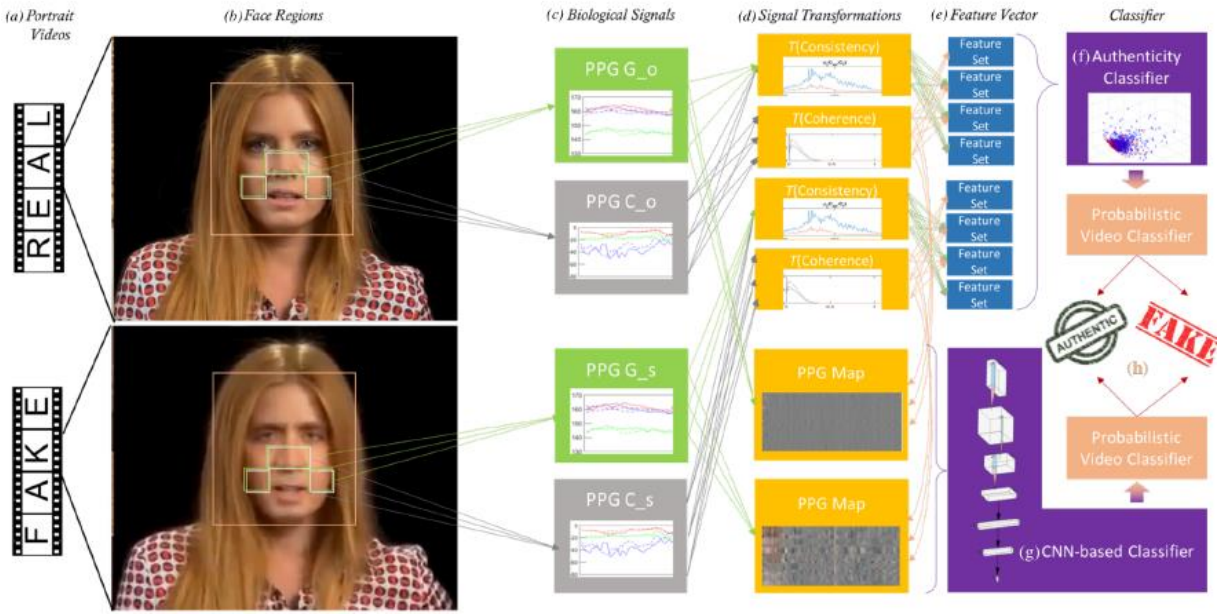


Рисунок 3.3 – Принцип роботи класифікатора на основі біологічних сигналів

Ґрунтуючись на витягнутих сигналах з оригінального і підробного відео, виділяється великий список ознак цих сигналів. Далі виділені ознаки подаються на вхід SVM класифікатора. Цей класифікатор маркує відео з точністю близько 80%. Основною проблемою цього класифікатора є час його навчання, а також точність. Класифікатор передбачає велику кількість неправдиво позитивних прикладів.

Для покращення результатів, до SVM класифікаторів в ансамбль була додана згортальна нейронна мережа. На нейронну мережу подавались різні канали різних областей інтересу обличчя. Результати різних архітектур показані на рис. 3.4. В якості потенційних покращень моделі, автори пропонували різні набори областей інтересу обличчя, які в середньому дають такі ж результати, проте результати можуть значно відрізнятися для різних категорій відео.

| Model | Frame | Face | Video |
|---|---|---|---|
| Simple CNN | 46.89% | 54.56% | 48.88% |
| InceptionV3 | 73.85% | 60.96% | 68.88% |
| Xception | 78.67% | 56.11% | 75.55% |
| ConvLSTM | 47.65% | 44.82% | 48.83% |
| [66] V1 | **86.26%** | - | 82.22% |
| [66] V3 | 76.97% | - | 73.33% |
| [66] ensemble | 83.18% | - | 80.00% |
| Ours | - | **87.62%** | **91.07%** |

Рисунок 3.4 – Результати роботи згортальних класифікаторів

Основними проблемами даної моделі є її складність і значний час навчання. Проте ця модель є універсальною для усіх типів осіб, що виділяє її на тлі інших.

## 3.3 MesoNet компактна нейромережа для виявлення підроблених відео

MesoNet – це архітектура, яка дозволяє ефективно розпізнавати підробні відео згенеровані за допомогою алгоритмів DeepFake або Face2Face. Дослідження показали, що зробити одну узагальнену модель для ефективного виявлення підробок, створених цими алгоритмами одночасно, за допомогою цієї архітектури неможливо. Проте, завдяки схожості в роботі фальсифікуючих алгоритмів, дозволяє застосувати ідентичні архітектури нейронних мереж для вирішення обох проблем [18].

Автори архітектури пропонували виявляти підробні відео, розмістивши нейронну мережу на мезоскопічному рівні аналізу. Дійсно, мікроскопічний аналіз на основі шуму зображення не може бути застосований в контексті стислого відео, де дуже бага зайвого шуму. Так само на більш високому семантичному рівні людське око намагається розрізнити підробні зображенн, особливо коли на зображеннях знаходиться

людське обличчя. Ось чому автори запропонували використати проміжний підхід, використовуючи глибоку нейронну мережу з невеликою кількістю шарів.

Meso-4 розпочинається з послідовності з чотирьох блоків звертки і подальшого за нею пулинга з подальшим dense шаром з одним прихованим шаром. Для покращення узагальненості моделі, згорткові шари використовують активаційну функцію ReLU і Batch Normalization, щоб упорядкувати їх вихід і запобігти ефекту розмитого градієнту. Так само для регулярізації і надійності використовується Dropout шар. У результаті модель має 27 977 параметрів. Архітектури продемонстровані на рис. 3.5.

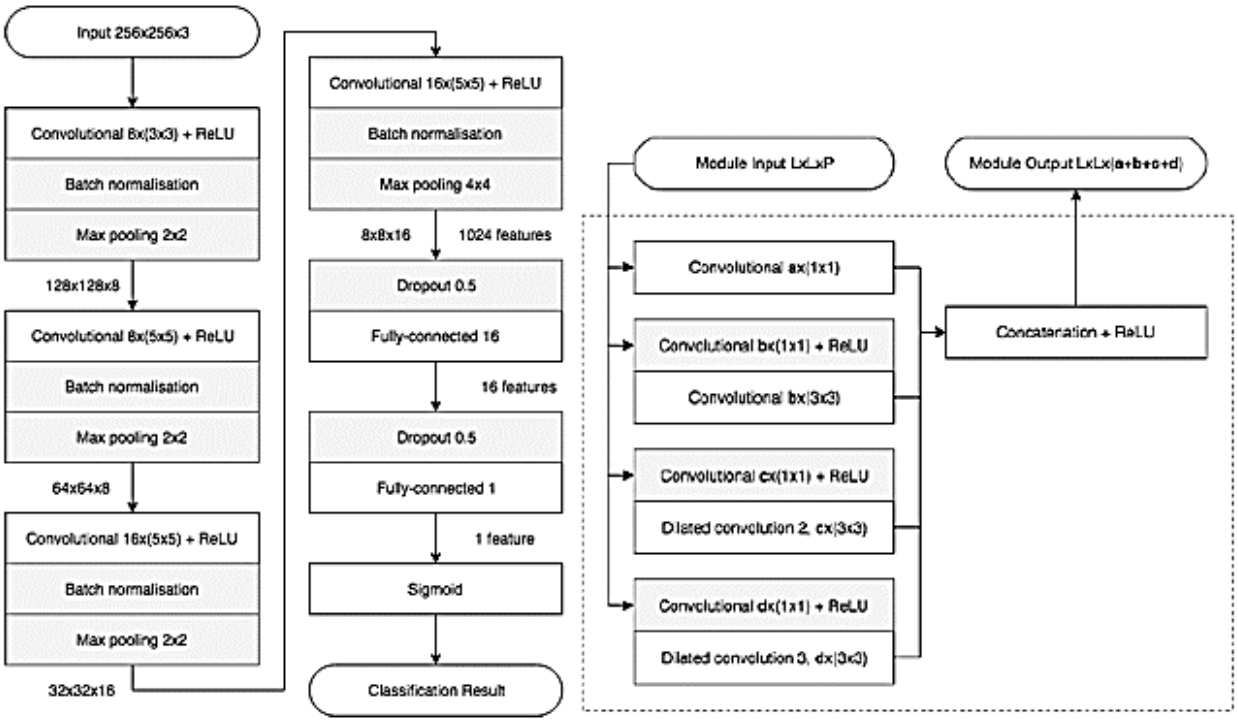


Рисунок 3.5 – Архітектури Meso-4 та MesoInception4

Альтернативна модель MesoInception4 відрізняється тим, що замість перших двох згортальних блоків, використовується Inception модуль. Ідея цього модуля полягає в об’єднанні декількох згортальних шарів різної розмірності для збільшення розмірності простору в якому модель оптимізується.

Під час проведення експериментів на 300 Face2Face відео і на 300 DeepFake відео, були отримані наступні дуже непогані результати, які продемонстровані на рис. 3.6.

| Network | Aggregation score | |
|---|---|---|
| Dataset | Deepfake | Face2Face (23) |
| Meso-4 | **0.969** | **0.953** |
| MesoInception-4 | **0.984** | **0.953** |

Рисунок 3.6 – Результати роботи MesoNet

Основною перевагою цієї моделі є її компактність. У моделі відносно мало параметрів, завдяки чому навчання відбувається швидко. Ця модель показує непогані результати тільки на відео високого розрішення, при невеликій роздільній здатності точність сильно падає. Так само модель була створена більше року тому, тому результатів її роботи з новішими і складнішими різновидами алгоритмів фальсифікації немає.

## 3.4 Виявлення артефактів деформації обличчя

Ядром моделі являється виявлення артефактів, що виникають при деформації обличчя під час генерації відео алгоритмом DeepFake. Поточні реалізації DeepFake, генерують зображення фіксованого розміру. Зображення піддаються афінному перетворенню (масштабуванню, обертанню і зрушенню), щоб відповідати позиціям цільових граней, які вони надалі замінять. У результаті область обличчя і області, що оточують її, в початковому зображенні міститимуть артефакти. Невідповідності роздільної здатності виникнуть, із-за таких перетворень на етапі стискування. Для виявлення наявності описаних артефактів,

використовується згортальна нейронна мережа. Приклад деформації обличчя показаний на рис. 3.7 [19].

Для експериментів, було узято 24442 зображення обличчя в якості оригінальних зображень. Для збору фальсифікованих прикладів, можна було б згенерувати підробки на основі обраних зображень. Проте цей процес дуже довгий і ресурсовитратний, тому автори емулювали один з кроків алгоритму DeepFake, а саме афінні перетворення. Ділянка обличчя була виділена за допомогою пакету dlib. Автори застосували різні види аугумації зображень для покращення узагальненості моделей.

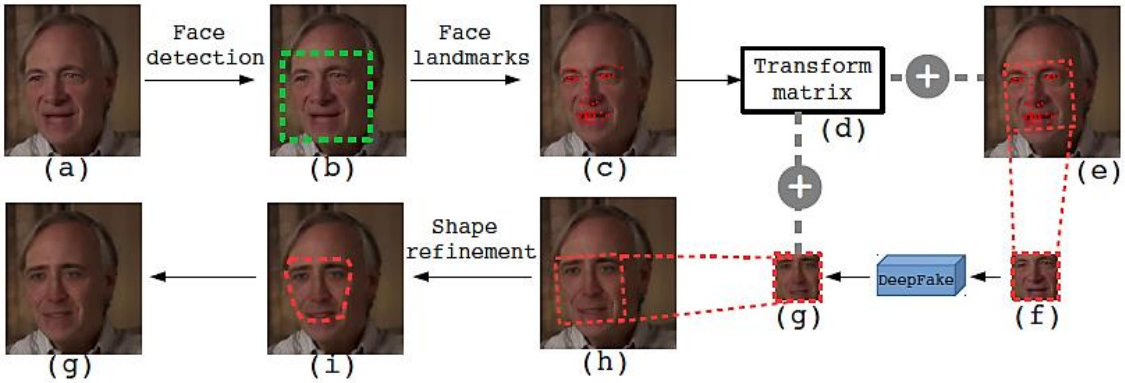


Рисунок 3.7 – Приклад виявлення артефактів деформації обличчя

Автори пробували навчати нейронні мережі різної архітектури: VGG16, ResNet50, ResNet 101 і ResNet 152. Для усіх моделей був застосований SGD оптимізатор. Валідація моделей проходила на двох DeepFake наборах даних: UADFV і DeepFakeTIMIT. Після підбору параметрів для моделей, були отримані результати, продемонстровані на рис. 3.8.

| Methods | UADFV | DeepfakeTIMIT | |
|---|---|---|---|
| | | LQ | HQ |
| Two-stream NN [35] | 85.1 | 83.5 | 73.5 |
| Meso-4 [1] | 84.3 | 87.8 | 68.4 |
| MesoInception-4 | 82.1 | 80.4 | 62.7 |
| HeadPose [34] | 89.0 | - | - |
| Ours-VGG16 | 84.5 | 84.6 | 57.4 |
| Ours-ResNet50 | **97.4** | **99.9** | **93.2** |
| Ours-ResNet101 | 95.4 | 97.6 | 86.9 |
| Ours-ResNet152 | 93.8 | 99.4 | 91.2 |

Рисунок 3.8 – Результати роботи моделей, ґрунтованих на деформаціях обличчя

ResNet50 показала відмінні результати, проте проблемою даних моделей є те, що гіперпараметри підбиралися, ґрунтуючись тільки на двох наборах даних, що могло привести до перенавчання. Так само ці набори цих відносно невеликих розмірів через що можуть містити меншу кількість нетипових випадків. Проте цей підхід виявився найефективнішим серед описаних в розділі, при цьому його складність полягає у формуванні навчальної вибірки.

## 3.5 Виявлення зображень згенерованих за допомогою DeepFake з використанням підходу парного навчання

Для відповідності вимогам до GAN подібним генераторам виявлення підробних зображень, пропонується використати модифіковану структуру згортальної мережі, у тому числі метод парного навчання, що називається мережею із загальними підробними функціями (CFFN). Використовуючи парне навчання, запропонована структура долає недоліки контрольованих CNN на основі навчання. Щоб перевірити ефективність запропонованого методу, пропонований детектор глибокої підробки (DeepFD) застосовується

для ідентифікації підробного обличчя і початкових зображень. Основний вклад цієї роботи полягає в наступному:

– пропонується класифікатор на основі нового підходу CFFN, що складається з вдосконаленого DenseNet backbone і архітектури Siamese;

– міжрівневі особливості досліджуються пропонованим CFFN, який можна використати для підвищення продуктивності;

– підхід парного навчання використовується для поліпшення узагальненості моделі.

Запропонований двоетапний метод навчання, який поєднує в собі CFF на основі стратегії парного навчання і навчання класифікатора, представлений на рис. 3.9.

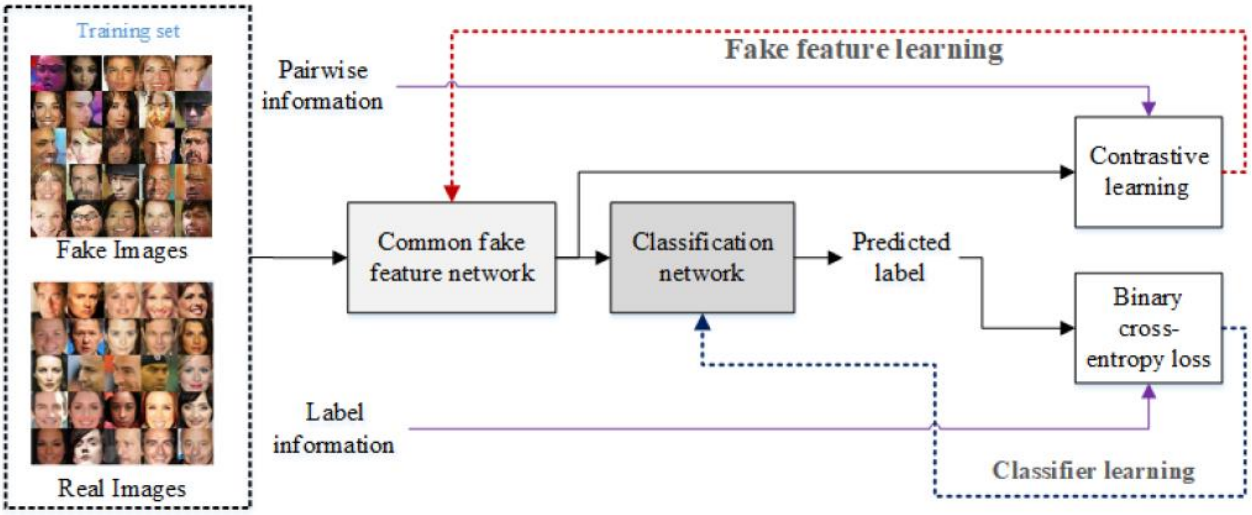


Рисунок 3.9 – Метод на основі CFF і стратегії парного навчання

Представлення стратегії контрольованого навчання у виявленні підробних зображень обличчя пов'язані з проблемами пошуку навчальної вибірки, яка б містила зображення згенеровані різними різновидами GAN. Щоб здолати ці проблеми, підробні і реальні зображення об'єднуються в пару і слідують одне за одним, використовуючи парну інформацію для побудови порівняльної функції втрат, щоб упізнати відмітну загальну

неправдиву особливість (CFF) за пропонованим CFFN. Як тільки CFF, що розпізнає, вивчений, класифікатор перехоплює дискримінаційний CFF, щоб ідентифікувати, чи є зображення реальним або підробним.

Багато просунутих CNN можуть бути використані для виявлення фальшивих ознак з тренувального набору. Просунуті CNN, такі як DenseNet, ResNet і Xception, також можуть бути використані для створення класифікатора. Проте, більшість з цих просунутих CNN навчаються контрольованим чином, тому ефективність класифікації залежить від системи навчання. Замість того, щоб вивчати підробні признаками усіх зображень, GAN шукає різні CFF. Таким чином, для вивчення CFF потрібна відповідна основа. Проте традиційні CNN (наприклад, DenseNet) не призначені для навчання дискримінаційного CFF. Щоб здолати цей недолік, пропонується інтегрувати Siamese мережу з DenseNet розвиваючи CFFN для досягнення дискримінаційного навчання CFF.

Крім того, добре відомо, що CNN фіксують представлення ієрархічної ознаки від низького рівня до високого рівня. Іншими словами, CNN використовують тільки в наданні ознак високого рівня, щоб ідентифікувати, чи є зображення підробним, чи ні. Проте CFF підробних зображень особи можуть існувати не лише в представленні високого рівня, але також і в представленні об'єктів середнього рівня. Тому міжрівневі елементи інтегровані в класифікаційний шар для підвищення продуктивності розпізнавання підробних зображень. Архітектура моделі представлена на рис. 3.10.

Запропонована CFFN містить три основні блоки, які у свою чергу, містять 2, 3 і 4 глибоких шарів відповідно, а кількість каналів в трьох блоках складає 48,60, 78 і 126. У підсумку мережа має 383 нейрони.

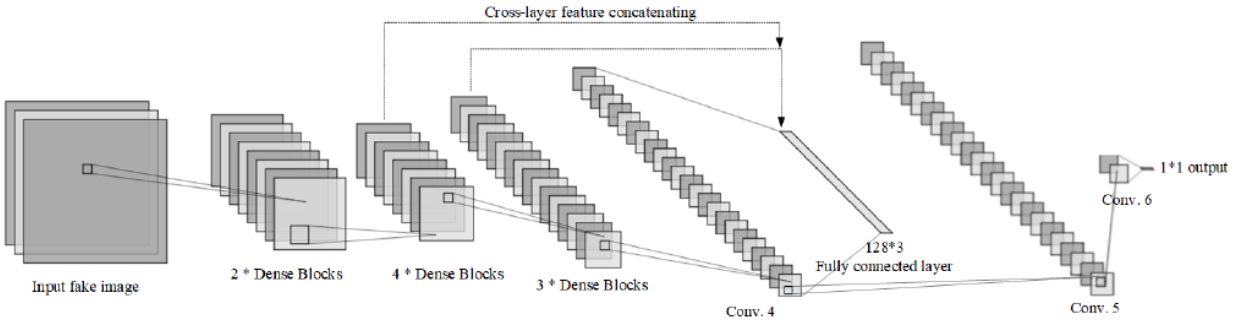


Рисунок 3.10 – Архітектура моделі на основі DenseNet

Для навчання моделі були узяті дві функції втрат, включаючи контрастну втрату і крос-ентропійну втрату для запропонованого CFFN і навчання класифікатора в пропонованому методі, відповідно. Спільна політика навчання може бути адаптована для оптимізації пропонованої мережі CFFN і класифікатора на основі двох функцій втрат одночасно. Проте важко визначити вагові значення для двох функцій. Оскільки основною метою пропонованого CFFN є навчання дискримінаційних ознак, це дозволяє навчати CFF шляхом мінімізації контрастної втрат в першу чергу. Після цього будь-який класифікатор можна використовувати для розпізнавання підробного зображення обличчя на основі навченого CFF. У цьому дослідженні використовується невелика нейронна мережа в якості класифікатора, що забезпечує можливість наскрізного навчання. Крім того, добре відомо, що класифікатор може бути легко навчений на основі вже наявного представлення ознак (feature vector). Отже, CFFN спочатку навчається на основі контрастних втрат, а потім оптимізується мережа класифікаторів шляхом мінімізації втрат від перехресної ентропії.

Основним недоліком контрольованого навчання є те, що важко визначити предмет, який виключений з етапу навчання. Щоб підвищити продуктивність пропонованого методу, вводяться контрастні втрати при вивченні CFF шляхом парного навчання. Кінцева архітектура продемонстрована на рис. 3.11.

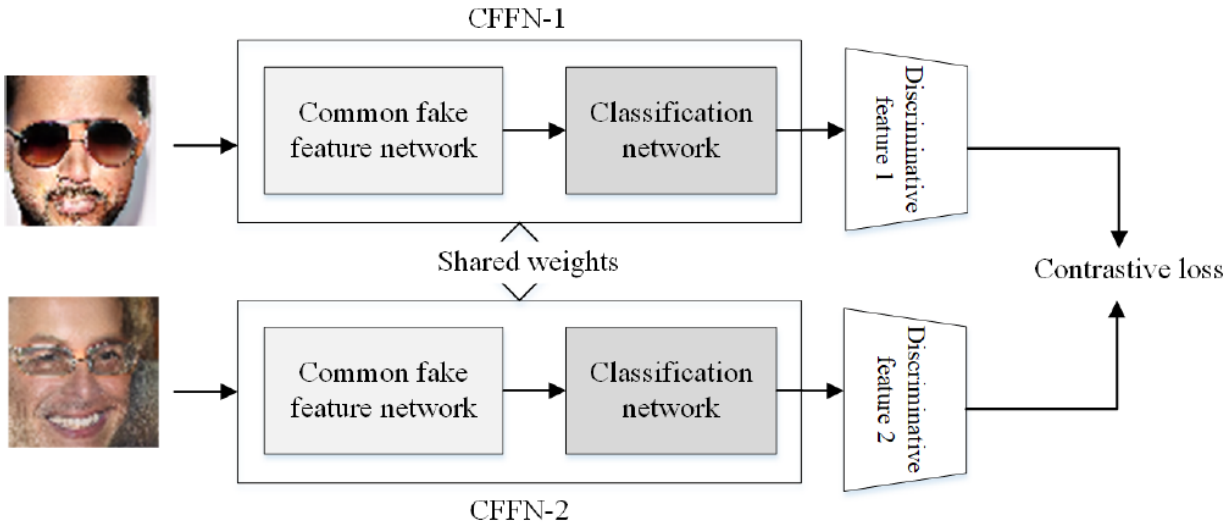


Рисунок 3.11 – Кінцева Siamese архітектура

Основною проблемою цього методу є те, що ця модель здатна класифікувати підробки, які згенеровані за допомогою базових алгоритмів DeepFake. Зображення створенні за допомогою просунутих модифікацій, таких як FSGAN, запропонована архітектура в основному класифікуватиме як початкові.

Головною перевагою цього підходу є використання сіамської архітектури мереж, яка дозволяє значно покращити результати звичайних моделей.

Проведений аналіз показав, що існуючі методи класифікації підробник зображень можуть показувати непогані результати на невеликих наборах даних. Проте внаслідок того, що описані моделі мають невеликі розміри нейро мереж то на великих наборах данних результати сильно погіршуються. Саме тому новий класифікатор створюється не на основі однієї з описаних моделей.

# 4 ПРАКТИЧНІ ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ ПІДРОБЛЕНИХ ЗОБРАЖЕНЬ

## 4.1 Виявлення підроблених зображень за допомогою ансамблю згортальних мереж

Пропонований метод ґрунтований на концепції ансамблю. Загальновідомо, що ансамбль моделей може привести до значного покращення точності різних класифікаторів на основі згортальних мереж для виявлення різних високорівневих семантичних особливостей, які доповнюють один одного, таким чином позитивно впливаючи на точність ансамблю для цієї конкретної проблеми.

Для цього в якості відправної точки було обрано EfficientNet сімейство моделей, так як саме вони є новітніми у багатьох задачах при роботі із зображеннями [20]. Цей набір архітектури досягає більшої точності та ефективності в порівнянні з іншими сучасними CNNs. Приклад цієї архітектури продемонстрований на рис. 4.1.

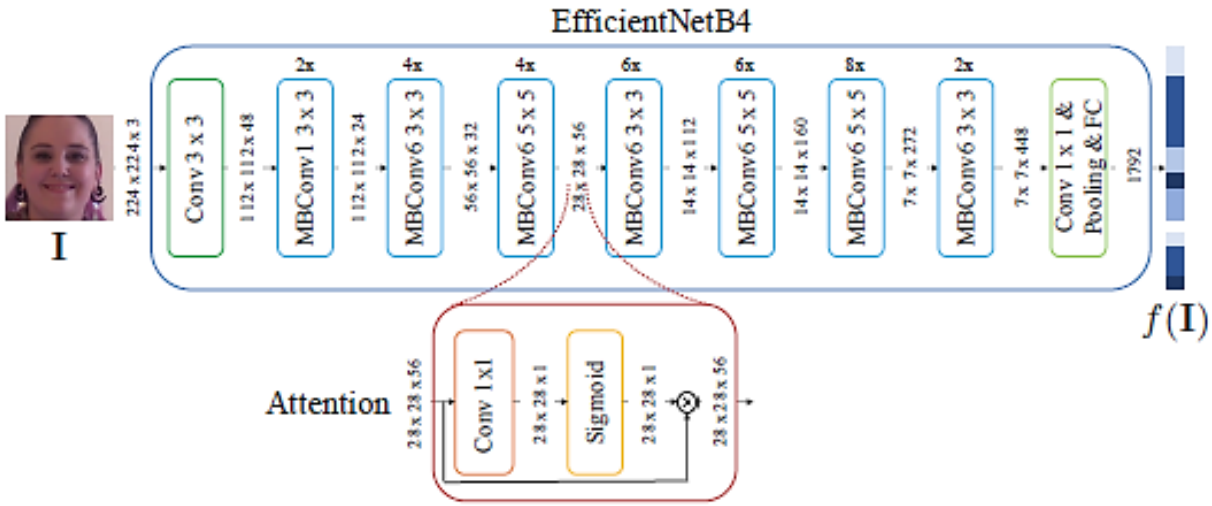


Рисунок 4.1 – Архітектура EfficientNetB4

Враховуючи особливості архітектури EfficientNet, пропонується наслідувати два шляхи, щоб зробити модель вигідною для ансамблювання. З одного боку, пропонується додати Attention механізм, який надає аналітичний метод, що дозволяє визначити, яка частина досліджуваного відео є більш інформативною для процесу класифікації. З іншого боку, досліджується, як сіамські стратегії навчання можуть бути додані в процес навчання для екстраполяції додаткової інформації про дані.

Далі наведений детальний опис архітектури EfficientNet з пропонованим Attention механізмом і стратегіями навчання мережі.

Серед сімейства моделей EfficientNet була вибрана EfficientNetB4 в якості базового рівня для цієї моделі. Цей вибір мотивований гарним компромісом, пропонованим цією архітектурою, з точки зору вимірів (тобто кількості параметрів), часу виконання (тобто вартості FLOPS) і класифікацією продуктивності. З 19 мільйонами параметрів і 4,2 мільярдами FLOPS, EfficientNetB4 досягає 83,8% точності, що є кращим результатом, на наборі даних ImageNet. У тому ж наборі даних XceptionNet, використовуваний авторами в якості базового методу виявлення маніпуляцій з обличчям, досягає 79% точність за рахунок 23 мільйонів параметрів і 8,4 мільярдів FLOPS.

Архітектура EfficientNetB4 представлена в синьому блоці на рис. 4.1, де усі рівні визначені з використанням тієї ж номенклатури, представленої в документації [19].

Входом в мережу є квадратне кольорове зображення, тобто в експериментах використовується обличчя, виділене з відеокадру. Насправді, зазвичай рекомендується відстежувати інформацію про обличчя, а не використовувати повний кадр в якості вхідних даних для мережі для підвищення точності класифікації. Крім того, обличчя можуть бути легко використані з зображень за допомогою будь-якого існуючого детекторів обличчя, запропонованих в літературі. Вихід мережі є вектором

ознак з 1792 елементів, визначуваним як *f(I)*. Остаточний результат, пов’язаний з обличчям, є результатом рівня класифікації.

Такі алгоритми, як архітектура Transformer або Residual Attention Networks демонструють як нейронні мережі можуть дізнатися, яка частина її вхідних даних, частина зображення або послідовністю слів, є актуальною для виконання поставленого завдання. У контексті виявлення зображень, згенерованих за допомогою технології DeepFake, дуже корисно знати, яка саме частина входу дає нейронній мережі велику частину інформації для ухвалення рішення. Механізм Attention був узятий із згаданих вище двох робіт. Attention блок продемонстрований на рис. 4.1, у червоному прямокутнику. У цього шару є дві основні особливості:

- на вхід подається карта ознак, витягнута за допомогою EfficientNetB4 до певного шару, обрана так, щоб ці ознаки мали достатню інформацію про вхідний кадр, не будучи занадто детальними або, навпаки, занадто невизначеними. Для цього беруться ознаки, витягнуті після третього блоку MBConv, розмірність якого 28×28×56;

- ознаки обробляються тільки одним згортальним шаром розмірністю 1, у наслідку чого передаються в активаційну функцію сигмоїда, для отримання єдиної Attention карти.

З одного боку, цей простий механізм дозволяє мережі зосередитися тільки на найбільш вагомих частинах ознак об’єктів, з іншого боку, він дає глибше розуміння того, які частини вхідного слою мережа вважає найбільш інформативними. Дійсно, отримана Attention карта може бути легко зіставлена з вхідним зразком, підкреслюючи, яким елементам було приділено більше уваги нейронною мережею. Результат Attention блоку остаточно обробляється блоками EfficientNetB4, що залишилися. Уся процедура навчання може бути виконана end-to-end підходом. Отримана модель була названа EfficientNetB4Att.

Навчання моделей відбувалося дотримуючись двох парадигм: наскрізне навчання і сіамське. Перший представляє більше класичну

стратегію навчання, що також використовується як метрика оцінки на наборі даних DFDC. Інше спрямоване на використання можливостей узагальнень, пропонованих нейронними мережами, для отримання дескриптора функції, який дає перевагу подібності між вибірками, що належать до одного й того ж класа. Кінцева мета полягає в тому, щоб навчити представлення в просторі декодованих шарів мережі, які добре розділяють вибірку (тобто обличчя) реального і фальшивого класу.

Наскрізне навчання: мережа отримує зразки обличчя, і мережа повертає score, пов'язаний з обличчям $\widehat{y_i}$. Ця оцінка ще не пройшла через функцію активації сигмоїда. Оновлення ваг здійснюється за допомогою використовуваної функції LogLoss:

$$L_L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(S(\widehat{y_i})) + (1 - y_i)\log(1 - S(\widehat{y_i}))], \qquad (4.1)$$

Змінна $\widehat{y_i}$ представляє $i$-у оцінку обличчя, $\widehat{y_i} \in \{0,1\}$, що відповідає міткам обличчя. Зокрема, мітка 0 пов'язана з обличчями, що виходять із справжніх початкових відео, а мітка 1 − з підробними. N − загальна кількість облич, використовуваних для навчання, а $S(\widehat{y_i})$ − активаційна функція сигмоїда. Сіамське навчання: triplet margin loss була адаптована для цього типу навчання:

$$L_T = \max(0, \mu + \delta_+ - \delta_-), \qquad (4.2)$$
$$\delta_+ = ||f(Ia) - f(Ip)||_2,$$
$$\delta_- = ||f(Ia) - f(In)||_2,$$

де $\delta_+$та $\mu$ − позитивні значення;

$I_a$ – реальне зображення;

$I_p$ − позитивний зразок, що належить до того ж класу, що й $I_a$ (інше реальне зображення);

$I_n$ – зображення, згенероване за допомогою DeepFake.

Потім завершується навчання, настроюючи простий шар класифікації поверх мережі, наслідуючи описаний вище наскрізний підхід.

## 4.2 Експерименти виявлення підроблених зображень з використанням ансамблю згортальних класифікаторів

Експерименти проводилися на двох наборах даних: FF++ та DFDC.

FF ++ − це великий набір даних по маніпуляціях з обличчям, створений з використанням state-of-the- art методів редагування відео. Цей набір даних містить два класичних підходи маніпуляції обличчями, тобто Face2Face і FaceSwap, разом з двома стратегіями, ґрунтованими на навчанні, тобто DeepFake і NeuralTextures. Кожен метод застосовується до 1000 високоякісних відео, завантажених з YouTube, які обираються вручну, щоб показувати образи особи без перешкод і зайвих об’єктів. Усі послідовності містять не менше 280 кадрів. Створюється база даних із понад 1,8 мільйонів зображень з 4000 оброблених відео. Для імітації реалістичних налаштувань відео стискаються з використанням кодека H.264. Відео високої якості, а також відео низької якості генеруються з використанням параметра квантування з постійною швидкістю, рівною 23 і 40 відповідно.

DFDC – найбільший набір підроблених відео, зібраний спільно компаніями Microsoft, Facebook і Amazon. Він складається із понад 119000 відеофрагментів, створених спеціально для задачі класифікації відео, згенерованих за допомогою DeepFake, що представляють як реальні, так і підробні відеоролики. Справжні відеоролики − це послідовності невеликих відео з акторами, обранними по декількох чинниках (стать, шкіра, вік і тому подібне) [21].

Підробні відеоролики створюються, паралельно із справжніми, застосовуючи різні методи DeepFake, наприклад, різні алгоритми зміни

особи. Точні алгоритми, використовувані для створення підроблених відео не відомі. Довжина послідовності складає приблизно 300 кадрів, і класи сильно розбалансовані по відношенню до підробних відео, налічуючи приблизно 100000 підробок і 19000 початкових відео.

Для створення ансамблю були обрані наступні моделі архітектури:

− XceptionNet оскільки це найбільш ефективна модель для виявлення маніпуляцій на зображеннях з обличчями;

− EfficentNetB4 оскільки ця архітектура являється state-of-the- art у багатьох завданнях пов’язаних з обробкою зображень;

− EfficentNetB4Att оскільки вона здатна виділяти найбільш релевантні області зображень.

Кожна модель навчається і тестується незалежно від інших. Що стосується FF++, розглядається тільки відео, згенеровані з квантуванням з постійною швидкістю, рівним 23. XceptionNet навчається з використанням одного і того ж підходу для навчання, в той час як дві моделі EfficientNet навчаються як наскрізним, так і сіамським способами.

При цьому в результаті виходить чотири навчених моделі: EfficientNetB4 і EfficientNetB4Att, які навчаються за допомогою класичного наскрізного підходу, а також EfficientNetB4ST і EfficientNetB4AttST, навчені з використанням сіамської стратегії. Усі ці моделі похідні від EfficientNetB4 та можуть сприяти остаточному об’єднанню.

Для кожного набору даних були застосовані різні способи розділення даних. DFDC був розбитий по структурі папок, використовуючи перші 35 папок для навчання, папки від 36 до 40 для перевірки і останні 10 папок для тестування. Що стосується FF ++, використовувалося таке ж розділення, як в [17], вибираючи 720 відео для навчання, 140 для перевірки і 140 для перевірки з пулу оригінальних послідовностей, узятих з YouTube. Відповідні неправдиві відео присвоюються одному і тому ж розділенню. Усі результати показані на тестових наборах даних у Додатку А на рис. А.1 та на рис. А.2.

У експериментах була розглянута обмежена кількість кадрів для кожного відео. На етапі навчання цей вибір мотивується двома основними міркуваннями: при використанні дійсно невеликої кількості кадрів на відео спостерігається сильна тенденція до перенавчання; збільшення кількості кадрів не покращує продуктивність виправданим чином. Це явище можна помітити на рис. 4.2, де представлені втрати при навчанні і перевірці залежно від ітерацій навчання, при виборі змінної кількості кадрів на відео.

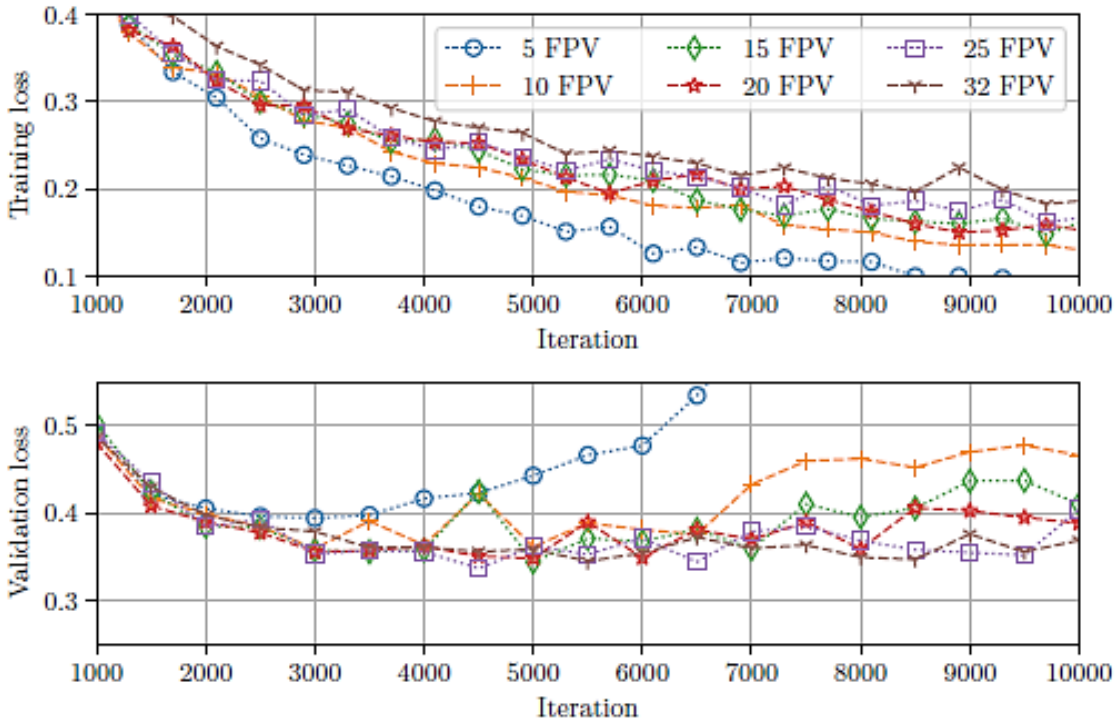


Рисунок 4.2 – Графік взаємозв’язку кількості кадрів з помилкою

Слід відзначити, що мінімальні втрати при перевірці не покращують вибір 15 кадрів на відео замість 32, проте вибір 32 кадрів на відео допомагає запобігти перенавчанню. Для тестування також треба брати до уваги апаратні і часові обмеження, що накладаються завданням DFDC. Маючи це на увазі, була обмежена кількість аналізованих кадрів з кожної послідовності до 32 для етапів навчання і тестування. Навіть при такому налаштуванні розміри наборів даних залишаються значними: для FF++

виходить приблизно 1,6 мільйонів зображень, а для DFDC – 3,4 мільйони кадрів.

Обсяг даних можна скоротити, нагадавши, що не повна інформація про кадри корисна для процесу глибокого виявлення [17]. Дійсно, можна в основному сфокусувати аналіз на регіоні, де знаходиться обличчя суб'єкта. Отже, в якості кроку попередньої обробки витягається обличчя з кожного кадру сцени за допомогою екстрактора BlazeFace [32], який в експериментах виявився швидший, ніж детектор MTCNN [31], використовуваний авторами [17]. У разі виявлення більш однієї особи, зберігається обличчя з найкращою оцінкою достовірності. В результаті вхідними даними для мереж є квадратне кольорове зображення розміром 224×224 пікселі.

Під час навчання і перевірки, щоб зробити моделі надійнішими, виконуються операції доповнення даних на вхідних гранях. Зокрема, випадковим чином застосовується масштабування, горизонтальне відображення, випадковий контраст яскравості, насиченість відтінку, додавання шуму і, нарешті, стискання JPEG. Зокрема, використовується бібліотека Albumentation [36] в якості бібліотеки доповнення даних, тоді як Pytorch [37] використовується як фреймворк Deep Learning. Для навчання використовується оптимізатор Адама [38] з гіперпараметрами, що становлять $\beta_1 = 0{,}9;\ \beta_2 = 0{,}999;\ \epsilon = 10^{-8}$, а початкова швидкість навчання становила $10^5$.

Незалежно від використовуваної стратегії навчання, враховуючи розмір наборів даних, вони не використовуються в повному розмірі для навчання однієї епохи.

Для наскрізного навчання береться або максимум 20000 ітерацій, вказуючи в якості ітерації обробку пакету з 32 прикладів (16 реальних, 16 підроблених), узятих випадковим чином і рівномірно по усіх відеозображеннях навчальної вибірки, або відбувається переривання навчання при досягнення плато на функції втрат. Валідація моделі в цьому

контексті виконується кожні 500 ітерацій навчання, на 6000 прикладів, знову узятих рівномірно і випадковим чином по усіх відео навчального набору даних. Початкова швидкість навчання зменшується з коефіцієнтом 0,1, якщо втрати перевірки не зменшуються після 10 процедур перевірки (5000 повчальних ітерацій), і навчання припиняється, коли досягнено мінімальної швидкості навчання $1\times10^{-10}$.

Для сіамського навчання, засіб витягання особливостей навчається з тією ж кількістю ітерацій, валідацією і швидкістю навчання, як і при наскрізному навчанні. Основною відмінністю цих двох підходів є в різних функціях втрати і у розмірі пакету, який в даному випадку складає дванадцять триплетів прикладів (шість початкових, шість підроблених і шість з наявністю як початкових так і підроблених ознак). Пакети обираються пропорційно їх навчальній вибірці. Точне налаштування рівня класифікації потім виконується послідовно, наслідуючи парадигму наскрізного навчання з гіперпараметрами, описаними раніше.

Навчання моделей відбувалося з використанням NVIDIA 2070 RTX Super та пакету Apex, а також із застосуванням техніки Accumulation Gradient. Ці підходи дозволили тренувати глибокі моделі на цій відеокарті, проте уповільнили навчання і показали результати гірше, ніж без них. Кращі моделі були отримані при навчанні на виділених компанією Google відеокартах Titan V, які надаються для вирішення цієї проблеми.

Після навчання був проведений аналіз отриманих результатів.

При застосуванні EfficientNetB4Att моделі, щоб показати ефективність Attention механізму при витяганні найбільш інформативного вмісту облич, була оцінена Attention карта, розрахована для декількох облич з набору FF++. На рис. 4.1 в якості вихідного шару був вибраний сигмоїд у блоці уваги, який є 2d-картою з розміром 28×28. Потім він був збільшений до вхідного розміру обличчя (224×224) і накладений на вхідне обличчя. Результати наведені на рис. 4.3.

Слід відзначити, що цей простий механізм уваги дозволяє виділити найбільш деталізовану частину обличчя, наприклад очі, рот, ніс і вуха. Навпаки, плоскі області (де градієнти малі) не є інформативними для мережі. Кілька разів було показано, що артефакти методів глибокої генерації в основному локалізовані навколо рис обличчя [16]. Наприклад, грубо змодельовані очі і зуби з надмірно білими областями, як і раніше, є основними ознаками методів підробки зображень.

Рисунок 4.3 – Ефект застосування Attention механізму

Щоб зрозуміти, чи є ознаки, що створюються шарами кодування мережі при навчанні сіамським способом, дискримінаційними для завдання, було обраховано проекцію по зменшеному простору, використовуючи добре відомий алгоритм t-SNE [39]. На рис. 4.4 показана проекція, отримана за допомогою EfficientNetB4Att, починаючи з 20 відеороликів набору даних FF++. Можна помітити, як кадри одного і того ж відео об’єднуються в невеликі субрегіони. Важливо, що усі початкові зразки згруповані у верхній області діаграми, тоді як підробні зразки знаходяться в нижній області. Кадри з тих же відео кластерів об’єднуються в менші субрегіони. Це виправдовує вибір на користь прийняття цієї парадигми навчання на додаток до класичного наскрізного підходу.

Оскільки важливо зрозуміти, чи можуть різні мережі використовуватися в ансамблі, досліджується, чи є оцінки, витягнуті кожною моделлю, в деякій мірі незалежними. На рис. А.1 і рис. А.2 усіх

графіків за межами основної діагоналі показують, що різні мережі дають трохи різні оцінки для кожного кадру. Дійсно, хмари точок не ідеально вирівняні за формою, яку можна легко описати простим співвідношенням. Це спонукає до використання різних навчених моделей для ансамблю. Якби усі мережі були ідеально корельовані, це було б нереально.

Рисунок 4.4 – t-SNE візуалізація ознак, отриманих EfficientNetB4Att при навчанні сіамським способом

З отриманих моделей формувалися ансамблі різними способами, використовуючи дві, три або більше моделей. Результат обчислювалося просто як середнє значення з усіх моделей, що входять в ансамбль. На рис. 4.5 показані результати обчислення AUC і LogLoss, отримані в експериментах. Результати надаються по кадрах.

Аналізуючи отримані результати, варто відмітити, що стратегія ансамблювання моделей зазвичай обирається з точки зору характеристик. Кращі результати завжди досягаються комбінацією з двох або більше мереж, а це свідчить, що об'єднання мереж допомагає збільшити точність глибокого виявлення (оціненого за допомогою AUC), так і якості виявлення (оціненого за допомогою міри LogLoss). Дійсно, для обох наборів даних LogLoss і AUC завжди краще, ніж baseline.

| Xception Net | EfficientNet B4 | B4ST | B4Att | B4AttST | AUC FF++ | AUC DFDC | LogLoss FF++ | LogLoss DFDC |
|---|---|---|---|---|---|---|---|---|
| ✓ | | | | | *0.9273* | *0.8784* | *0.3844* | *0.4897* |
| | ✓ | | | | 0.9382 | 0.8766 | 0.3777 | 0.4819 |
| | | ✓ | | | 0.9337 | 0.8658 | 0.3439 | 0.5075 |
| | | | ✓ | | 0.9360 | 0.8642 | 0.3873 | 0.5133 |
| | | | | ✓ | 0.9293 | 0.8360 | 0.3597 | 0.5507 |
| | ✓ | ✓ | | | 0.9413 | **0.8800** | 0.3411 | 0.4687 |
| | ✓ | | ✓ | | 0.9428 | **0.8785** | 0.3566 | 0.4731 |
| | ✓ | | | ✓ | 0.9421 | 0.8729 | 0.3370 | 0.4739 |
| | | ✓ | ✓ | | 0.9423 | 0.8760 | 0.3371 | 0.4770 |
| | | ✓ | | ✓ | 0.9393 | 0.8642 | **0.3289** | 0.4977 |
| | | | ✓ | ✓ | 0.9390 | 0.8625 | 0.3515 | 0.4997 |
| | ✓ | ✓ | ✓ | | **0.9441** | **0.8813** | 0.3371 | **0.4640** |
| | ✓ | ✓ | | ✓ | 0.9432 | 0.8769 | **0.3269** | **0.4684** |
| | ✓ | | ✓ | ✓ | **0.9433** | 0.8751 | 0.3399 | 0.4717 |
| | | ✓ | ✓ | ✓ | 0.9426 | 0.8719 | 0.3304 | 0.4800 |
| | ✓ | ✓ | ✓ | ✓ | **0.9444** | 0.8782 | **0.3294** | **0.4658** |

Рисунок 4.5 – Результати роботи різних ансамблів

Можливість визначити, чи містить відео підробний контент, нині має первинне значення, враховуючи значний вплив відео в повсякденному житті і в масових комунікаціях. Описана модель займається виявленням маніпуляцій з обличчям у відеопослідовностях, націлених на класичну комп'ютерну графіку, а також на підробних відео, згенерованих за допомогою глибинного навчання. Запропонований метод є похідним з

сімейства моделей EfficientNet і покращує нещодавно запропоноване рішення, досліджуючи безліч моделей, навчених з використанням двох основних концепцій: Attention механізм, який генерує зрозуміле для людини виведення моделі, збільшуючи здатність мережі до навчання; стратегія навчання по сіамському триплету, яка витягає з даних глибокі особливості для досягнення кращих результатів класифікації. Результати, оцінені по двох загальнодоступних наборах даних, що містять майже 120000 відеороликів, показують, що запропонована стратегія ансамблю є ефективним рішенням для виявлення маніпуляцій на обличчі. Інтелектуальні схеми голосування, коли аналізується більше кадрів, можуть привести до підвищеної точності.

Для покащення точності ансамблю доцільно спробувати глибші види архітектури EfficientNet, такі, як EfficientNetB6 і EfficientNet7. Ця архітектура не була розглянута, оскільки для їх навчання потрібний кластер відеокарт. Так само можлива зміна коефіцієнтів моделей в ансамблі могло б покращити результати, проте, це так само могло привести до меншої узагальненості моделі для інших наборів даних. Для стратегії наскрізного навчання можливо застосувати групу функцій втрат, що зробить модель більше узагальненою, проте може привести до погіршення результатів на досліджуваних наборах даних. Так само мало сенс спробувати подавати на вхід моделі зображення більшої роздільної здатності, проте, для цього б знадобилося набагато більше обчислювальних ресурсів.

## 4.3 Результати інших досліджень на вибірці DFDC

Незабаром повинна вийти наукова стаття з назвою «Keep it simple and deep. Fake images detection through deep neural networks». Автор цієї статті Селім Сефербоков опублікував стисле зміст майбутньої статті на платформі Kaggle. У цій роботі для підсумкової моделі також використовується ансамбль мереж на основі EfficientNet архітектури.

У розпорядженні автора був кластер з чотирьох NVIDIA 2080 TI відеокарт, тому у нього була можливість тренувати глибші моделі ніж в експериментах, описаних в попередніх розділах.

Перший ансамбль містив 15 EfficientNetB5 моделей, які показали в ансамблі результати LogLoss рівні 0,449. Другий ансамбль містив 7 EfficientNetB7 моделей, результат ансамбля яких дав LogLoss рівний 0,428.

Так само були описані експерименти з EfficientNetB3, EfficientNetB4 і EfficientNetB6. Зі збільшенням глибини моделі покращувалися і результати, проте EfficientNetB6 показав результат трохи гірше, ніж EfficientNetB4. Саме тому для ансамблювання були обрані дві кращих архітектури. Через незбалансований набір даних, моделі показували LogLoss на підроблених відео значно менший, ніж на реальних, через що авторові довелося боротися з проблемою False Positive передбачення. Автор приділив багато часу на попередню обробку зображень. На кожному зображенні за допомогою MTCNN детектора, виділялася ділянка обличчя. Так само з виділеного обличчя бралася SSIM маска. На вхід моделей подавалося зображення розмірністю 380×380 пікселів. Для зображень застосовувалася наступна базова техніка аугумації: стискування зображення, шум, розмиття, зміна розміру з різними інтерполяціями, тремтіння кольору, масштабування і повороти.

Для того, щоб модель краще витягала особливості пов’язані зі зміною частин обличчя, на вхід моделі також подавалися зображення наступних видів: половина обличчя по вертикалі або горизонталі видалялась із зображення (накладалася чорна маска, яка не враховувалася б мережею); затемнення частин обличчя (очі, ніс або ріт), використовуючи для цього орієнтири виділені за допомогою MTCNN. Приклади зображень після аугумації показані на рис. 4.6.

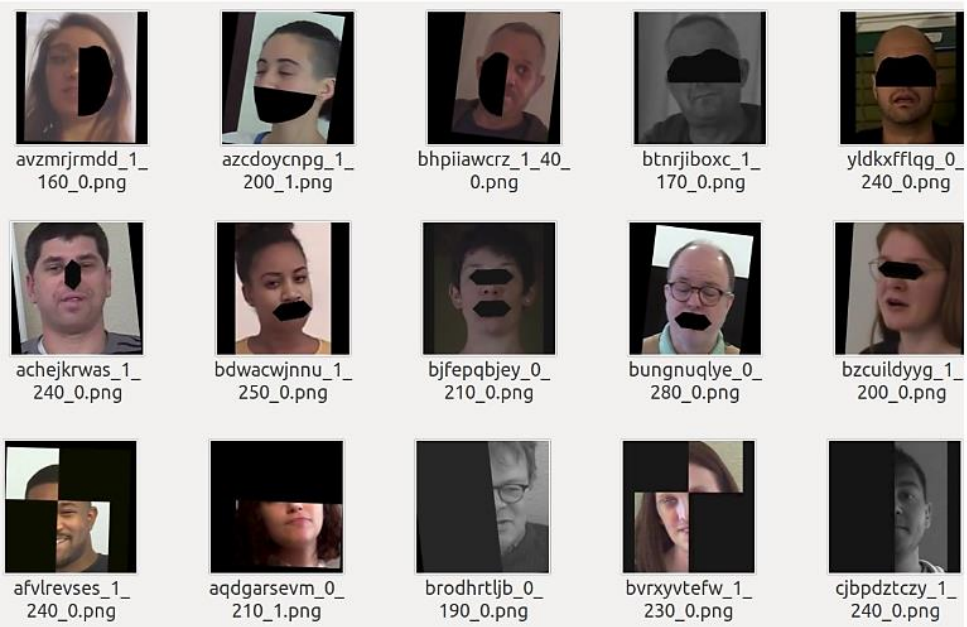


Рисунок 4.6 – Приклади зображень після застосування аугумації.

Для навчання використовувалося 2500 ітерацій для однієї епохи. Більша кількість епох вела до перенавчання. Так само була застосована техніка стохастичного градієнтного спуску, щоб зменшувати вірогідність перенавчання. Для зміни швидкості навчання був узятий підхід PolyLR з початковим значенням 0,01. Моделі навчалися на чотирьох NVIDIA 2080 TI у якості одного кластера і двох NVIDIA Titan V в якості другого кластера із застосуванням пакету Apex з режимом змішаної точності. Як видно, глибші моделі дають результати краще. А точність EfficientNetB5 і EfficientNetB7 помітно краще ніж у стандартної EfficientNetB4. Проте завдяки Attention механізму EfficientNetB4 показує такі ж результати, як і глибший аналог EfficientNetB7, а при використанні сіамського навчання, модель навіть здатна пересилити свій глибший аналог. З цього можна зробити висновок, що за наявності обчислювальних ресурсів, можна було навчити EfficientNetB7Att моделі, а також об’єднавши їх в ансамбль, отримати результати набагато кращі, чим в проведеному дослідженні. Так само в досліджуваному ансамблі використовувалися всього 4 моделі, у той час коли автор найновітнішої моделі навчив 7 моделей.

# ВИСНОВКИ

С кожним роком процес розвитку глибокого навчання стає дедалі швидшим завдяки збільшенню потужності техніки та інформаційним збагаченням. Нові технології значно спрощують життя та роботу багатьом людям, але не всі алгоритми використовують на користь суспільству. Та чим більше потенційної шкоди можуть нанести алгоритми, тим більше способів протидії їм знаходять.

Технологія DeepFake на сьогоднішній день становить загрозу для інформаційного суспільства. Адже його можливо використовувати не тільки на користь (замінивши обличчя кіноактора на інше), але й на шкоду. Реальною проблемою цього алгоритму є масова дезінформація та генерування провокаційних медіа матеріалів.

Використовуючи найновітніші підходи роботи із зображеннями глибокого навчання, в цій роботі була продемонстрована модель, що показує результати краще, ніж ті, що існують до появлення набору даних DFDC. Як описано в атестаційній роботі, використання глибших моделей може значно покращити результати.

Гіпотеза, яка була покладена в основу виконаної роботи підтверджена: застосування сучасної архітектури згортальних мереж, таких як сімейства EfficientNet разом з Attention механізмом, надало відмінний результат. Також при об’єднанні декількох різних моделей в ансамбль отримані результати були значно кращі.

У дослідженні використовувалися новітні технології і програмні продукти у сфері комп’ютерних наук. Для проведення дослідження був вивчений необхідний навчальний матеріал, був знайдений набір даних, а також був проведений детальний аналіз щодо визначення послідовності дій проведення дослідження. Усе це відповідає виконанню раніше поставлених завдань.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Rafael C. Gonzalez, Richard E. Woods Digital Image Processing, 4th edition Pearson/Prentice Hall, 2018. – 1168p.

2. David A. Forsyth, Jean Ponce Computer Vision: A Modern Approach (2nd ed.). Pearson Education Limited, 2015. – 792p.

3. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis, and Machine Vision (4th Edition). Cengage Learning, 2014.

4. Ian Goodfellow, Yoshua Bengio, Aaron Courville Deep Learning. – MIT Press, 2016. – 787p.

5. Peter Norvig , Stuart Russell Artificial Intelligence: A Modern Approach, Global Edition. – Pearson Education Limited, 2016. – 1152p.

6. Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, Andrew Zisserman. 2010, The pascal visual object classes (voc) challenge. International journal of computer vision 88(2), pp. 303-338.

7. Pierre Baldi. 2012. Autoencoders, Unsupervised Learning, and Deep Architectures.

8. Alex Sherstinsky. 2018. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.

9. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. 2014. Generative Adversarial Networks.

10. Alec Radford, Luke Metz, Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.

11. Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey. 2015. Adversarial Autoencoders.

12. Ayush Jaiswal, Wael AbdAlmageed, Yue Wu, Premkumar Natarajan. 2017. Bidirectional Conditional Generative Adversarial Networks.

13. Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc

Thanh Nguyen and Saeid Nahavandi. 2019. Deep Learning for DeepFakes Creation and Detection.

14. Yuval Nirkin, Yosi Keller, Tal Hassner. 2019. FSGAN: Subject Agnostic Face Swapping and Reenactment.

15. Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, Jan Kautz. 2017. MoCoGAN: Decomposing Motion and Content for Video Generation.

16. David G¨uera Edward J. Delp. 2019. DeepFake Video Detection Using Recurrent Neural Networks.

17. Umur Aybars Ciftci, Ilke Demir, Lijun Yin. 2019. FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals.

18. Darius Afchar, Vincent Nozick, Junichi Yamagishi, Isao Echizen. 2019. MesoNet: a Compact Facial Video Forgery Detection Network.

19. Yuezun Li, Siwei Lyu. 2019. Exposing DeepFake Videos By Detecting FaceWarping Artifacts.

20. Mingxing Tan, Quoc V. Le. 2019 EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.

21. DeepFake Detection Challenge (DFDC): https://DeepFakedetectionchallenge.ai/