МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут захисту інформації

На рецензію
Завідувач кафедри УІКБ
доктор економічних наук, професор
_____ С.В. Легомінова
«_____» _____2020 р.

До захисту
Завідувач кафедри УІКБ
доктор економічних наук, професор
_____ С.В. Легомінова
«_____» _____2020 р.

ДИПЛОМНА РОБОТА

на тему

СПОСОБИ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У ФОРМАТІ «DEEP FAKE

СТУДЕНТ: Недодай Михайло Геннадійович _____
(підпис)

КЕРІВНИК: доцент кафедри Управління інформаційною та кібернетичною безпекою, к.т.н., доцент
Дзюба Тарас Михайлович _____
(підпис)

НОРМОКОНТРОЛЕР: _____
(підпис)

Київ - 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут захисту інформації
Кафедра управління інформаційною та кібернетичною безпекою
Освітньо-кваліфікаційних рівень – магістр
Спеціальність «Кібербезпека»
Спеціалізація «Управління інформаційною безпекою»

«ЗАТВЕРДЖУЮ»
Завідувач кафедри УІКБ
_____ С.В. Легомінова
(підпис)
“___”_____ 2021 р.

ЗАВДАННЯ
на дипломну роботу
студенту Недодай Михайлу Геннадійовичу

1. Тема роботи «СПОСОБИ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У ФОРМАТІ «DEEP FAKE», затверджена наказом по університету від «11» жовтня 2021 р. № 170.

2. Термін здачі студентом оформленої роботи « 10 » січня 2022 р.

3. Об’єкт дослідження: дезінформація у форматі «deep fake».

4. Предмет дослідження: виявлення дезінформації у форматі «deep fake».

5. Мета дослідження: визначення способів виявлення дезінформації у форматі «deep fake».

6. Перелік питань, які мають бути розроблені:

6.1. Дослідити особливості створення та використання дезінформації у форматі «deep fake».

6.2. Провести аналіз існуючих підходів до виявлення дезінформації у форматі «deep fake».

6.3. Визначити доцільні способи виявлення дезінформації у форматі «deep fake».

7. Дата видачі завдання «14» жовтня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання: « 14 » жовтня 2021 р

№ з/п	Етапи дипломної роботи	Термін виконання етапів	Примітка
1.	Визначення об’єкту, предмету, мети та завдань дослідження	21.10.2021	
2.	Збір та аналіз літератури	15.11.2021	
3.	Дослідження особливостей створення та використання дезінформації у форматі «deep fake»	25.11.2021	
4.	Аналіз існуючих підходів до виявлення дезінформації у форматі «deep fake»	07.12.2021	
5.	Визначення доцільних способів виявлення дезінформації у форматі «deep fake»	24.12.2021	
6.	Формулювання висновків за результатами проведеного дослідження	25.12.2021	
7.	Оформлення роботи	10.01.2022	
8.	Оформлення презентації	11.01.2022	
9.	Отримання рецензії на роботу	12.01.2022	
10.	Захист в ДЕК	18.01.2022	

Студент М.Г. Недодай

Науковий керівник Т.М. Дзюба

РЕФЕРАТ

Текстова частина магістерської роботи : 75 сторінок, 45 рисунків, 4 таблиці, 22 джерела.

Об'єкт дослідження – дезінформація у форматі «deep fake».

Предмет дослідження – виявлення дезінформації у форматі «deep fake».

Мета роботи - дослідження та створення алгоритму визначення deepfake за допомогою алгоритмів штучного інтелекту.

Методи дослідження – у магістерській роботі використані методи розробки алгоритмів штучного інтелекту засновані на перевірках різноманітних гіпотез, які визначаються з наявних даних і розуміння роботи алгоритмів генерації контенту.

Виконуючи поставлені завдання у магістерській роботі проаналізовано сфери застосування deepfake, визначено ті, для яких існує потреба у визначенні згенерованого контенту. Було визначено основні сфери застосування алгоритму, створено рекомендації щодо розробки та використання алгоритмів. Перевірено різноманітні методи створення алгоритмів визначення, визначено найоптимальніший варіант та протестовано його на різних типах deepfake контенту. Надано практичні рекомендації використання створеного алгоритму для реальних задач.

Галузь використання – інформаційні мережі.

ШТУЧНИЙ ІНТЕЛЕКТ, НЕЙРОННІ МЕРЕЖІ, ГЕНЕРАТИВНІ НЕЙРОННІ МЕРЕЖІ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ОБЛИЧЧ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ГЗМ – генеративно-змагальна мережа

ЗГМ – згорткова нейронна мережа

ШМН – штучна нейронна мережа

ШІ – штучний інтелект

ВСТУП

Поява перших відео з відомими кінозірками, які знімались в фільмах не зовсім звичного для них жанру викликала велике здивування у суспільності. Згодом була визначена технологія, яка дозволяє будь-якому користувачу мережі Інтернет замінити обличчя на будь-якому відео обличчям довільної особи. Після цього стався розвиток алгоритмів, які були названі Deepfake. Відтоді такі відео ставали дедалі кращими та фотореалістичнішими, а відео з використанням цієї технології стали частіше з'являтись на відомих платформах. Звісно, користувачі швидко знайшли оригінали відеозаписів та зрозуміли, що вони були згенеровані, але це ще більше посилило інтерес до нового алгоритму.

Після того, як суспільність зрозуміла небезпеку таких відеозаписів. Через невеликий проміжок часу почалось і справжнє реальне використання згенерованих відео. Першим від цього постраждали політики, які і стали жертвами для різних ентузіастів, які тепер могли створювати реалістичні відеозаписи з текстами, які не відповідають дійсності.

Після такого успіху в стороні не лишились і розробники мобільних додатків, які використовували алгоритми deepfake в розважальних цілях. Користувачам дали можливість замінювати обличчя на відеозаписах з кінофільмів на своє обличчя та ділитись такими відео з друзями. Звісно, на ці відео використовуються лише в розважальних цілях і не можуть бути використані в інших цілях.

Користувачі швидко визнали переваги deepfake та використовувати його в різноманітних цілях. Від використання аматорами для створення кінофільмів до використання в цілях отримання неправомірної вигоди.

Актуальність: досліджуваної теми полягає в тому, що зараз алгоритми створення deepfake досягли фотореалістичної якості, а простота їх використання дає змогу будь-якому користувачу з потужним персональним комп'ютером створити відео, яке можна використовувати в різних цілях. Частіше всього такі відеозаписи порушують права громадян, а також можуть завдавати значних репутаційних та фінансових збитків.

Мета роботи: дослідження та створення алгоритму визначення deepfake за допомогою алгоритмів штучного інтелекту.

Об'єкт дослідження – дезінформація у форматі «deep fake».

Предмет дослідження – виявлення дезінформації у форматі «deep fake».

Для досягнення мети роботи вирішувались часткові наукові завдання:

досліджені особливості створення та використання дезінформації у форматі «deep fake»;

проведений аналіз існуючих підходів до виявлення дезінформації у форматі «deep fake»;

визначені доцільні способи виявлення дезінформації у форматі «deep fake» з розробкою власного алгоритму.

Методи дослідження: у магістерській роботі використані методи розробки алгоритмів штучного інтелекту засновані на перевірках різноманітних гіпотез, які визначаються з наявних даних і розуміння роботи алгоритмів генерації контенту.

Наукова новизна одержаних результатів: в магістерській роботі я пропоную використовувати власний алгоритм виявлення дезінформації в форматі «deepfake», який дозволяє виявити найкращу комбінацію для свого «навчання» в умовах обмежених обчислювальних ресурсів, тобто його практична реалізація можлива на абсолютній більшості звичайних комп’ютерів.

Практичне значення одержаних результатів: створення працюючого алгоритму для визначення deepfake.

РОЗДІЛ 1 ОСОБЛИВОСТІ СТВОРЕННЯ ТА ВИКОРИСТАННЯ ДЕЗІНФОРМАЦІЇ У ФОРМАТІ «DEEP FAKE»

1.1 Опис технології, проблеми.

Розвиток інформаційних технологій сучасності призвів до того, що зараз для отримання інформації потрібно лише ввести свій запит до пошукової системи, яка одразу поверне безліч результатів. З цього виникає проблема – в різних джерелах може бути представлена різна відповідь на одне й те саме питання. Інша проблема – досить мала частинна користувачів Інтернету буде перевіряти факти, які вони знаходять в мережі. А з розвитком штучного інтелекту, зловмисники, шахраї, інші зацікавлені особи можуть генерувати реалістичну інформацію, яка на перший погляд нічим не буде відрізнятись від інформації, яку наприклад може сказати авторитетний учений. Проте штучний інтелект може як генерувати фейки, так і допомагати в їх перевірці та ідентифікації. Насамперед, в області фото- відео контенту. Deepfake – конкатенація слів «глибинне навчання» deep та «підробка» fake - методика синтезу зображення людини, яка базується на штучному інтелекті. Вона використовується для поєднання і накладення існуючих зображень та відео на початкові зображення або відеоролики. Це відносно нова технологія, перші роботи над якою було опубліковано у 2014 році. Проте за невеликий проміжок часу вона зробила стрибок від невеликих згенерованих зображень низької якості до фото реалістичних зображень високої роздільної здатності. Це стало можливим з розвитком обчислювальних технологій, більш потужних та дешевих графічних прискорювачах та завдяки мережі Інтернет, завдяки якій дослідники змогли зібрати необхідну кількість даних для тренування нейронних мереж, що мали на меті генерувати фотореалістичний контент. Як відомо, «Інтернет пам’ятає все», тож навіть якщо видалити свій профіль в соціальній мережі, велика ймовірність, що десь на серверах може залишитись його копія, або автоматичні програми вже завантажили інформацію до архіву Інтернету, включно з відкритими фотографіями. Не кажучи про безліч атак, з якими борються всі популярні ресурси в мережі. Більшість атак не є успішною, проте в разі успіху зловмисники можуть отримати безліч даних про клієнтів. В найкращому, для клієнта випадку зловмисники отримують данні про електронну пошту, паролі, та іншу не зовсім чутливу інформацію, наприклад ПІБ, дата народження, історія замовлень в інтернет-магазині. Для убезпечення себе потрібно використовувати різні паролі на всіх ресурсах мережі інтернет.

Сьогодні всі веб-браузери дозволяють автоматично генерувати за зберігати паролі. Звісно, більшу безпеку дадуть окремі програми для збереження паролів, адже користувачі також не застраховані від комп'ютерних вірусів, які можуть дістати бережені паролі та надати зловмисникам доступ до всіх акаунтів жертви. Проте, скоріше всього зловмисникам не цікаво цілеспрямовано атакувати конкретного користувача мережі, більш вигідно атакувати певний ресурс, який зберігає дані багатьох користувачів. До чутливої інформації можна віднести банківські дані, дані персональних документів, дані про адреси. Звісно, рекомендації про захист користувачів та інтернет-ресурсів виходять за рамки даної роботи. З розвитком технології, зловмисники отримавши ваші паспортні дані та фотографію можуть створити вашу цифрову копію для власних цілей, таких як отримання кредиту, реєстрація в певних фінансових сервісах, створення фотографії та безліч інших потенційних цілей. До того ж, створення такої копії не потребує великої кількості ресурсів, або досвідченої команди комп'ютерних дизайнерів.

Технологію Deepfake також можна використовувати на більш якісному рівні, наприклад в інформаційних або політичних кампаніях, які зазвичай мають великі ресурси, що можна використати для більш якісної обробки результатів роботи алгоритмів Deepfake та ширшого розповсюдження неправдивого контенту. Наприклад, якщо директор компанії конкурента, або політик з протилежної виступить з відеозверненням, в якому буде доносити певну інформацію, які в реальному житті ніколи не казав. Звісно, можна як завгодно довго спростовувати цю інформацію, проте це займе певний час, на створення звернення-спростування, його розповсюдження. За цей час дезінформація згенерованого відео встигне розповсюдитись, та завдати збитки, як репутаційні, так і потенційно фінансові. Також зловмисники можуть використовувати фековий контент для шантажу високопоставлених осіб, або ж для залякування. Наприклад, якщо замінити обличчя людини в порнографічному матеріалі за участю з неповнолітніх осіб, знайти який в «глибокому інтернеті» за наявності певних знання в цій області не складно, відео може стати реальним приводом для зацікавлення правоохоронних органів, що може створити обмеження, наприклад в займанні жертвою певних керівних позицій в компанії. Цей час зловмисники можуть використовувати за власним розсудом. Інший приклад – генерація цифрової версії людини для отримання певних привілеїв в компанії, де працює реальна людина. Наприклад,

неуважний співробітник, якому зателефонував директор, і попросив надати доступи до корпоративної мережі, надіслати документи, або виконати певні інші несанкціоновані дії, які неминуче призведуть до збитків компанії.

Подібні використання deepfake можна віднести до інструментів соціальної інженерії, за допомогою якої хакери і зловмисники частіше всього зламують системи та завдають збитків. Адже якою б не була система захисту – найслабша частина в ній це людина. Тому атаки з використанням цієї технології на інформаційні ресурси значною мірою розраховані на атаку на сприйняття людини. Є певні випадки, в яких згенерованим контентом намагаються проходити автоматичну верифікацію, яка являє собою алгоритм, але все ж на даний час більше розвинуті атаки саме на людське сприйняття. Це може бути атакою на конкретну особу, але також може бути застосоване для керування масами.

До найбільш нової проблеми використання Deepfake можна віднести розвиток індустрії NFT та штучних вселених. Невзаємозамінні токени (NFT) — це особливий тип криптовалюти, який є незамінним, на відміну від більшості криптовалют та багатьох мережевих або службових токенів. Являє активи, що існують лише у власних криптосистемах. Блокчейн – розподілена база даних, що зберігає впорядкований ланцюжок записів, що постійно довшає. Захистом від підробки або спотворення слугує включення хешу всього блоку в наступний блок. Таким чином, внесення змін в один з блоків вимагає зміни всіх наступних блоків після нього. Так, користувачі можуть створювати власні копії для використання в блокчейн системах, або в програмах з доповненою реальністю. Що також означає, що створити віртуальну особу або аватара можна для будь-якої людини, без її дозволів. І далі, зловмисники можуть використовувати віртуального аватара людини для будь-яких цілей. Для людини, чий аватар використовують без її відому це може мати різні наслідки, як від потенційної втрати прибутку, якщо аватар використовують в рекламних цілях, до прямих репутаційних або фінансових втрат від неправомірного використання аватару. Так як це блокчейн система, і будь-який користувач може купити та використовувати аватар, а його видалення майже неможливе з блокчейн системи, можна використовувати штучний інтелект як для продажу створених моделей, так і від його використання. Також, NFT-контент підтримує фото та відео, і зловмисник може в цілях шантажу може створити невидяляєму копію контенту з використанням біометричних даних людини, яка буде доступна для

всіх користувачів мережі Інтернет. Або ж, жертва може власноруч викупити токен та не показувати його іншим.

Всі ці проблеми виникли відносно нещодавно, але вже становлять реальну інформаційну небезпеку. До того ж, засоби протидії таким загрозам поки що знаходяться в стадії розробки. І навіть якщо такі програми стануть доступними, зловмисники теж стануть покращувати свої алгоритми, тож це може стати схожим на протистояння комп'ютерних вірусів та антивірусних програм. Штучний інтелект для створення Deepfake значно покращився з моменту перших публікацій такого контенту, проте він все ще не є ідеальним: на вихідному зображенні можуть бути артефакти (області з незрозумілим змістом – зміна кольорів, плями, розмитість), проте такі області досить рідко зустрічаються, а їх площа невелика. До того ж, зловмисники добре володіють соціальною інженерією, і такі технології скоріше всього будуть допоміжним інструментом в їх цілях, адже інформаційні загрози існували і до появи штучного інтелекту, з ним вони лише посиляться. Це пов'язано з низькою кваліфікацією багатьох користувачів в основах інформаційної безпеки та цифровій гігієні. Наприклад, якщо раніше люди могли в телефонній розмові зі «співробітниками банку» сказати дані своєї кредитної картки шахраям, то скоріше всього, якщо таким користувачам зателефонувати за допомогою відео зв'язку, і попросити сказати дані картки – результат не зміниться. Нажаль, результат скоріше всього не змінить навіть гіпотетична система розпізнавання Deepfake, користувачі скоріше всього проігнорують її попередження про шахраїв. Проте використання технологій розпізнавання Deepfake в медіа може значно зменшити вплив згенерованого контенту на суспільність. Коли платформа, якій людина довіряє, і на якій дивиться згенерований відеоролик буде попереджати користувача, що ймовірніше, контент згенерований автоматично, і містить інформацію, яка не відповідає дійсності, скоріше всього не буде довіряти такому контенту. До цього, Deepfake відео з політиками, медійними особами та іншими різними відомими особистостями, в яких озвучується «повідомлення», матимуть значний вплив на суспільну думку. Наприклад, якщо Deepfake відео з Ілоном Маском буде розповідати про певну криптовалюту, це може мати значний вплив на коливання її ціни на ринку. Через це певні люди, які створили відео можуть отримати значні кошти, але інші люди які повірять цьому відео можуть отримати значні збитки. Через новизну технології можливо, ми ще навіть не знаємо всі можливі варіанти

використання, але алгоритми визначення згенерованого контенту потрібно розробляти вже. Для створення засобів захисту потрібно розуміти алгоритми роботи зловмисників, використовувані технології та потенційні сфери застосування.

Для захисту від Deepfake в засобах біометричної аутентифікації вже розроблені стандарти які враховують потенційні ризики від застосування технологій Deepfake. Наприклад, стандарт ISO/IEC 30107-1:2016 Information technology — Biometric presentation attack detection, випущений у 2016 році описує алгоритми верифікації користувачів за обличчями за допомогою звичайних камер та атаки на них. Стандарт описує атаки на біометричні системи аутентифікації та авторизації та методи автоматичного їх визначення. Атаки, які розглядає стандарт направлені на біометричні сканери під час біометричної авторизації та біометричного сбору даних. Веб-камери також можна віднести до біометричних сканерів, адже вони можуть передавати дані про обличчя користувача, а спеціальне програмне забезпечення може бути застосоване для аутентифікації користувачів.

1.2 Історія виникнення Deepfake.

Розробка методів комп'ютерної лицевої анімації почалась з 1970-х, але прогрес в цій технології з'явився значно пізніше – у кінці 1980-х. Одна з перших найважливіших робіт у цій області – Система кодування рухів обличчя (FACS) була розроблена Карлом-Херманом Хьортсьйо у 1960-х роках, та оновлена Екманом та Фризеном у 1978 році. FACS визначає 46 основних одиниць рухів на обличчі (AU). Основна група одиниць рухів являє собою примітивні рухи м'язів обличчя в різні діях, наприклад посмішка, кліпання, підняття повік. Вісім одиниць рухів призначалась для тривимірних рухів – оберт і нахил вправо і вліво, підйом вверх, вниз, вперед, назад. FACS використовувалась для опису бажаних рухів синтетичних обличчя, а також для відстеження активності обличчя. Використовуючи FACS програмісти можуть вручну кодувати практично будь-які анатомічно можливі рухи обличчя. Використання FACS було запропоновано для використання при аналізі депресії та вимірювання болі в пацієнтів, що не могли виражати свої думки вербально. Також систему було адаптовано для анімації обличчя різних тварин, спочатку для приматів, а потім також для домашніх тварин. Широкого використання система набула у комп'ютерній графіці та створенні комп'ютерних ігор. Через

апаратні обмеження комп’ютерів у часи створення цієї системи, створена нею анімація не могла ввести в оману людей.

У 1990-х комп’ютерна анімація розвивалась значною мірою у кінематографі. Так, при створенні деяких анімаційних фільмів значною мірою застосовувалась і комп’ютерна анімація обличчя. До таких фільмів можна віднести «Історія іграшок» (1995) [60] , «Шрек» (2001) [61]. На прикладі цих фільмів можна побачити значний розвиток анімації обличчя. Проте, створення навіть невеликих штучних анімацій того часу потребувало значних обчислювальних потужностей, а також команду спеціалістів з комп’ютерної графіки. Також з початком росту популярності персональних комп’ютерів почала розвиватись ігрова індустрія, яка теж потребувала якісних лицевих анімацій для персонажів. Проте на відміну від фільмів, у ігор не було часу на якісний рендер кадрів, який використовували кінематографи, а також не було потужного заліза, яке могло б якісно відмалювати картинку в реальному часі.

Подальший розвиток лицевої анімації стався з розвитком технології захоплення рухів, що бере свій початок з технології ротоскопіювання, яка використовувалась ще піонерами кінематографу. Метод захоплення рухів полягає в захопленні рухів актору і створенні його 3-х вимірної моделі. На початку розвитку цієї технології захоплення рухів проводилось за допомогою декількох камер для розрахунку тривимірних положень, які потім проектувались на тривимірну модель. Для захоплення рухів зазвичай використовують спеціальні камери і датчики, які розміщують на акторі. Таким чином створюється тривимірна модель рухів актору, яку потім можна накласти на цифрову оболонку персонажу. Часто цю технологію використовують для захоплення анімації обличчя персонажу. Для цього на обличчя актора наносять датчики, і камера перед обличчям фіксує зміну їх положення. Далі отриману тривимірну модель руху цих датчиків переносять на цифрову модель персонажу.

До переваг цього методу можна віднести:

Низьку затримку, близьку до реального часу.

Низький об’єм робіт – один актор може робити різні анімації для різних моделей.

Складні рухи з фізичною взаємодією.

Недоліки цього методу:

Для отримання результатів потрібне спеціальне обладнання і програмне забезпечення, що значно збільшує вартість створення такого контенту.

Складність маніпуляції отриманими даними.

Неможливість створити універсальну модель рухів для будь-якої тривимірної моделі, на яку вона буде накладатись.

Хоча сьогодні цей метод є дуже популярним у кінематографі та розробці ігор, він все ще потребує коштовного обладнання, акторів та спеціалістів для створення фінального результату. Таким методом можна генерувати дуже якісний контент, що здатен ввести в оману, проте для використання в режимі реального часу потрібне потужне і коштовне обладнання.

Deepfake – технологія синтезу зображення людини, що використовується для поєднання і накладення існуючих зображень та відео на початкові зображення або відеоролики. Створення Deepfake стало можливим завдяки еволюції в дослідженні генеративно-змагальних нейронних мереж. Генеративно-змагальна мережа – це алгоритм штучного інтелекту для навчання без учителя, який реалізований системою з двох нейронних мереж які змагаються між собою. Одна мережа тренує кандидатів (генератор), інша оцінює їх (дискримінатор). Отже, генератор навчається «обманювати» дискримінатор, намагаючись згенерувати максимально правдоподібне с точки зору людини зображення.

технологія, яка не має недоліків попередніх прикладів, а саме необхідність дорогого обладнання, спеціалістів та акторів, а також може працювати в режимі реального часу. Звісно, у неї також є недоліки, але технологія ще розвивається, тож в недалекому майбутньому згенерований контент у правомірному використанні може стати нормою. Досягти такої якості стало можливо з розвитком штучного інтелекту і появі потужних домашніх графічних прискорювачів, на яких і проходить обробка. Для використання технології Deepfake достатньо мати потужний домашній комп’ютер, що значно дешевше за професійне обладнання для захоплення рухів. Спеціальне програмне забезпечення, яке дозволяє використовувати Deepfake в режимі реального часу не потребує знань програмування, і також дозволяє створювати власні моделі для заміни конкретних обличь. За наявності потужного графічного прискорювача останніх поколінь і даних про цільове обличчя, створення нової моделі може зайняти декілька діб. На виході отримуємо нейронну мережу, здатну замінити обличчя у будь-якому фото- відео контенті,

а також здатній робити це в режимі реального часу. Таким чином зловмисник може генерувати контент з обличчям жертви в будь-яких умовах. Це стало можливим з розвитком генеративно-змагальних нейронних мереж.

Наразі доступно декілька програм для заміни обличчя за допомогою штучного інтелекту. До відомих можна віднести FaceApp [62], і вона доступна для мобільних пристроїв. Проте більшість доступних програм для мобільних платформ (iOS, Android) можуть змінювати обличчя на обличчя користувача лише на обмеженому наборі прикладів відео, які зазвичай взяті з відомих фільмів, серіалів або іншого відомого відеоконтенту. Також, зазвичай, на відео наноситься емблема застосунку. Це створено для рекламних цілей, а не для безпеки, тому що такі діпфейки зазвичай використовують звичайні користувачі в особистих цілях, для спілкування з друзями. Доступність таких програм підвищує усвідомленість користувачів про технології генерації контенту.

1.3 Приклади використання Deepfake.

Найперша згадка про використання Deepfake контенту датується 2017 роком, коли на форумі Reddit було опубліковано перший порнографічний ролик з використанням заміни обличчя. Далі було опубліковано ще декілька роликів з відомими кінозірками, всі вони були порнографічного характеру. Всі ролики було швидко спростовано, проте це викликало резонанс в засобах масової інформації. Якість перших роликів була низькою, тому ввести в оману за допомогою такого контенту було досить складно. Проте надалі якість Deepfake контенту тільки росла. Тож, перше застосування – генерація порнографічного контенту для шантажу, продажу або інших цілей. Такий контент може завдати як репутаційних втрат, так і грошових. Хоча для визначення згенерованого контенту жертва може знайти оригінал, і таким чином довести згенерованість контенту. Інша можлива сфера застосування – політичний контент. Так, можливо використовувати технологію для генерування контенту, який покаже що певний політик доносить певну інформацію, яку в реальному житті він ніколи не казав. Можливий сценарій також включає використання згенерованого контенту в рекламних цілей різних шахрайських проектів за допомогою відомих особистостей, наприклад, реклама «МММ» згенерованим померлим С. Мавроді. Такий контент при правильному використанні може мати значний вплив на людей.

Deepfake можна використовувати також для оформлення кредитів, отримання інших послуг чи продуктів, коли спілкування з людиною відбувається віддалено, за допомогою різного програмного забезпечення (Skype, Zoom, Teams). Через високу якість оброки обличчя в реальному часі, іншу людину просто ввести в оману, аби вона повірила шахраям, які використали технологію. Також, окрема модель може імітувати навіть голос жертви, що ще більше вводить в оману оператора або іншу людину, з якою спілкуються шахраї, що додатково підвищить ефективність атаки. Наприклад, шахраї можуть згенерувати модель керівника компанії і зателефонувати співробітникам для отримання доступів, виконання вказівок або інших неправомірних дій. Також, якщо потрібно зробити фотографію з документом для оформлення кредитів в мікро кредитних організаціях можна використовувати Deepfake для заміни обличчя жертви на фотографії з шахраєм та підробленим документом, знайти який в мережі Інтернет не є проблемою. Хоча в цьому випадку зловмисникам потрібні базові знання у використанні фоторедакторів для редагування документу.

Звісно, технологія використовується не лише в поганих цілях. Наприклад, перспективне застосування deepfake відео в сфері реклами. При виборі актору для рекламного ролику важливим є його впізнаваність. У випадку залучення відомих людей більшу частину витрат на ролик становитиме його гонорар. Також, при створенні реклами велика кількість бюджету витрачається на переклад та дубляж роликів на інших мовах. В цьому випадку алгоритми Deepfake можуть значно зменшити вартість створення рекламних роликів, адже відомі люди можуть офіційно дати дозвіл на застосування свого обличчя та інших біометричних даних в рекламних цілях, що буде значно дешевше їх особистої присутності на знімальному майданчику. До того ж, технологія дозволить використовувати біометрію вже померлих людей, звісно, при отриманні відповідних дозволів у родичів.

Роботи та віртуальні консультанти в майбутньому також потребуватимуть застосування алгоритмів штучного інтелекту. На корейському телебаченні вже використовуються віртуальні ведучі новин. Наступним кроком стане створення віртуальних консультантів та асистентів з фото реалістичними обличчями та мімікою. Наразі цифрові асистенти вміють працювати лише з текстовими та голосовими даними, і відповідати також текстом або голосом. Та в майбутньому будуть доступні віртуальні асистенти, що схожі на реальних

людей, копіюють їх міміку, жести та голоси. Використання Deepfake буде корисним також при створенні роботів-андроїдів, для створення більш реалістичних рухів обличчя. Але наразі віртуальні асистенти відповідають лише в текстовому форматі і також із застосуванням згенерованого голосу, який, проте далекий від людського. Тож першим покращенням віртуальних асистентів з застосуванням Deepfake може стати корекція віртуальних голосів і їх персоналізація для кожного користувача, враховуючи його побажання.

Індустрія ігор також може застосовувати цю технологію, хоча на даний момент її використовують лише в деяких сценах ігор, в майбутньому можна повністю створювати персонажів за допомогою deepfake. Також, з розвитком індустрії доповнених реальностей технологія також покращить досвід користувачів ігор. Створення реалістичних віртуальних аватарів за допомогою простих методів значно прискорить розвиток ігор, а також зменшить вартість їх розробки. Це саме стосується і виробництва фільмів. Замість гриму або масок дешевше та швидше використання алгоритмів deepfake. Також, при виробництві фільмів можливо скорегувати певні помилки та артефакти, які виникають при створенні deepfake, адже перед випуском фільму в кінотеатри відео проходить процес рендеру, тобто покращення та корекції кадрів. Зараз кіноіндустрія мало використовує цю технологію, проте в майбутньому це може стати однією з основних технологій створення кіно. Наразі, існує певний симбіоз ігрових технологій та технологій кінематографу. Ігрові рушії вже активно використовують в кінематографі через їх простоту та низьку вартість, а також можливість створити будь-яку сцену або фон. Таким чином, виробники фільмів отримують відносно дешеву можливість знімати все в студії, на фоні створеного ігровим рушієм фону або інших частин фільму, не витрачаючи час та ресурси на поїздки в реальні місця. Можна очікувати, що розвиток алгоритмів генерації надасть можливість їх використання в більшості випадків створення відеоконтенту.

За допомогою генеративних мереж можна створити фотографію людини, яка ніколи не існувала. Сфера застосування таких фотографій не дуже велика – інтернет-шахраї, реклама, презентації. Частіше всього фотографії неіснуючих людей можна зустріти на ресурсах знайомств, персональних сторінках в соціальних мережах. Зловмисники за допомогою соціальної інженерії та під видом молодої дівчини можуть використовувати такі профілі для втирання в довіру та отримання зиску з інших людей. При перевірці фотографії такого

профілю в пошукових системах за допомогою функції пошуку за зображеннями знайти власника фотографії не вдасться, при використанні фото реальної особи, жертва може скористатись такою функцією і зрозуміти, що спілкується зі зловмисником та припинити спілкування. Або ж під видом фотографії консультанта в сервісах онлайн-підтримки відповідати на запитання користувачів та допомагати користувачам виконувати дії, що можуть завдати збитків користувачам. Також, за допомогою штучного створення голосу можна створити додаткову довіру жертви до фейкового профілю, що значно спростить зловмисникам роботу. Застосування комбінації алгоритмів штучного інтелекту для створення фотографії, підміни голосу та анімації фотографії може остаточно прибрати сумніви жертви, та змусити, наприклад, перевести кошти на рахунок шахраїв.

Використання фотографій людей для реклами може стати в нагоді компаніям, які не хочуть платити за використання біометрії інших людей, або їх фотографій. Таким чином, згенероване обличчя можна використовувати в логотипах будь-яких продуктів безкоштовно.

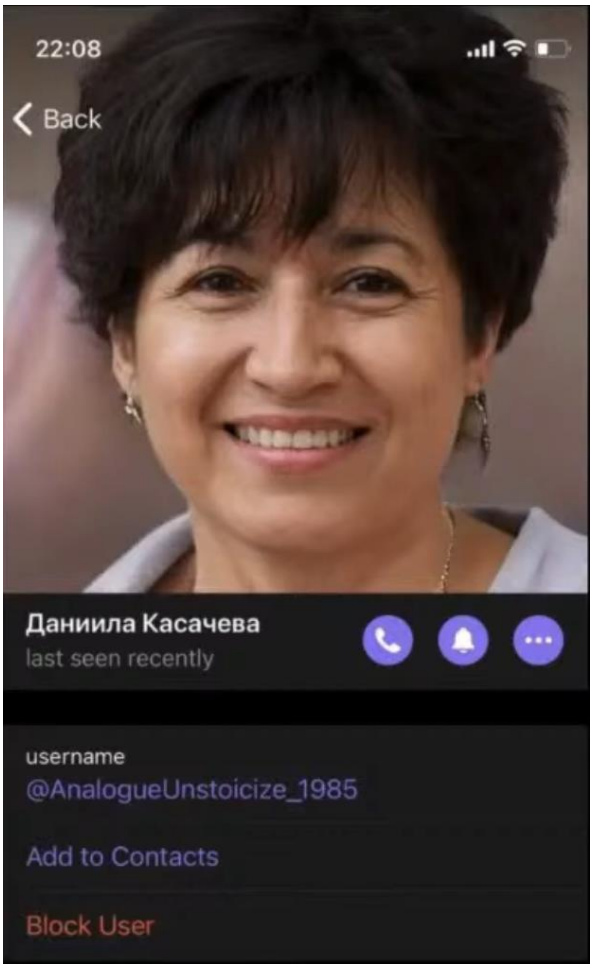

Рисунок 1. Приклад штучної фотографії профілю в соціальній мережі

Одним з найпопулярніших застосувань Deepfake алгоритмів є проходження верифікації на різноманітних платформах. Більшість платформ для роботи з криптовалютами потребують проходження верифікації з використанням алгоритмів, які перевіряють документи людини та потребують визначення того, що це жива людина, а не комп'ютерна програма. Також такі платформи не дозволяють створювати багато облікових записів на одну людину, тому була введена автоматична перевірка документів та зображення в реальному часі власника цих документів. Хоча більшість подібних сервісів не перевіряє справжність документів з державними базами, а фотографії документів звіряють в автоматичному режимі з відповідними шаблонами, а також перевіряють введену інформацію. Така перевірка розрахована більше на захист самих платформ у випадках, коли правоохоронцями можуть висуватись претензії щодо шляху отримання коштів, які користувачі вводять на платформи, або інших можливих протиправних дій користувачів на цих платформах. В такому випадку платформа може видати по запиту правоохоронних органів історію транзакцій, входів на платформу, адрес і сканів документів, які користувачі самі віддали на платформу.

Типовий алгоритм проходження верифікації на платформі може складатись з таких кроків:

Отримання документів. Документи людини можна придбати в «глибокому Інтернеті» за відносно невеликі кошти. Існує безліч сервісів які пропонують користувачам купити отримані протиправним чином документи, записи з державних баз даних, або копії самих баз. Іноді старі версії державних баз даних можна завантажити безкоштовно. Іноді навіть самі користувачі можуть викладати свої персональні документи в мережу, з якої ці дані вже ніколи не зникнуть. Але більш ідеальним варіантом даних для проходження верифікації є селфі користувачів, які тримають розгорнутий паспорт поряд з обличчям. В такому випадку зловмисники матимуть як фотографію паспорта, так і фотографію користувача в природніх умовах. Такі дані отримати складніше, адже вони можуть міститись лише в базах даних фінансових установ з мікрокредитування, які дозволяють оформити кредит після перевірки фотографії. Але такі установи зазвичай мають низьку інформаційну безпеку і часто стають об'єктами атаки хакерів, або ж навіть самі співробітники можуть продавати такі дані. У випадку, коли зловмисник має велику базу фотографій документів, і потребує обличчя їх власників в реальних умовах можна провести

розвідку за відкритими джерелами у соціальних мережах і зібрати велику кількість обличь. Застосування обличчя не з фотографії паспорту є більш ефективним через більшу якість додаткового зображення, а також більш природнім виразом обличчя. Іноді зловмисники за допомогою фоторедактору можуть створювати і неіснуючі документи з фотографією людини для якої вже мають модель Deepfake.

Створення Deepfake моделі. Є два способи створити модель обличчя для проходження верифікації. В першому способі потрібно натренувати нейронну мережу на обличчі конкретної людини. Після тренування вона буде здатна в режимі реального часу змінити будь-яке обличчя на цільове. Для цього потрібен набір фотографій або відеозаписів, з яких програма автоматично вибере обличчя та почне тренування нейронної мережі. ДО плюсів такого методу можна віднести добру якість вихідного зображення. Нейронна мережа здатна змінювати обличчя в різних позах голови, що знадобиться, якщо верифікація потребуватиме виконання певних рухів головою. До недоліків методу відноситься необхідність в наборі даних та високі вимоги до заліза і час на створення моделі. Також від якості тренувальних даних буде залежатиме фінальний результат роботи, тому ідеально створити такий набір, де цільове обличчя буде знаходитись в різних позах, при різному освітленні, тобто збирання різноманітних фото та відео матеріалів з обличчям. Якщо зловмисники знайдуть профіль людини в соціальних мережах, то створення набору даних не стане проблемою. Проте вимоги до заліза та час на створення моделі все ще заважають широкому розповсюдженню цього методу. В середньому, тренування моделі для одного обличчя може тривати близько доби на останніх моделях графічних прискорювачів. Цих недоліків не має інший метод, при якому достатньо лише фотографії обличчя. Зловмисник переносить рух свого обличчя з веб-камери на цільове обличчя, тим самим анімуючи фотографію відповідно до своїх рухів. Тоді йому не потрібно тренувати модель, і вимоги до графічних прискорювачів нижче за попередній метод. Але в такому разі важко виконувати складні рухи головою, адже алгоритм анімації не має даних про зображення, і намагається домалювати його простими методами. В таких випадках зазвичай з’являється велика кількість артефактів на зображенні, тому легко зрозуміти, що картинка не справжня.

Проходження верифікації. Для успішного проходження верифікації на таких платформах потрібно завантажити або зробити фотографію документів, і

далі за допомогою веб-камери або смартфону пройти перевірку обличчя. Іноді під час перевірки потрібно виконати певні дії обличчям, наприклад рух голови вправо, вліво, вниз. Це створено для розпізнавання того, що перед нами жива людина, а не фотографія. Також зловмисники створюють віртуальну веб-камеру, яка і являє собою змінене обличчя і його рухи. Реальну веб-камеру застосовують для отримання даних про рух голови самого зловмисника. Зазвичай, документи використовують для отримання даних про обличчя, які потім порівнюють з даними веб-камери. Тобто, очевидно, що людина не пройде верифікацію з чужими документами. Якщо система не змогла розпізнати Deepfake то верифікація проходить успішно. Далі такі акаунти можна використовувати в різноманітних власних цілях, або продавати іншим. Попит на такі акаунти досить високий, що означає, що люди які їх створюють непогано заробляють. Також це означає, що користувачі таких акаунтів теж мають з них добрий зиск.

На сьогодні існує небагато систем автоматичної верифікації, стійких до Deepfake. Найпростішим захистом може бути визначення віртуальної камери. В такому випадку можна припустити, що з високою ймовірністю хтось змінює відеопотік з оригінальної камери, та не дозволити проходити верифікацію до зміни камери. Також непоганим рішенням може стати проходження верифікації за допомогою смартфонів. Більшість з них мають дві камери, дані з яких отримує система верифікації. Також, замінити відеопотік на смартфоні значно складніше, і вимагатиме кваліфікованих спеціалістів і значних ресурсів. Тобто, такий метод може стати не рентабельним. Також варто зазначити, що обчислювальної потужності смартфону не вистачить для генерації відео в реальному часі.

Все це дає підстави вважати, що створення алгоритмів визначення Deepfake не стоїть на місці і має реальний попит і реальні сфери застосування. Хоча зараз визначення таких відео покладено на більш звичні аналізатори, які працюють на основі даних про пристрій, параметрів відео та інших даних, що ніяк не залежать від того, що показане на відеозаписі. Проте такі системи все ж можна обійти, тому в майбутньому буде необхідно також аналізувати інформацію з відеозаписів додатково до технічного аналізу засобів, з яких цей відеозапис надходить.

Також потрібно боротись з витоками персональних даних. В «глибокому Інтернеті» можна знайти повний комплект документів випадкової людини, і

використовувати його в різних цілях. Якщо раніше за допомогою таких документів та підставної особи в банку можна було взяти кредит на людину, яка дізналася про нього лише після нарахування значних відсотків, то зараз не потрібна навіть підставний працівник банку. Достатньо програмного забезпечення, яке може дозволити зловмисникам не виходячи з дому використовувати документи людини для отримання кредитів в багатьох банках.

Сфера продуктів, яка зараз набуває популярності – нові компанії, що створюють цифрові копії людей, які померли та дають можливість їх родичам спілкуватись з цифровою копією людини. Для цього потрібно надати компанії фотографії, відео та переписки з померлим. Тоді штучний інтелект може створити візуалізацію людини, а інший – створити емуляцію чат-боту, який може підлаштуватись під діалоги людини. Це може полегшити людині переносити втрату близьких, проте етичне питання виходить за рамки цієї роботи.

Відносно нове застосування, яку придумали інтернет-шахраї з Індії досить не нова, але використання сучасних алгоритмів лише полегшила шахраям їх роботу. Вони використовують сервіси інтернет-знайомств, в яких видають себе за жінок, які знайомляться з чоловіками. Так, цей метод досить не новий, і виник напевне трохи пізніше за такі сервіси. До цього шахраї також використовували сервіси інтернет-знайомств для отримання вигоди. Використання соціальної інженерії при спілкуванні з потенційною жертвою для того, щоб змусити її наприклад, ввести дані кредитної карти на фішинговому сайті з купівлі квитків в кіно або театр, купівля квітів. Алгоритми deepfake дають шахраям ще більше можливостей для отримання вигоди з довірливих користувачів. Зловмисники з Індії використовували генерацію обличь при спілкуванні з жертвами. Так, спілкуючись з «реальною» людиною, обличчя якої було змінене зловмисникам простіше використовувати соціальну інженерію для отримання вигоди. За даними новин, жертвами такої маніпуляції вже стали кілька десятків чоловіків. Під час спілкування з зловмисником, жертва могла не усвідомлювати, що певні дії в особистому спілкуванні можуть її дискримінувати. Цим і користувались шахраї – заставляли жертву робити певні дискримінуючи її дії і за допомогою запису цих подій далі шантажували жертву розповсюдженням цього матеріалу серед друзів і родичів жертви. Тому більшість постраждалих від таких дій переводили зловмисникам кошти щоб зберегти власну репутацію.

Список можливих використання Deepfake, який описаний вище не вичерпний, і скоріше всього сфери їх використання значно ширша, а в майбутньому буде розширюватись. Розвиток алгоритмів також може допомогти дослідникам в інших сферах. Розвиток регулювання в цій сфері відстає від алгоритмів, проте все більше державних діячів та регуляторних органів розуміє небезпеки, що виходять від нових технологій, і починають створювати законодавство, яке регулює використання технологій deepfake в нашому житті. Створюються закони, які забороняють використання алгоритмів для дискримінації або інших дій, які можуть завдати репутаційні збитки, створюються стандарти безпеки, які дозволяють розробникам біометричних систем авторизації та аутентифікації враховувати можливі небезпеки від застосування новітніх технологій. Перші випадки отримання реальних покарань за використання вже є. Так, в США суд засудив до реального терміну жінку, що створила порнографічне deepfake-відео за участі однокласниці своєї доньки та розповсюдила його серед інших. Суд постановив, що вона створила це усвідомлено та для цілеспрямованого завдання репутаційних втрат та приниження гідності. Хоча це радше виключення, адже більшість авторів порнографічних роликів з обличчями відомих людей не несуть жодного покарання через складність їх визначення в мережі.

Створення просунутих алгоритмів розпізнавання згенерованого контенту теж набирає оберти. Люди зрозуміли, що подібні алгоритми можуть нести реальну загрозу, не дивлячись на ще більші сфери застосування, в яких вони можуть спростити роботу спеціалістів різних сфер. Дослідники займаються розвитком та поліпшенням якості генерації, і тому з кожним роком можна бачити великий прогрес в генерації зображень та звуків.

Рисунок 2. Покращення генерації обличчя

Також дослідники працюють над алгоритмами розпізнавання, проте для їх роботи також важливе правильне застосування, при якому вони будуть максимально зручними для користувачів та розробників. Користувачі повинні мати максимально простий інтерфейс програми розпізнавання, можливість перевірити контент з будь-якого джерела з мінімальною кількістю дій. Розробники в свою чергу мають мати прості інструменти для додавання підтримки алгоритмів розпізнавання в свої продукти для мінімізації вартості та часу розробки продуктів. Найбільш ймовірно, що першими запустять подібні алгоритми найбільші Інтернет компанії, які дають змогу користувачам створювати та переглядати контент. Такими компаніями можна вважати YouTube, Facebook, Twitter. Саме ці компанії мають домінуюче місце в рекламі політики, продуктів та звичайного спілкування між користувачами, тому очікувано, що інформаційний вплив діпфейків на цих платформах буде найбільшим. Враховуючи те, що на цих платформах вже є велика кількість неправдивої інформації (шкода 5G, теорії заговорів, противники вакцинації) – це свідчення про те, що користувачі цих платформ рідко перевіряють факти в декількох джерелах, та схильні вірити всьому, що побачать. Незважаючи на наявність людської та автоматичної модерації навіть після видалення контенту його швидко відновлюють та поширюють. В такому випадку потрібно додатково розробляти алгоритми покарань за розповсюдження сумнівного контенту, який може бути використано для маніпуляції. Цей випадок також можна вирішити за допомогою штучного інтелекту.

Штучний інтелект все більше проникає в наше повсякденне життя. Більшість алгоритмів значно його спрощують, але як ми показали, існують і небезпечні алгоритми. Хоча, скоріше всього небезпечними їх робить неправильне використання людьми. Все ще не існує органів регулювання використання алгоритмів штучного інтелекту, але початкові кроки в цьому напрямку також з’являються. Явна користь від використання штучного інтелекту в сферах охорони здоров’я де він допомагає лікарям діагностувати хвороби, в сфері безпеки, де алгоритми допомагають запобігати жертвам на виробництвах, інформаційній сфері, де цифрові асистенти вмикають музику, роблять покупки, прокладають маршрути да допомагають з вибором фільму ввечері. Все це може зіпсувати шкода при неправильному використанні таких алгоритмів. Наприклад, використання розумних камер для стеження у випадках що непередбачені законодавством є вторгненням в особисте життя.

Застосування методів атаки на системи зі штучним інтелектом може порушити їх правильну роботу, від якої також можуть залежати цінні активи, ресурси або навіть людські життя. Все це може викликати занепокоєння, проте це відносно нова технологія, і сфери її використання ще не повністю явні, як і наслідки від такого використання.

Віддалено, але до Deepfake також можна відносити розумних діалогових ботів. Але не як самостійну технологію, а скоріше як додаток для зображення або відео. Такі системи можуть взаємодіяти з користувачем автоматично, без втручання оператора, що значно збільшує як сферу їх застосування так і зменшує необхідність наявності оператора. Наприклад, у випадку спілкування в мережах для знайомств шахраї з метою отримання зиску з жертви можуть генерувати фотографії, відео повідомлення від неіснуючої людини, а за допомогою діалогової системи вести просту переписку в автоматичному режимі. Але все ще людина необхідна для соціальної інженерії, метою якої і є використання подібних систем.

Висновки до розділу 1

В цьому розділі було описано історія виникнення технології генерації контенту за допомогою штучного інтелекту. Потенційні шляхи використання технології, реальні приклади сфери де технологія вже використовується. Стає зрозуміло, що неконтрольоване використання нових технологій несе значну небезпеку в інформаційному просторі. Широкий спектр напрямків неправомірного використання може завдати значні збитки політичним строю держави, репутаційних збитків відомим людям і матеріальних збитків звичайним користувачам мережі Інтернет. Тому зараз розвиток алгоритмів визначення починає наздоганяти в якості алгоритми генерації, які теж не стоять на місці. А державні органи починають усвідомлювати небезпеку від подібних технологій, і починають створювати закони та норми регулювання подібних алгоритмів

РОЗДІЛ 2 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У ФОРМАТІ «DEEP FAKE»

2.1. Основні та допоміжні алгоритми створення Deepfake.

Для створення Deepfake алгоритмам необхідно, щоб алгоритми спочатку знайшли обличьчя на фотографії або відео, потім знайти ключові точки обличчя, і виділити основну інформацію про обличчя. Задача розпізнавання обличь є однією з перших задач в області комп'ютерного розпізнавання образів, і найбільшим чином стимулювала розвиток теорії комп'ютерного розпізнавання образів. До систем розпізнавання обличь завжди був значним, особливо зі зростаючими практичними потребами: охоронні системи, верифікація, криміналістика, безпека. Не дивлячись на простоту розпізнавання обличь для людини, досить не очевидним залишався процес створення комп'ютерного алгоритму для автоматичної роботи в режимі реального часу. Ще більшу складність викликає оцінка схожості двох обличь. Для розв'язання цієї задачі за допомогою алгоритмів її можна розділити на дві підзадачі:

Детекція обличчя – алгоритм, що має з високою точністю прийняти на вхід зображення і повернути координати обличь що там знаходяться. Також додатково алгоритми можуть повертати свою оцінку у впевненості, що за цими координатами на зображенні справді знаходиться обличчя.

Кодування обличчя – процес виділення інформації про конкретне обличчя для подальшого розпізнавання або порівняння. Алгоритм має кодувати обличчя однозначно і зручно для подальшого використання з метою обчислення відстані між двома обличчями.

Перші роботи почались в 1960-х роках. Більша частина методів працює з двомірними зображеннями через їх широке використання і доступне обладнання. Існують алгоритми, що працюють з тривимірними даними обличь, проте для роботи вони частіше всього потребують спеціального обладнання для тривимірного сканування обличь, і сьогодні такі приклади найактивніше використовує система FaceID за допомогою лазерних датчиків, що створюють тривимірну копію обличчя користувача. Такий метод більш надійний ніж двомірне розпізнавання, адже до системи надходить значно більше даних про обличь користувача. Основний недолік такого методу – необхідність додаткового обладнання для сканування обличчя.

Перші спроби використовували напіватоматичний підхід у розпізнавання обличь, використовуючи введені оператором ключові точки обличчя (ніс, рот,

очі) і далі автоматично відстежуючи їх. Комп’ютер також автоматично рахував відстані між точками, після чого відбувалось порівняння характеристик між обличчями для ідентифікації. Задача полягала в створенні бази даних зображень, з якої потрібно було вибрати найбільш відповідний запис. База створювалась у вигляді відповідностей між іменем людини на вирахуваним списком параметрів відстаней між ключовими точками. Для розпізнавання множину відстаней порівнювали з відповідною множиною відстаней для кожної фотографії, на основі цього обчислювалась міра відмінності між фотографією та базою даних.

Підхід мав очевидні недоліки – для обробки великої бази обличь потрібна ручна розмітка оператором, спеціальне обладнання для кожного оператора та необхідні знання для коректної розмітки. Що не дозволяло широко використовувати систему. Метод мав низьку точність при порівнянні навіть обличь однієї людини з поворотом голови, і не працював в реальному часі.

Перший детектор обличь, що працював в реальному часі був створений на основі «признаків Хаара». Признаки Хаара – ознаки цифрового зображення, що використовуються в комп’ютерному розпізнаванні образів [33]. Вперше цей алгоритм було адаптовано Віолою і Джносном на основі вейвлетів Хаара для розпізнавання образів. Ідея методу полягає в тому, щоб за допомогою сукупності простих фільтрів знайти закономірності в зображеннях. Признак Хаару складається з суміжних прямокутних областей. Вони позиціюються на зображенні, і далі додаються інтенсивності пікселів в областях, після чого обчислюється різниця між сумами. Ця різниця і буде значенням певного признаку певного розміру, певним чином позиційована на зображенні [34].

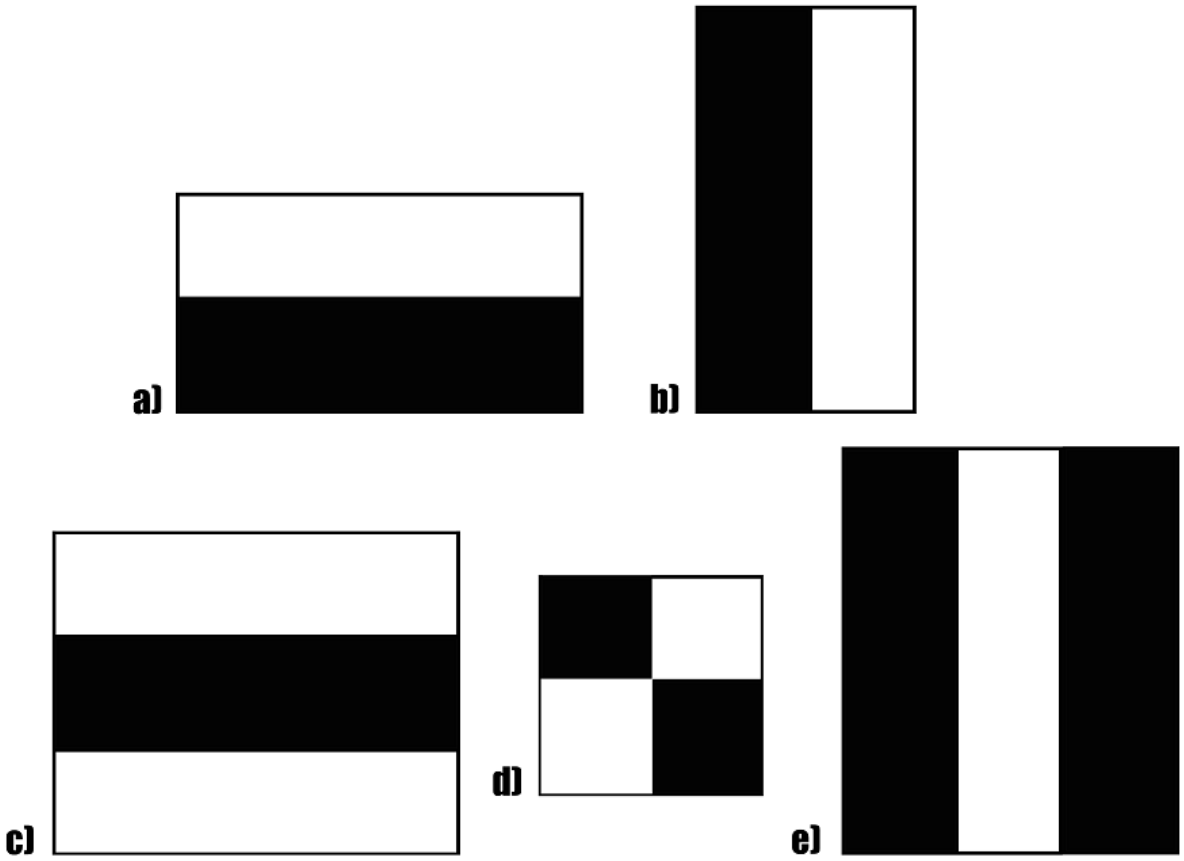

Рисунок 3. Приклади фільтрів ознак Хаару для детекції

Для прикладу з обличчями, спільним для всіх зображень буде те, що область в районі очей темніша, ніж на щоках. Таким чином, спільним признаками Хаару для обличь буде два суміжних прямокутних регіони, що знаходяться на очах і щоках. Звісно, для більшої точності алгоритму використовується значно більша кількість ознак. На етапі визначення в методі Віоли-Джонса вікно встановленого розміру рухається по зображенню, і для кожної області зображення розраховується признак Хаару. Наявність або відсутність предмету у вікні визначається різницею між розрахованою ознакою і порогом навчання. Через те, що одинична ознака Хаару не підходить для навчання через низьку якість, близьку до випадково розподіленої величини, для опису об’єкту потрібна більша кількість ознак. Тому в методі Віоли-Джонса признаки Хаару організовані в каскадний класифікатор. Все це дозволяє методу працювати в режимі реального часу. Існує декілька основних ознак Хаару. До них відносять прямокутні та похилі ознаки.

Прямокутні ознаки Хаару. Найпростішу ознаку Хаара можна визначити як різність сум пікселів двох суміжних областей всередині прямокутника, що може займати різні положення і масштаби на зображенні. Кожна ознака може

показати наявність чи відсутність конкретної характеристики зображення, такої як межі або зміни структур.

Похилі ознаки Хаару створені для збільшення розмірності простору признаків, і відрізняються від прямокутних лише кутом нахилу фільтру. Проте через більш складну обчислювальну вартість таких ознак детектор використовує зображення низької обчислювальної здатності, що призводить до помилки округлення, тому такі ознаки не використовуються на практиці.

Розвиток нейронних мереж та збільшення обчислювальної потужності комп'ютерних систем зробив можливим використання їх в завданнях з розпізнавання образів в реальному часі навіть на мобільних платформах. Нейронні мережі використовуються як для знаходження обличь на фото, так і для кодування вже знайдених обличь. Підхід з використанням нейронних мереж дозволяє алгоритмам працювати в більш різноманітних умовах, таких як більші кути нахилу голови, погане світло, частково закриті обличчя, менша площа обличчя на фото. Згорткові нейронні мережі використовуються в широкому спектрі задач комп'ютерного зору, і в деяких типах завдань, наприклад таких як розрізнення обличь вони краще за людську оцінку.

Згорткова нейронна мережа (ЗНМ, convolutional neural network, CNN, ConvNet) – клас глибинних штучних нейронних мереж прямого поширення, які частіше всього використовуються в обробці зображень. ЗМН взяли за основу біологічний процес об'єднання нейронів зорової кори. Перші роботи про дослідження зорової кори були опубліковані в 1950-х та 1960-х роках. Перший алгоритм – неокогнітрон було представлено в 1980-х роках. Неокогнітрон - згорткова ієрархічна багатошарова штучна нейронна мережа, заснована на принципах навчання без учителя. Алгоритм пристосований для розпізнання візуальних даних. Неокогнітрон складається з чергуючись простих та складних шарів. Кожний шар містить певному кількість площин нейронів, та одну площину гальмуючих нейронів. Прості нейрони виконують функцію розпізнання образів, складні нейрони виконують функцію неоднорідного периферійного розмивання. В процесі навчання змінюються коефіцієнти для кожного нейрону. Подальший розвиток неокогнітрону – ЗМН, перша версія якої була запропонована у 1998 році, і була створена для розпізнавання рукописних цифр. LeNet-5 – перша ЗМН, яка була застосована в реальних завданнях для розпізнавання цифр. З часом ЗМН тільки розвивались, і наразі

використовуються в широкому спектрі задач комп'ютерного зору для розпізнавання, класифікації, штучного генерування зображень.

Згорткова нейронна мережа схожа на більш простий тип нейронних мереж – звичайний перцептрон, або повнозвязна нейронна мережа. Вона також складається з нейронів, які містять в собі ваги. Кожний нейрон отримує деякі вхідні дані, розраховує скалярний добуток, та може використовувати нелінійну функцію активації. Вся нейронна мережа представляє собою диференційовану функцію оцінки – з кожного набору пікселів на одному кінці до розподілу ймовірностей приналежності до певного класу на виході з мережі. У всі ШНМ є функції втрат на останньому повнозвязному шарі, що використовується для класифікації. На відміну від звичайної повнозвязної нейронної мережі згорткова нейронна мережа добре масштабується для великих зображень. Через те, що зазвичай зображення мають розміри n*n*3, де n – кількість пікселів по горизонталі та вертикалі, 3 – це кількість каналів звичайного RGB зображення, то перший нейрон матиме n*n*3 вагів. Для прикладу, якщо взяти зображення розміром у 200*200*3, то лише перший шар нейронної мережі матиме 120000 вагів. Для правильної роботи потрібен не один прихований шар, тому число загальних вагів в такій мережі буде дуже високим, що призведе до надмірних обчислювальних витрат для навчання та роботи.

Нейронні мережі, в тому числі згорткові складаються з певної кількості універсальних блоків, що відрізняються параметрами, кількістю та розтушуванням в архітектурі нейронної мережі, що дозволяє вирішувати нейронним мережам різні задачі.

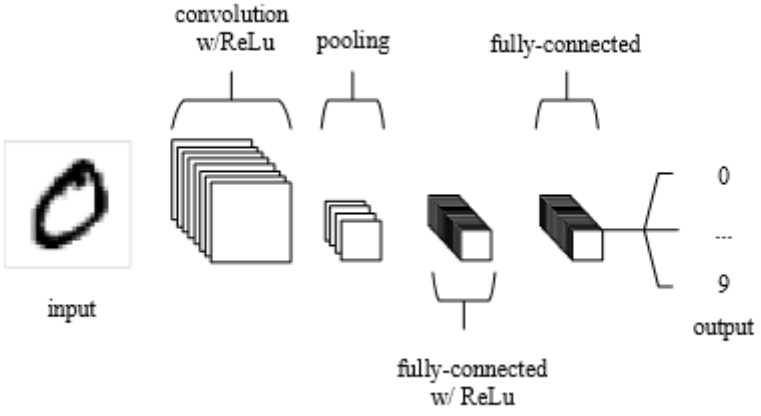

Рисунок 4. Приклад роботи алгоритму ЗГН

Основний блок ЗМН – згортковий шар. Згортковий шар є основним будівельним блоком ЗМН. На вхід він приймає n-вимірний квадратний тензор

пікселів зображення. Далі виконується основна математична операція згорткового шару – згортка. Згортка за своїм процесом схожа на фільтри Хаару, що використовувались в перших системах розпізнавання обличчя. Окремі нейрони реагують на стимули лише в обмеженій області зорового поля. Зорові поля декількох нейронів частково перетинаються. ЗМН складаються з шарів входу, виходу та прихованих шарів. Приховані шари згорткової нейронної мережі складаються зі згорткових шарів, агрегувальних шляхів, повнозвязних шарів, шарів нормалізації. Згорткові шари застосовують операції згортки, яка імітує реакцію окремого нейрону на зоровий імпульс. Кожен згортковий нейрон обробляє дані лише свого рецептивного поля. Параметри згорткового шару складаються з набору фільтрів, здатних до навчання. Кожен фільтр являє собою невелику сітку вздовж ширини та висоти, проте працюючих по всій глибині вхідного представлення зображення. Фільтр ЗМН матиме розмір n*n*3 (n-пікселів ширина та висота, 3 – кількість каналів, n – непарне число), наприклад для першого шару це може бути 5*5*3 або 3*3*3. Під час першого проходу ми переміщаємо фільтр вздовж ширини і висоти вхідного представлення даних і розраховуємо скалярний добуток між значенням фільтру та значеннями вхідного представлення в будь-якій точці. В процесі переміщення фільтру вздовж ширини та довжини вхідних даних ми отримуємо двомірну карту ознак активації, яка містить значення застосування даного фільтру до кожної з областей вхідного представлення. Нейронна мережа має навчити фільтри активуватись при знаходженні певної візуальної ознаки на більш високих рівнях.

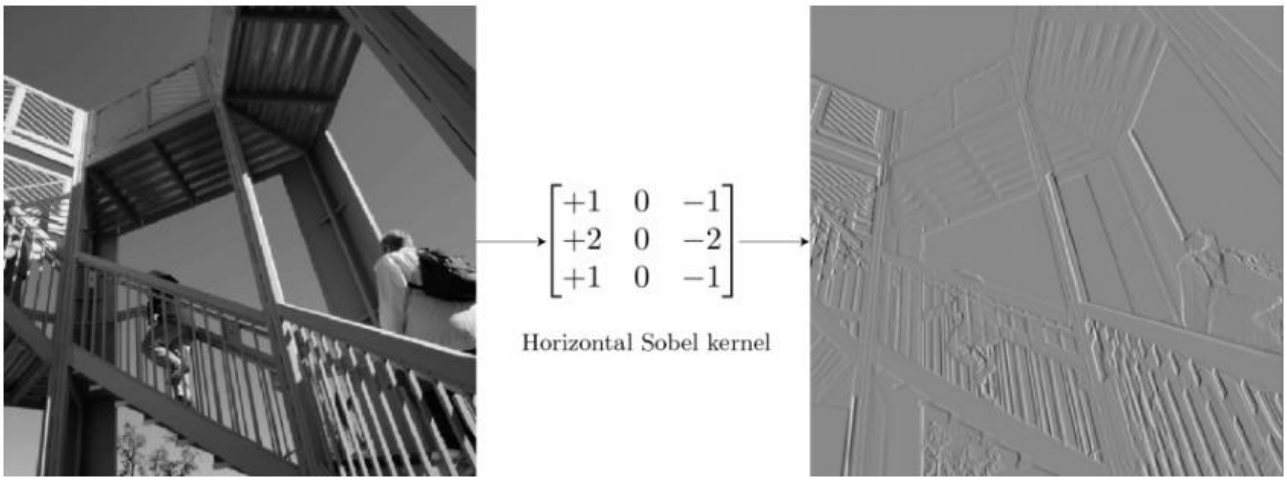

Рисунок 5. Зображення після проходження фільтру для знаходження горизонтальних ліній.

Потім потрібно об’єднати двомірні карти ознак по третьому вимірі. Існує декілька параметрів, які можна налаштувати перед навчанням, і які можуть змінити кінцеву якість моделі після навчання. Перше, що можна налаштувати – розмір фільтру в кожному згортковому шарі. Розмір фільтру має бути не праним, частіше всього обирають числа від 3 до 7 пікселів, який буде зменшуватись після певної кількості шарів в архітектурі. Більший фільтр буде захоплювати більше ознак, що корисно для перших шарів ЗМН для пошуку загальних візуальних ознак, і ближче до кінця корисніше шукати менші візуальні патерни. Далі можна регулювати відстань, яку буде проходити фільтр по зображенню перед розрахунком скалярного добутку. Частіше всього корисно рухатись не пропускаючи сусідні пікселі, проте іноді є архітектури, в яких згортка застосовується з пропуском деякої кількості сусідніх пікселів. До цього також можна віднести техніку, коли для розрахунку скалярного добутку ми значно збільшуємо розмір фільтру, але для розрахунку використовуємо окремі пікселі, що не мають сусідів, до яких застосовується скалярний добуток. Часто використовувана техніка padding – для кращої обробки меж зображень додавати порожню область пікселів до зображення, таким чином збільшуючи вхідне зображення на декілька пікселів, щоб ядро при обрахуванні перших ознак могло краще захопити ознаки на межах зображення. Striding – техніка, що має на меті отримати вихідні дані меншого розміру. Це зменшує необхідні обчислення, що дозволяє швидше та дешевше навчати нейронну мережу, і також використовувати її на більш слабких пристроях. Метод обирає пікселі за заздалегідь встановленим алгоритмом (середнє, максимальне, мінімальне) з розрахованого скалярного добутку згортки, що дозволяє вдвічі зменшувати карту ознак. Такі параметри називають гіперпараметрами, їх визначає розробник нейронної мережі. Часто, потрібно провести багато ручних експериментів для оптимізації, або застосувати автоматичні алгоритми оптимізації для досягнення кращих метрик ЗМН, і ці операції часто потребують багато обчислювальних годин та ресурсів.

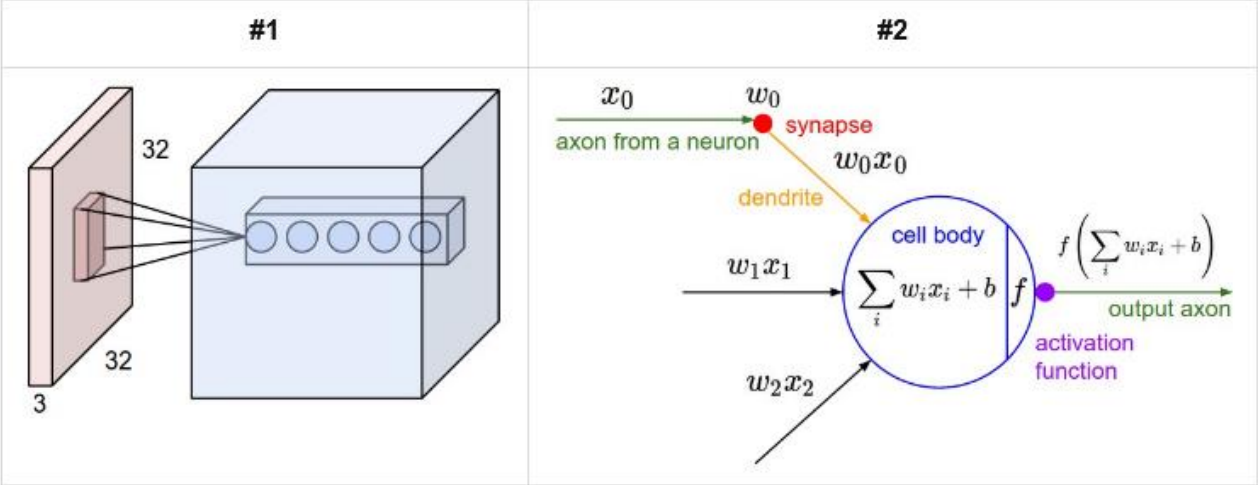

Рисунок 6. Процес роботи нейрону всередині ШНМ

З лівої сторони: вхідна вистава відображена червоним кольором (наприклад, зображення розміром 32x32x3 CIFAR-10) та приклад подання нейронів у першому згортковому шарі. Кожен нейрон у згортковому шарі пов’язаний лише з локальною областю вхідного уявлення, але повністю по глибині (у прикладі – по всіх колірних каналах). Зверніть увагу, що на зображенні безліч нейронів (у прикладі – 5) і розташовані вони за 3-м виміром (глибиною) – трохи нижче будуть дані пояснення щодо такого розташування. З правого боку: нейрони з нейронної мережі, як і раніше, залишаються незмінними: вони, як і раніше, обчислюють скалярний твір між своїми вагами та вхідними даними, застосовують функцію активації, але їх зв’язність тепер обмежена просторовою локальною областю.

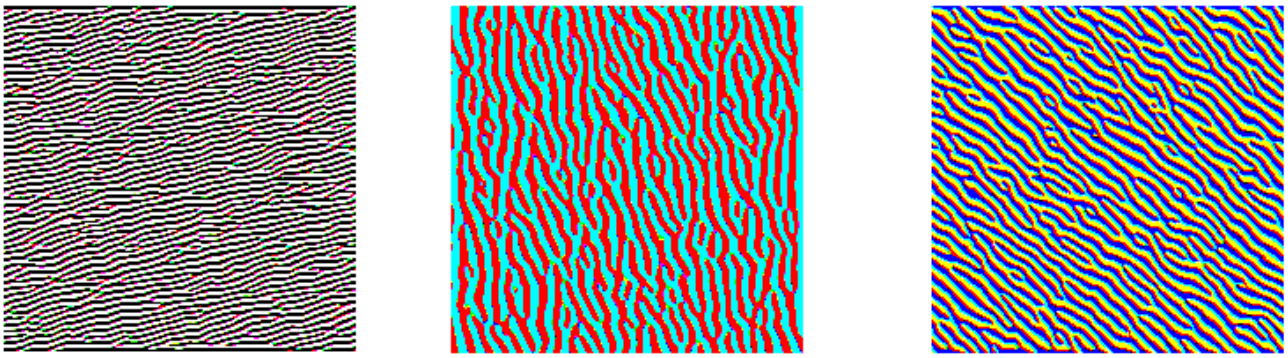

Рисунок 7. Візуалізація зображення всередині нейронної мережі

На рисунку 7 зображена візуалізація для трьох каналів після першого згорткового шару.

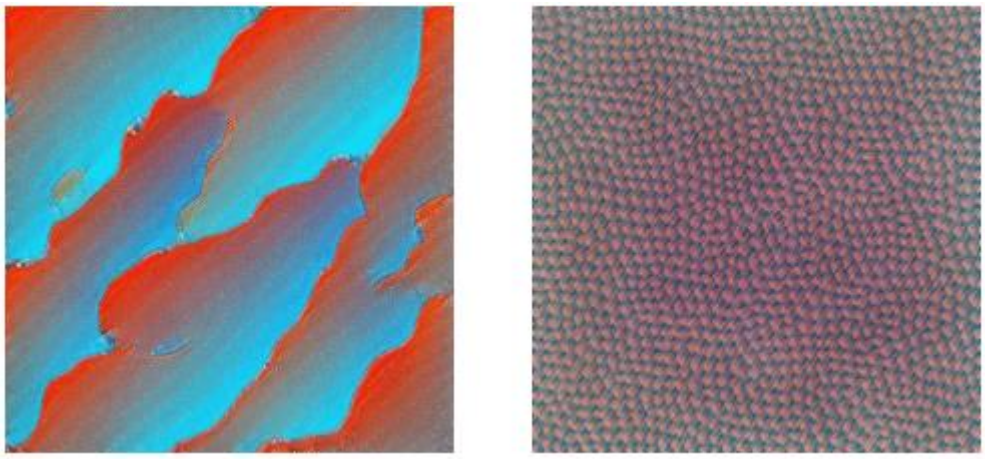

Рисунок 8. Візуалізація зображення в пізніх шарах нейронної мережі

Перед основним етапом в створенні готового алгоритму перше що потрібно – створити набір даних для навчання. Набір даних має найбільший вплив на якість кінцевої моделі. Він має бути максимально подібним до умов, в яких планується використання навченої нейронної мережі, бути максимально репрезентативним та мати достатню кількість даних для тренування та перевірки. Перевірка якості нейронної мережі має проводитись на даних, які не використовувались під час тренування. Зазвичай весь набір даних ділять в пропорції 80% прикладів для тренування, 20% для тестування. Також існують методи обробки даних, що дозволяють змінювати дані, щоб імітувати певні умови, що не були відображені в наборі даних, або додатково створити більш складні приклади для тренування для запобігання перенавчанню. Аугментації – методи додавання прикладів на основі існуючих з певними змінами [35]. Наприклад, найпростіша аугментація для згорткових нейронних мереж – оберт зображення на певний кут. Також до простих методів можна віднести віддзеркалення, розмиття, вирізання певної області, зміна контрасту, гами, зміна колірних каналів. Всі ці методи є автоматичними, і застосовуються з певною ймовірністю до кожного зображення під час навчання. Зазвичай, кількість змінених зображень не перевищує 30% від всього набору даних, і як і список параметрів для зміни зображення, ймовірність перетворення і параметри для кожної операції перетворення налаштовуються перед навчанням моделі. Далі нейронна мережа має «тренуватись» для виконання поставленого завдання. В загальному випадку навчання нейронної мережі представляє собою послідовне подання на даних на вхід нейронної мережі з навчального набору та порівняння відповіді з бажаною відповіддю, отримана різниця між відповідями являє собою результат функції помилки. Потім цю дельту потрібно застосувати

в коригуванні вагів всієї нейронної мережі. Таким чином навчання нейронної мережі зводиться до мінімізації функції помилки шляхом зміни вагових коефіцієнтів зав'язків між нейронами. У випадку детекції обличь на фото, у найпростішому випадку ми можемо застосувати простий бінарний класифікатор, де 0 – обличчя немає, 1 – обличчя є. В такому випадку, якщо вихід нейронної мережі буде 0.8, а бажаний 1 то помилка нейронної мережі складатиме 0.2. Для того, щоб скорегувати всі коефіцієнти нейронної мережі застосовується алгоритм зворотного розповсюдження помилки. Метод було запропоновано в 1986 році Румельхартом, Макклеландом та Вільямсом. Для вихідного шару нейронної мережі корегування помилки найпростіше. Ваги прихованих шарів мають змінюватись прямо пропорційно помилці тих нейронів, з якими він пов'язаний. Всі ваги ШНМ на початку тренування представляють собою випадкові числа, тому величина помилки на початку висока. В цьому випадку зворотне розповсюдження помилки правильно корегує ваги мережі для зменшення загальної помилки. На кожній ітерації виконується два проходи мережі – прямий та зворотній. На прямому вхідний вектор розповсюджується від входів ШНМ до її виходів і формує деякий вихідний вектор відповідний до фактичного стану вагів мережі. Потім розраховується помилка нейронної мережі як різність між отриманими та бажаними значеннями відповідних навчальних прикладів. На зворотному проході помилка розповсюджується від виходів до входів нейронної мережі і виконується коригування вагів мережі за формулою:

$$\Delta\omega_{i,j}(n) = -\eta \frac{\partial E_{av}}{\delta \omega_{ij}} \quad (1)$$

де $\omega_{i,j}$ – коефіцієнт і-го зв'язку j-го нейрону, η – параметр швидкості навчання, який дозволяє керувати величиною кроку корекції $\Delta\omega_{i,j}$ з метою більш точного налаштування мінімізації помилки, і підбирається експериментально. Якщо описати більш точно формулу розрахунку для конкретного нейрону, то вона матиме вигляд:

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial O} * \frac{\partial O}{\partial x} \quad (2)$$

Де $\frac{\partial L}{\partial x}$ – градієнт для оновлення фільтру L, $\frac{\partial L}{\partial O}$ – градієнт помилки з попереднього нейрону, $\frac{\partial O}{\partial x}$ – локальний градієнт. Таким чином, повний зворотній прохід по нейронній мережі для коригування коефіцієнтів потребує

обчислення великої кількості часткових похідних для кожного шару з урахуванням всіх попередніх шарів. Це складне обчислювальне завдання, що вимагає розрахунку великої кількості даних, тому частіше всього виконується на графічних прискорювачах з великою кількістю пам’яті, а якщо врахувати той факт, що для цього використовується лише невеликий шматок навчальних даних – batch, задля більш точного налаштування і для того, щоб всі дані розрахунків постійно знаходились і оновлювались в пам’яті графічного прискорювача. Кількість батчів визначає кількість даних в них, та сумарна кількість навчальних прикладів. Коли всі навчальні приклади пройшли крізь нейронну мережу – проходить епоха навчання. Для навчання потрібна різна кількість епох, зазвичай для навчання нейронної мережі з нуля потрібно декілька десятків епох.

В якості методу оптимізації часто використовують метод градієнтного спуску. Суть методу зводиться до пошуку локального мінімуму (максимум) функції за рахунок руху вздовж вектору градієнта. Градієнтні методи — це широкий клас оптимізаційних алгоритмів, які використовують не лише в машинному навчанні.

Використання градієнтного спуску не є обов’язковим, на даний час існує безліч алгоритмів оптимізації, які створено з розвитком машинного навчання. Часто, нейронні мережі певної архітектури, або певного призначення можуть мати «власні назви». Зазвичай, це акроніми основних технологій, які використані під час створення нової моделі. Нижче описано декілька найвідоміших нейронних мереж, що показали найкращий результат в загальноприйнятих завданнях для оцінки роботи моделі.

LeNet [36]. Перша успішна реалізація згорткової нейронної мережі була розроблена в 1990 році. З них найвідомішою стала архітектура LeNet, яка використовувалася для читання ZIP-кодів, цифри та ін.

AlexNet [37]. Перша робота, яка популяризувала згорткові нейронні мережі в комп’ютерному зорі. AlexNet була протестована на змаганні ImageNet ILSVRC у 2012 році та значно випередила конкурента на другому місці (відсоток помилок: 16% проти 26%). Архітектура мережі була дуже схожа на архітектуру LeNet, але була глибшою, більшою і використовувала послідовності згорткових шарів (до цього було стандартною практикою використовувати тільки комбінацію згорткового шару відразу за яким слідував шар підвиборки).

ZFNet [38]. Переможцем ILSVRC 2013 року стала згорткова нейронна мережа від Matthew Zeiler та Rob Fergus. Стала вона відома під назвою ZFNet. Це була вдосконалена версія AlexNet, яка використовувала інші значення гіперпараметрів, зокрема, що збільшувала розмір середнього згорткового шару і зменшувала крок і розмір фільтра на першому згортковому шарі.

GoogLeNet [38]. Переможцем ILSVRC 2014 стала згорткова нейронна мережа від Google. Основною зміною в архітектурі став Inception-модуль, який дозволяв значно знизити кількість параметрів у мережі (4 Мб проти 60 Мб в AlexNet). На додаток до цього мережа використовує підбірку за середнім значенням замість повнозв'язкових шарів до кінця мережі, виключаючи тим самим велику кількість параметрів, які не мають великого значення. Ця версія мережі має кілька модифікацій, одна з останніх — Inveption-v4.

VGGNet [5]. Слідом за переможцем 2014 року ILSVRC була мережа від Karen Simonyan та Andrew Zisserman, яка стала так само відома як VGGNet. Основним внеском цієї нейронної мережі у розвитку уявлень про згорткові нейронні мережі був той факт, що глибина мережі є ключовим компонентом хорошої продуктивності. Фінальна версія їх мережі містила 16 пар згорткових + повнозв'язкових шарів та використовувала єдині розміри фільтрів (3x3 для згортки та 2x2 для операції підвиборки). Їх передбачена модель перебуває у публічному доступі і з нею можна поекспериментувати. Зворотний бік медалі VGGNet – вартість обчислень та використання великої кількості пам'яті (140Мб). Більшість цих параметрів знаходяться на першому повнозв'язному шарі і, як було надалі вивчено і протестовано, ці повнозв'язкові шари можуть бути виключені без впливу на продуктивність, але з колосальним зменшенням кількості необхідних параметрів.

ResNet [40]. Residual-мережа була розроблена Kaiming He et al. І стала переможцем у ILSVRC 2015. Вона активно використовує пропуск зв'язків та блокову нормалізацію. В архітектурі також відсутні пов'язані шари в кінці мережі. На даний момент це найкраще рішення з використанням згорткових нейронних мереж (травень 2016).

Основна технологія в області Deepfake – Генеративно-змагальна нейронна мережа (GAN) [41][42]. ГЗМ – алгоритм машинного навчання без вчителя, побудований на комбінації двох нейронних мереж, одна з яких генератор (мережа Г) генерує зразки, інша дискримінатор (мережа Д) намагається відрізнити згенеровані зразки від справжніх [43]. Так як мережі Г і

Д мають протилежні цілі – створити зразки та відбракувати зразки, між ними виникає гра з нульовою сумою. Дискримінатор намагається класифікувати вхідні дані. Він отримує на вхід зображення з тренувального набору та від Генератора, і його задача, в найпростішому випадку – бінарна класифікація зображення. В цьому випадку 0 – реальне зображення, 1 – згенероване. Генератор – нейронна мережа, що отримує на вхід випадкові дані, і на виході віддає вже узагальнені дані, що мають бути схожими на тренувальні приклади. Після створення генератором прикладу він потрапляє до дискримінатора, і той рахує величину похибки, яка методом зворотного розповсюдження помилки корегує ваги Генератора [44].

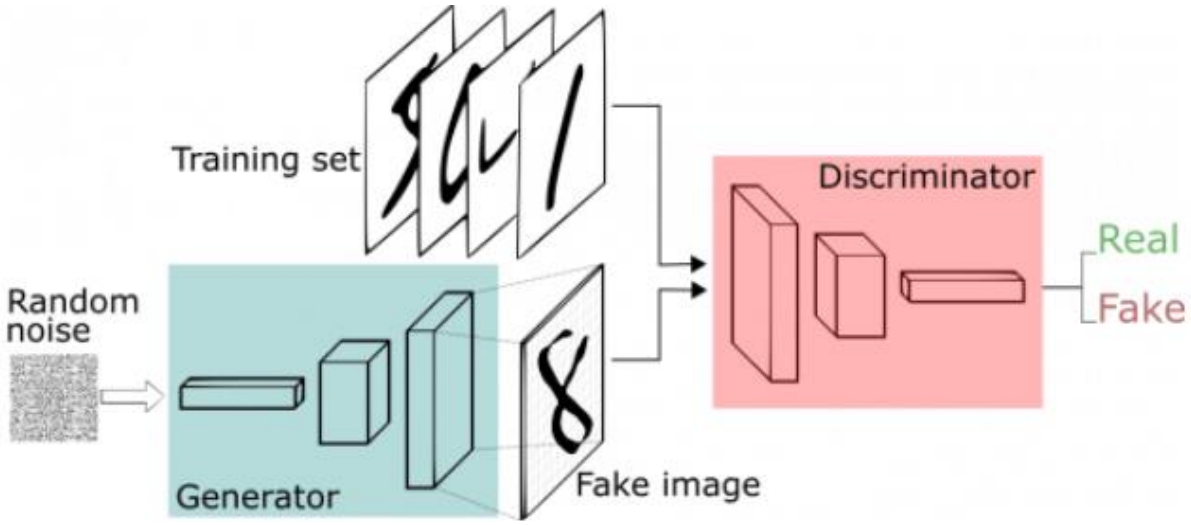

Рисунок 9. Приклад будови ГЗН

Зазвичай, будова Дискримінатора простіша за будову Генератора. Адже у нього доволі просте завдання на початку тренування Генератора – відрізнити шум від справжніх даних. Проте, чим більше навчається Генератор, тим складніше простому Дискримінатору відрізнити дані, тому зазвичай будова Дискримінатора майже повторює архітектуру Генератора для досягнення кращих результатів генерації. Щоб визначити ймовірнісний розподіл генератору p_g над набором даних X, визначимо апріорну ймовірність шуму $P_z(z)$ і представимо генератор, як відображення $G(z, \varphi_z)$, де G – диференційована функція, представлена нейронною мережею. Аналогічним чином представимо дискримінатор $D(z, \varphi_z)$, який на вхід подає скалярне значення – ймовірність того, що x належить до тренувальних даних, а не до p_g. Завдання згенерувати певне зображення еквівалентне завданню генерації вектору зі статистичного розподілу зображень в n-вимірному просторі. Тобто,

потрібно згенерувати випадкову величину з певного статистичного розподілу. Складність полягає в тому, що перш за все це багатовимірний розподіл, по-друге – потрібно створити правильний розподіл, який би якомога більше відповідав тренувальним даним, враховуючи всі параметри вхідних зображень. Також, формула такого статистичного розподілу неочевидна, так як нейронна мережа апроксимізує цей розподіл з певною точністю.

Під час тренування Дискримінатору ми намагаємось максимізувати ймовірність правильної ідентифікації об'єктів з тренувальної та з генерованої вибірок. В той же час ми тренуємо Генератор для мінімізації $log\ log\ \left(1 - D\left(G(z)\right)\right)$. Іншими словами це і є гра з нульовою сумою:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p_z}[\log(1-D(G(z)))] \quad (3)$$

В процесі навчання потрібно робити два кроки оптимізації почергово: спочатку оновити ваги Генератора при фіксованих вагах Дискримінатора, потім оновити ваги Дискримінатора при фіксованих вагах Генератора. На практиці, ваги Дискримінатора оновлюються значно рідше, зазвичай k разів замість одного, через те що його оптимізація кожен раз обчислювальне неефективна. Також на практиці, описаний метод використовується частіше в навчальних цілях через низьку якість згенерованих зображень.

Це був опис перших, найпростіших підходів для генерації зображень, ідея якого використовується і зараз, але методи тренування постійно розвиваються для покращення результатів роботи.

Іншим популярним підходом для генерації зображень є варіаційний автоенкодер (VAE) [45]. ВА складається з двох нейронних мереж – енкодера та Деенкодера. Додатковий метод, що застосовується в цьому підході – метод зменшення розмірності. Метод головних компонент (МГК, англ. Principal component analysis, PCA) [46] — метод факторного аналізу в статистиці, який використовує ортогональне перетворення множини спостережень з можливо пов'язаними змінними (сутностями, кожна з яких набуває різних числових значень) у множину змінних без лінійної кореляції, які називаються головними компонентами. Ідея PCA полягає в тому, щоб побудувати n нових незалежних ознак, які є лінійними комбінаціями m старих ознак і так, щоб проекції даних на підпростір, визначений цими новими ознаками, були максимально наближені до вихідних даних (з точки зору евклідового відстань). Іншими словами, PCA шукає найкращий лінійний підпростір початкового простору

(описаного ортогональним базисом нових ознак), щоб похибка апроксимації даних їх проекціями на цей підпростір була якомога меншою.

Загальна ідея автоенкодерів досить проста і полягає в налаштуванні кодувальника і декодера як нейронних мереж і в ознайомленні з найкращою схемою кодування-декодування за допомогою ітераційного процесу оптимізації. Отже, на кожній ітерації ми надаємо архітектуру автоенкодера (кодувальник, а потім декодер) деякими даними, ми порівнюємо закодований-декодований вихід з початковими даними і поширюємо помилку назад через архітектуру, щоб оновити ваги мереж. Таким чином, інтуїтивно, загальна архітектура автоенкодера (кодувальник і декодер) створює вузьке місце для даних, яке гарантує, що лише основна структурована частина інформації може проходити й відновлюватися. Дивлячись на нашу загальну структуру, сімейство E розглянутих кодувальникиів визначається архітектурою мережі кодувальник, сімейство D розглянутих декодерів визначається архітектурою мережі декодера, а пошук кодувальника та декодера, які мінімізують помилку реконструкції, здійснюється шляхом градієнтного спуску. Над параметрами цих мереж. До цього часу ми обговорювали проблему зменшення розмірності та представляли автоенкодери, які є архітектурами енкодер-декодер, які можна навчати за допомогою градієнтного спуску. Давайте тепер зв'яжемося з проблемою генерації вмісту, подивимося на обмеження автоенкодерів у їхній поточній формі для цієї проблеми та представимо варіаційні автоененкодери. Обмеження автоенкодерів для генерації контенту. У цей момент на думку спадає природне запитання: «Який зв'язок між автоенкодера ми та генерацією контенту?». Дійсно, коли автоенкодер був навчений, ми маємо і енкодер, і декодер, але все ще немає реального способу створення нового вмісту. На перший погляд у нас може виникнути спокуса подумати, що, якщо латентний простір досить регулярний (добре «організований» енкодером під час навчального процесу), ми можемо випадково взяти точку з цього прихованого простору та декодувати її, щоб отримати новий зміст. Тоді декодер буде діяти більш-менш як генератор генеративної змагальної мережі. Отже, щоб мати можливість використовувати декодер нашого автоенкодера для генеративної мети, ми повинні бути впевнені, що прихований простір достатньо регулярний. Одним з можливих рішень для отримання такої регулярності є введення явної регуляризації під час процесу навчання. Таким чином, як ми коротко згадували у вступі до цієї публікації, варіаційний автоенкодер можна визначити як

автоенкодер, навчання якого упорядковано, щоб уникнути переобладнання та гарантувати, що латентний простір має хороші властивості, які дозволяють генерувати процес. Як і стандартний автоенкодер, варіаційний автоенкодер — це архітектура, що складається з енкодера і деенкодера, яка навчена мінімізувати помилку реконструкції між закодованими-декодованими даними та вихідними даними. Однак, щоб ввести деяку регуляризацію прихованого простору, ми переходимо до невеликої модифікації процесу кодування-декодування: замість кодування вхідних даних як єдиної точки, ми кодуємо його як розподіл у прихованому просторі. Потім модель навчається наступним чином: по-перше, вхідні дані кодуються як розподіл у прихованому просторі по-друге, з цього розподілу відбирається точка з латентного простору по-третє, вибіркова точка декодується, і помилка реконструкції може бути обчислена нарешті, помилка реконструкції поширюється назад через мережу Різниця між автоенкодером (детермінованим) і варіаційним автоенкодером (імовірнісним). На практиці закодовані розподіли вибираються нормальними, щоб енкодер можна було навчити повертати середнє значення та коваріаційну матрицю, які описують гаусівський розподіл даних. Причина, чому вхідні дані кодуються як розподіл з деякою дисперсією замість однієї точки, полягає в тому, що це дає можливість дуже природно виразити регулярізацію прихованого простору: розподіли, які повертає енкодер, змушені бути близькими до стандартного нормального розподілу. У наступному підрозділі ми побачимо, що таким чином ми забезпечуємо як локальну, так і глобальну регуляризацію латентного простору (локальну через контроль дисперсії та глобальну через контроль середнього значення).

Таким чином, функція втрат, яка мінімізується під час навчання VAE, складається з «терміну реконструкції» (на кінцевому рівні), який прагне зробити схему кодування-декодування якомога ефективнішою, і «терміну регуляризації» (на кінцевому рівні). Латентний шар), який має тенденцію регулювати організацію прихованого простору, роблячи розподіли, які повертає енкодер, близькими до стандартного нормального розподілу. Цей термін регуляризації виражається як розбіжність Кульбака-Лейблера [47] між поверненим розподілом і стандартним гаусовим і буде додатково обґрунтовано в наступному розділі. Ми можемо помітити, що розбіжність Кульбака-Лейблера між двома гаусовими розподілами має замкнену форму, яка може бути безпосередньо виражена в термінах середніх і коваріаційних матриць двох

розподілів. У варіаційних автоенкодерах функція втрат складається з терміну реконструкції (що робить схему кодування-декодування ефективною) і терміну регуляризації (що робить латентний простір регулярним). Інтуїція щодо регуляризації, яка очікується від латентного простору, щоб зробити генеративний процес можливим, може бути виражена через дві основні властивості: безперервність (дві близькі точки в латентному просторі не повинні давати два абсолютно різних вмісту після декодування) і повноту (для вибраного розподілу , точка, відібрана з латентного простору, після декодування повинна давати «значущий» вміст).

Різниця між «звичайним» та «нерегулярним» латентним простором [48]. Єдиний факт, що VAE кодують вхідні дані як розподіли замість простих точок, недостатньо для забезпечення безперервності та повноти. Без чітко визначеного терміну регуляризації модель може навчитися, щоб мінімізувати помилку реконструкції, «ігнорувати» той факт, що розподіли повертаються і поводяться майже як класичні автоенкодери (що призводить до переобладнання) [49]. Для цього енкодер може або повертати розподіли з крихітними дисперсіями (які мають тенденцію бути пунктуальними розподілами), або повертати розподіли з дуже різними засобами (які тоді будуть дуже далеко один від одного в латентному просторі). В обох випадках розподіл використовується неправильним чином (скасовуючи очікувану вигоду), і безперервність та/або повнота не задовольняються. Отже, щоб уникнути цих ефектів, ми повинні упорядкувати як коваріаційну матрицю, так і середнє значення розподілів, які повертає енкодер. На практиці ця регуляризація здійснюється шляхом забезпечення розподілу, наближеного до стандартного нормального розподілу (центрованого та зменшеного). Таким чином, ми вимагаємо, щоб коваріаційні матриці були близькими до тотожності, запобігаючи точковим розподілам, а середнє значення було близьким до 0, запобігаючи тому, щоб закодовані розподіли були занадто віддалені один від одного. Для отримання прихованого простору з хорошими властивостями необхідно регуляризувати повернені розподіли VAE. За допомогою цього терміну регуляризації ми запобігаємо кодуванню моделі далеко один від одного в латентному просторі і заохочуємо якомога більше повернених розподілів до «перекривання», задовольняючи таким чином очікувані умови безперервності та повноти. Природно, як і для будь-якого терміну регуляризації, це відбувається за ціною більшої помилки реконструкції навчальних даних. Проте компроміс між помилкою

реконструкції та дивергенцією KL можна відкоригувати, і в наступному розділі ми побачимо, як вираз балансу природним чином випливає з нашого формального виведення. На завершення цього підрозділу можна помітити, що безперервність і повнота, отримані за допомогою регуляризації, мають тенденцію створювати «градієнт» над інформацією, закодованою в латентному просторі. Наприклад, точка латентного простору, яка була б на півдорозі між середніми двома закодованими розподілами, що надходять з різних навчальних даних, повинна бути декодована в щось, що знаходиться десь між даними, які дали перший розподіл, і даними, які дали другий розподіл як він може бути відібраний автоенкодером в обох випадках. Завдяки цьому терміну регуляризації ми запобігаємо кодуванню моделі далеко один від одного в латентному просторі та заохочуємо якомога більше повернених розподілів до «перекривання», задовольняючи таким чином очікувану безперервність і повноту умови. Природно, як і для будь-якого терміну регуляризації, це відбувається за ціною більшої помилки реконструкції навчальних даних.

Проте компроміс між помилкою реконструкції та дивергенцією KL можна відкоригувати, і в наступному розділі ми побачимо, як вираз балансу природним чином випливає з нашого формального виведення. На завершення цього підрозділу можна помітити, що безперервність і повнота, отримані за допомогою регуляризації, мають тенденцію створювати «градієнт» над інформацією, закодованою в латентному просторі. Наприклад, точка латентного простору, яка була б на півдорозі між середніми двома закодованими розподілами, що надходять з різних навчальних даних, повинна бути декодована в щось, що знаходиться десь між даними, які дали перший розподіл, і даними, які дали другий розподіл як він може бути відібраний автоенкодером в обох випадках.

Технологія верифікації [50] за обличчям також використовує нейронні мережі. Алгоритм верифікації можна створити за допомогою двох нейронних мереж: перша визначає область фотографії, на якій ймовірно знаходиться обличчя, а друга – створює зі знайденої області з обличчям створює інформаційний вектор фіксованої довжини для подальшого порівняння з іншими обличчями [51]. Для кодування інформації про обличчя також використовують згорткові нейронні мережі, але зі зміненою архітектурою виходів та більш складним алгоритмом тренування. Після тренування така ЗНМ може впізнавати обличчя людей навіть з більшою точністю ніж люди. Новітні

роботи в сфері розпізнавання обличь показують точність від 91% до 99%, в той час як люди на аналогічних наборах даних помиляються значно частіше [52].

Задачу розпізнавання обличь також можна вирішувати двома способами. Перший це «навчання метриці», який полягає у тому, що нейронна мережа навчається збільшувати відстань між фотографіями різних людей та зменшувати відстань між фотографіями однієї людини. Другий метод полягає в тому, що нейронна мережа напряму класифікує обличчя по заданим класам. Але для реальної роботи потрібно прибрати останній її шар, який перетворює вектори на класи, які відносяться до конкретних людей. Зазвичай використовують перший метод через більшу зручність тренування та більшу точність роботи.

Для роботи методу «навчання метриці» потрібно тренувати ЗГН на спеціально створеному наборі даних обличь, який містить пари фотографій однієї людини. Таких пар може бути довільна кількість, головне, щоб вони належали одній людині але відрізнялись між собою, та були зібрані в різноманітних умовах: різне освітлення, різний кут нахилу голови, різна зачіска, одяг, міміка тощо. Також існує проблема вибірки даних. Нейронна мережа буде добре працювати з даними, які схожі на тренувальний набір. Тобто важливо, щоб умови роботи нейронної мережі відповідали тренувальній вибірці, яка в свою чергу має відповідати генеральній сукупності всіх можливих даних, з якими може стикнутись нейронна мережа. Тому якщо тренувальний набір складається лише з людей певної раси, або певного кольору шкіри, то нейронна мережа може погано працювати на людях інших рас, або з іншим кольором шкіри.

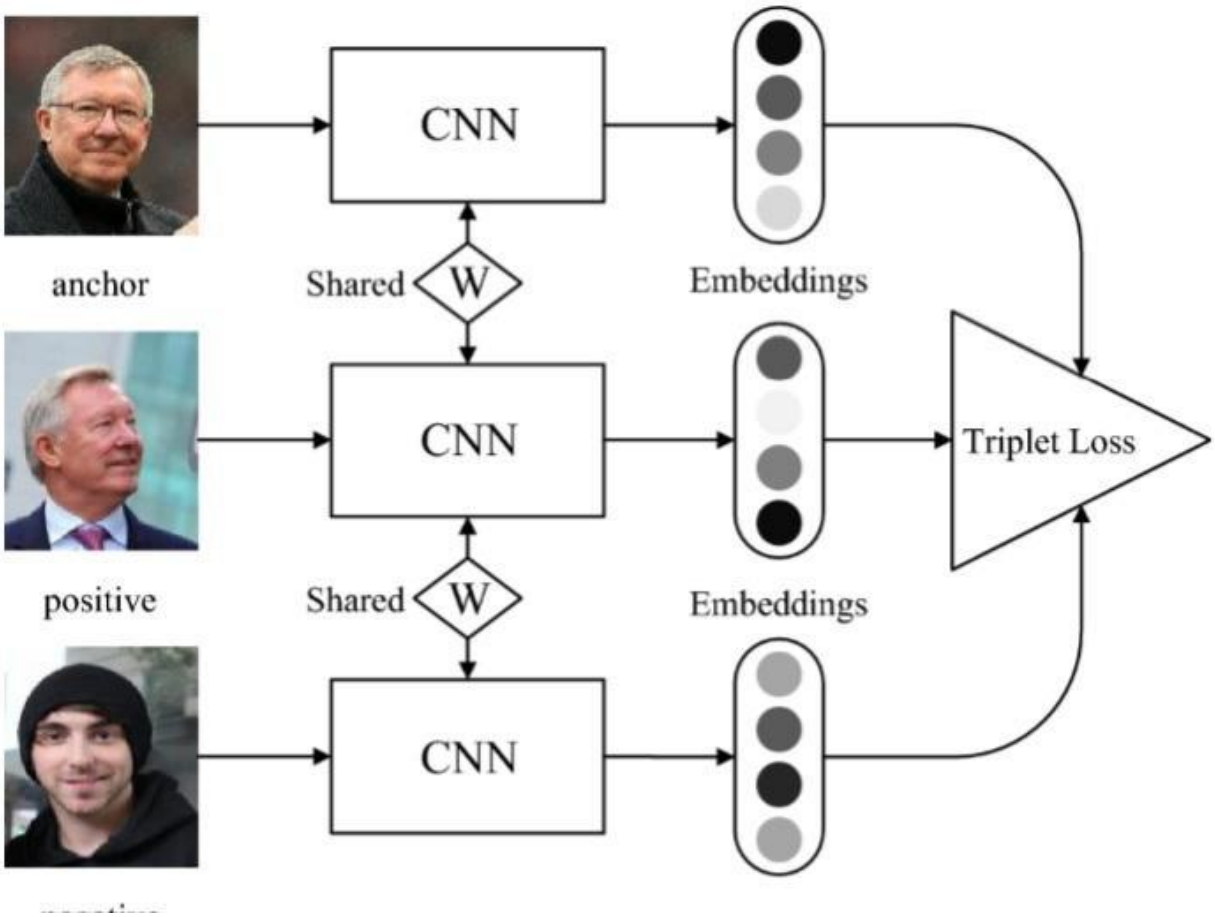

Рисунок 10. Приклад роботи алгоритму пошуку обличчя

Так, допомогою спеціальної функції помилки нейронна мережа і навчається розташовувати обличчя однієї людини близько, а інших – далеко у багатовимірному векторному просторі. Під час реальної роботи такої мережі залишається реалізувати лише пошук найбільш схожих до відповідного вектору що повертає нейронна мережа. Зазвичай використовують косинусну відстань, яка дозволяє вимірювати кут між двома векторами та працює в проміжку [0, 1]. Завдяки властивостям отриманого простору обличь та косинусній відстані можна легко ідентифікувати та порівняти обличчя.

Методи розпізнавання обличчя наразі набули широкої розповсюдженості. За їх допомогою можна знаходити фотографії людини в інтернеті, знаходити злочинців за камерами відеонагляду в містах, використовувати їх в системах безпеки та в різноманітному програмному забезпеченні. Також методі розпізнавання обличчя використовують в алгоритмах генерації контенту. Якщо відео, в якому потрібно замінити обличчя містить декілька обличчя, які необхідно лишити без змін, очевидно що потрібно вказати алгоритму заміни обличчя те, з яким він має безпосередньо працювати. Також корисно мати деяку інформацію про обличчя, яку можна застосовувати при заміні. Варто

зазначити, що алгоритми генерації на основі генеративно-змагальних нейронних мереж беруть свій початок з алгоритмів пошуку та розпізнавання обличь, і більшість методів, які використовують в розпізнаванні так чи інакше, з певними модифікаціями також використовують і в генерації.

Незважаючи широке розповсюдження такого методу, в спеціалізованих установах (банки, підприємства з високим рівнем захисту) його використовують скоріше як додатковий метод. Для двомірного розпізнавання обличь не потрібно дорогого обладнання, достатньо звичайної камери відеонагляду, яка може зафіксувати обличчя з фронтального боку в хорошій якості. Тому метод частіше всього використовують правоохоронці в камерах нагляду у місті, та в мережі Інтернет для пошуку за фотографіями. Більшу точність може давати метод тривимірної біометричної верифікації за обличчям, але для нього потрібне дороге обладнання для сканування, і він очевидно не підходить для розпізнавання обличь на фотографіях та відеозаписах у зв'язку з відсутністю інформації про глибину зображення.

2.2 Модель генерації рухів першого порядку типу «Ляльковод».

До найбільш простого типу Deepfake можна віднести алгоритм «Ляльковода». Цей алгоритм вимагає лише одне зображення обличчя жертви, на яке нейронна мережа перенесе анімацію обличчя зловмисника. Анімація зображення складається зі створення відеопослідовності, щоб об'єкт у вихідному відеоролику анімувався відповідно до руху відеоролика. Алгоритм вирішує це завдання без використання будь-яких анотацій або попередньої інформації про конкретний об'єкт для анімації. Це означає, що алгоритм можна навчити один раз на наборі відео із зображеннями однієї категорії (обличчя, тіла) і далі застосовувати для будь-яких об'єктів цього класу [1]. Для зловмисника це означає, що одна навчена нейронна мережа може анімувати будь-які обличчя з фотографій без необхідності додаткового навчання.

Створення відео за допомогою анімації об'єктів у нерухомих зображеннях має незліченну кількість застосувань у різних областях, включаючи виробництво фільмів, фотографію та електронну комерцію. Точніше, анімація зображень відноситься до завдання автоматичного синтезу відеозаписів шляхом поєднання вигляду, отриманого з вихідного зображення з моделями руху, отриманими з цільового відео. Наприклад, зображення обличчя певної людини можна анімувати за виразом обличчя іншої особи. У літературі

більшість методів вирішують цю проблему, припускаючи сильні пріоритети щодо представлення об’єкта (наприклад, 3D-модель) та вдаючись до технік комп’ютерної графіки. Ці підходи можуть можна називати об’єктно-специфічними методами, оскільки вони передбачають знання про модель конкретного об’єкт для анімації. Останнім часом глибокі генеративні моделі застосовуються як ефективні методи анімації зображень і ретаргетингу відео. Зокрема, генеративні змагальні мережі (GAN) і варіаційні автоенеенкодери (VAE) використовувалися для передачі виразів обличчя, або моделей руху між людьми на відео. Тим не менш, ці підходи зазвичай покладаються на попередньо навчені моделі, щоб отримати специфічні для об’єкта представлення, такі як ключова точка місця розташування. На жаль, ці попередньо підготовлені моделі створені з використанням дорогих анотацій достовірних даних і взагалі недоступні для довільної категорії об’єктів. Щоб вирішити цю проблему, нещодавно Сярохін та ін. представив Monkey-Net, першу глибоку модель для анімації зображень, яка не залежить від об’єктів. Monkey-Net кодує інформацію про рух за допомогою ключових точок, засвоєних за допомогою самоконтролюваного навчання. Під час тестування вихідне зображення анімується відповідно до відповідних траєкторій ключової точки оціненої на відео. Головною слабкістю Monkey-Net є те, що вона погано моделює об’єкт перетворення зовнішнього вигляду в околицях ключових точок, припускаючи модель нульового порядку. Це призводить до низької якості генерації у разі зміни пози великого об’єкта. Для вирішення цієї проблеми було запропоновано використовувати набір ключових моментів, засвоєних самостійно за допомогою локальних афінних перетворень для моделювання складних рухів. По-друге, було введено генератор, що усвідомлює оклюзію, який приймає маску оклюзії і автоматично оцінюється, щоб позначити частини об’єкта, які не видно на вихідному зображенні, і це слід виводити з контексту. Це особливо необхідно, коли відео містить великі розміри і характерні моделі руху та оклюзії. По-третє, було розширено втрату еквіваріації, яка зазвичай використовується для навчання детектора ключових точок, щоб покращити оцінку локальних афінних перетворень. По-четверте, було експериментально показано, що такий метод значно перевершує сучасне зображення анімаційних методів і може обробляти набори даних з високою роздільною здатністю там, де інші підходи зазвичай дають збій.

Нас цікавить анімація об’єкта, зображеного на вихідному зображенні S, на основі руху подібного об’єкта на відео D. Оскільки пряме спостереження недоступне (пари відео, на яких об’єкти рухатися аналогічно), алгоритми дотримуються стратегії самоконтролю, натхненної Monkey-Net [20]. Для навчання, було використано велику колекцію відеорядів, що містять об’єкти однієї категорії об’єктів. Модель навчається реконструювати навчальні відео, поєднуючи один кадр і вивчене латентне відображення руху на відео. Спостерігаючи за парами кадрів, кожна з яких витягнута з одного відео, він вчиться кодувати рух як комбінацію зміщень ключових точок, специфічних для руху, та локальних афінні перетворення. Під час тестування застосовуємо модель до пар, що складаються з вихідного зображення та з кожен кадр ведучого відео та виконують анімацію зображення вихідного об’єкта. Така структура складається з двох основних модулів: модуль оцінки руху та модуль створення зображення.

Модуль оцінки має передбачити щільне поле руху з кадру D ∈ R3×H×W розмірності H ×W відео D до вихідного кадру S ∈R3×H×W. Пізніше поле щільного руху використовується для вирівнювання карт об’єктів, обчислених із S, із позицією об’єкта в D. Поле руху є моделюється функцією TS←D : R2 →R2, яка відображає розташування кожного пікселя в D з відповідним розташування в S. TS←D часто називають зворотним оптичним потоком. В алгоритмах використовуємо зворотний оптичний потік, а не прямий оптичний потік, оскільки зворотне деформування може бути ефективно реалізовано в диференційовані спосіб з використанням білінійної вибірки. В реалізації ми припускаємо, що існує абстрактна система відліку R. Можна самостійно оцінити два перетворення: від R до S (TS←R) і від R до D (TD←R). На відміну від X2Face система відліку є абстрактним поняттям, яке скасовується в виведення пізніше. Тому вона ніколи не обчислюється явно і не може бути візуалізовано. Цей вибір дозволяє незалежно обробляти D і S. Це бажано, оскільки під час тестування модель отримує пари вихідне зображення та кадри керування, взяті з іншого відео, які візуально можуть дуже відрізнятися. Замість того, щоб безпосередньо передбачати TD←R і TS←R, модуль оцінки руху виконується в два етапи. На першому кроці апроксимуємо обидва перетворення з наборів розріджених траєкторій, отриманих за допомогою використання ключових моментів, засвоєних у спосіб самоконтролю. Розташування ключових точок у D і S є окремо передбачено мережею енкодер-декодер. Подання

ключових точок виступає як вузьке місце в результаті чого утворюється компактне представлення руху. Як показали Сярохін та ін., такий розріджений рух представлення добре підходить для анімації, оскільки під час тестування ключові точки вихідного зображення можуть бути переміщено за допомогою траєкторій ключових точок у відео водіння. Моделюємо рух по сусідству кожної ключової точки за допомогою локальних афінних перетворень. У порівнянні з використанням лише зміщення ключових точок, локальні афінні перетворення дозволяють моделювати більшу кількість перетворень. Ми використовуємо ряд Тейлора для розширення для представлення TD←R набором розташування ключових точок і афінних перетворень. До цього кінця, мережа детектора ключових точок виводить розташування ключових точок, а також параметри кожного афінного трансформація. Під час другого кроку мережа щільного руху поєднує локальні наближення для отримання результуюче поле щільного руху ˆTS←D. Крім того, крім щільного поля руху, ця мережа виводить маску оклюзії ˆOS←D, яка вказує, які частини зображення D можна відновити за допомогою викривлення вихідного зображення та те, які частини слід зафарбувати, тобто визначити з контексту. Нарешті, модуль генерації відтворює зображення вихідного об’єкта, що рухається, як це передбачено в відео. Тут ми використовуємо генеративну мережу G, яка деформує вихідне зображення відповідно до ˆTS←D і зафарбовує частини зображення, які закриті у вихідному зображенні.

Модуль оцінки руху оцінює зворотний оптичний потік TS←D від керуючого кадру D на вихідний кадр S[19]. Як обговорювалося вище, ми пропонуємо апроксимувати TS←D його першим порядком Тейлора розширення поблизу ключових точок. В решті частини цього розділу ми опишемо мотивація цього вибору та деталізація запропонованої апроксимації TS←D. Ми припускаємо, що існує абстрактна система відліку R. Отже, оцінка TS←D полягає в оцінка TS←R і TR←D. Крім того, враховуючи кадр X, ми оцінюємо кожне перетворення TX←R поблизу вивчених ключових точок. Формально, враховуючи перетворення TX←R, ми розглянемо його розкладання Тейлора першого порядку в K ключових точках p1, pK. Тут p1,pK позначають координати ключових точок в системі відліку R. Зауважимо, що для простоти в після розташування точок у просторі опорної пози всі позначаються p, тоді як розташування точок у позі X, S або D пробіли позначаються z. Ми отримуємо:

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p) = \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k) + \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right)(p-p_k) + o(\|p-p_k\|), \quad (4)$$

У цій формулюванні функція руху TX←R представлена її значеннями в кожній ключовій точці pk і його якобіани обчислюються в кожному місці pk:

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p) \simeq \left\{\left\{\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_1), \frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\Big|_{p=p_1}\right\}, \ldots \left\{\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k), \frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\Big|_{p=p_K}\right\}\right\}.$$

(5)

Крім того, щоб оцінити TR←X = T−1 X←R, ми припускаємо, що TX←R локально бієктивний у околиці кожної ключової точки. Нам потрібно оцінити TS←D поблизу ключової точки zk в D, враховуючи це zk — це розташування пікселя, що відповідає розташуванню ключової точки pk у R. Для цього спочатку ми оцінимо перетворення TR←D поблизу точки zk в системі керування D, напр. pk = TR←D(zk). Тоді ми оціните перетворення TS←R біля pk в еталонні R. Нарешті TS←D виходить таким чином:

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}_{\mathbf{R}\leftarrow\mathbf{D}} = \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}} \circ \mathcal{T}^{-1}_{\mathbf{D}\leftarrow\mathbf{R}}, \quad (6)$$

Після повторного обчислення розкладання Тейлора першого порядку рівняння отримуємо:

$$\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{D}}(z) \approx \mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + J_k(z - \mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k)) \quad (7)$$

з

$$J_k = \left(\frac{d}{dp}\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right)^{-1} \quad (8)$$

На практиці TS←R(pk) і TD←R(pk) в екв. (4) прогнозуються за допомогою предиктора ключової точки. Більше саме ми використовуємо стандартну архітектуру U-Net, яка оцінює K теплових карт, по одній для кожної ключовий момент. Останній рівень деенкодера використовує софтмакс функції активації, щоб передбачити теплові карти, які можуть можна інтерпретувати як карту довіри виявлення ключових точок. Кожне очікуване розташування ключової точки оцінюється використовуючи середню операцію. Зауважте, якщо ми встановимо Jk = 1 (1 — це ідентична матриця 2 × 2), ми отримати модель руху Monkey-Net. Тому Monkey-Net використовує апроксимацію нульового порядку TS←D(z) −z. Для обох кадрів S і D мережа провісників

ключових точок також виводить чотири додаткові канали для кожну ключову точку. З цих каналів отримуємо коефіцієнти матриць d dp TS←R(p)|p=pk і d dp TS←R(p)|p=pk в рівнянні. (5) шляхом обчислення просторового середньозваженого з використанням як ваг відповідних складання карти довіри ключових точок. Поєднання місцевих рухів. Ми використовуємо згорткову мережу P для оцінки ˆTS←D з множини апроксимації Тейлора TS←D(z) у ключових точках та вихідному кадрі S. Важливо, оскільки ˆTS←D відображає кожне розташування пікселя в D з його відповідним розташуванням у S, локальні шаблони в ˆTS←D, такі як краї або текстура, вирівнюються від пікселя до пікселя за D, але не за S. Це зміщення проблема робить завдання мережі важче передбачити ˆTS←D від S. Щоб надати вхідні дані вже приблизно вирівняний з ˆTS←D, ми деформуємо вихідний кадр S відповідно до локальних перетворень оцінюється. Таким чином, ми отримуємо K перетворених зображень S1,…SK, кожне з яких вирівняно ˆTS←D поблизу ключової точки. Важливо, що ми також розглянемо додаткове зображення S0 = S для фону. Для кожної ключової точки pk ми додатково обчислюємо теплові карти Hk, що вказують на мережу щільного руху де відбувається кожне перетворення. Кожен Hk(z) реалізується як різниця двох теплових карт з центром у TD←R(pk) і TS←R(pk):

$$\mathbf{H}_k(z) = exp\left(\frac{(\mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k) - z)^2}{\sigma}\right) - exp\left(\frac{(\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) - z)^2}{\sigma}\right). \quad (9)$$

У всіх наших експериментах ми використовуємо σ = 0,01 відповідно до Jakab et al.. Теплові карти Hk і перетворені зображення S0,…SK об’єднуються та обробляються U- Net. ˆTS←D оцінюється за допомогою часткової моделі, натхненної Monkey-Net. Ми припускаємо, що об’єкт складається з K жорстких частин і кожна частина переміщується відповідно до рівняння. Тому ми оцінюємо K+1 маски Mk,k = 0,…K, які вказують, де виконується кожне локальне перетворення. Остаточний прогноз щільного руху ˆTS←D(z) визначається як:

$$\hat{\mathcal{T}}_{\mathbf{S}\leftarrow\mathbf{D}}(z) = \mathbf{M}_0 z + \sum_{k=1}^{K} \mathbf{M}_k \left(\mathcal{T}_{\mathbf{S}\leftarrow\mathbf{R}}(p_k) + J_k(z - \mathcal{T}_{\mathbf{D}\leftarrow\mathbf{R}}(p_k))\right) \quad (10)$$

Зауважте, що термін M0z розглядається для моделювання нерухомих частин, таких як фон.

Генерація зображень з урахуванням оклюзії. Як згадувалося в розділі 3, вихідне зображення S не вирівнюється від пікселя до пікселя із зображенням, яке буде створено ˆD. Щоб впоратися з цим неузгодженістю, ми використовуємо стратегію деформації ознак, подібну до. Точніше, після двох згорткових блоків зниження вибірки ми отримуємо карту ознак ξ∈RH′×W ′ розмірності H′ ×W′. Потім ми деформуємо ξ відповідно до ˆTS←D. При наявності оклюзій в S, оптичного потоку може бути недостатньо для створення ˆD. Дійсно, закриті частини в S не можуть бути відновлені викривленням зображення і, таким чином, має бути зафарбовано. Отже, ми вводимо карту оклюзії ˆOS←D ∈ [0,1]H′×W ′ щоб замаскувати області карти об'єктів, які слід зафарбувати. Таким чином, маска оклюзії зменшує вплив ознак, що відповідають оклюзованим частинам. Трансформована карта об'єктів записується так:

$$\boldsymbol{\xi}' = \hat{O}_{\mathbf{S}\leftarrow\mathbf{D}} \odot f_w(\boldsymbol{\xi}, \hat{T}_{\mathbf{S}\leftarrow\mathbf{D}}) \quad (11)$$

де fw(·,·) позначає операцію зворотного викривлення та позначає добуток Адамара. Ми оцінюємо маску оклюзії з нашого розрідженого представлення ключової точки, додавши канал до останнього шару мережі щільного руху. Нарешті, перетворена карта ознак ξ′ подається в наступну мережу шари модуля генерації для відтворення шуканого зображення.

Функція втрат. Ми тренуємо нашу систему наскрізним способом, поєднуючи кілька втрат. Спочатку використовуємо реконструкцію втрата, заснована на втраті сприйняття за Johnson et al. З використанням попередньо навченої мережі VGG-19 як нашої основна втрата водіння. Втрата заснована на реалізації Wang et al. З введенням водіння кадр D і відповідний відновлений кадр ˆD, втрати реконструкції записуються як:

$$L_{rec}(\hat{\mathbf{D}}, \mathbf{D}) = \sum_{i=1}^{I} \left| N_i(\hat{\mathbf{D}}) - N_i(\mathbf{D}) \right|, \quad (12)$$

де Ni(·) — і-я характеристика каналу, витягнута з конкретного шару VGG-19, а I — кількість канали функцій у цьому шарі. Крім того, ми пропонуємо використати цю втрату для низки резолюцій, утворюючи піраміду, отриману шляхом понижуючого дискретизації ˆD і D, подібно до MS-SSIM. Роздільна здатність 256 × 256, 128 × 128, 64 × 64 і 32 × 32. Всього 20 умов втрати.

Накладення обмеження еквіваріантності. Наш предиктор ключових точок не вимагає жодних повідомлень про ключові точки під час навчання. Це може призвести до нестабільної роботи. Обмеження еквіваріантності є одним з найважливіші чинники, що спричиняють відкриття неконтрольованих ключових точок. Це змушує модель для передбачення узгоджених ключових точок щодо відомих геометричних перетворень. Використовуємо тонкі деформації сплайнів пластин, як вони раніше використовувалися для неконтрольованого виявлення ключових точок і подібні до природних деформацій зображення. Оскільки наш інструмент оцінки руху не тільки передбачає ключові точки, але також якобіани, ми розширюємо добре відому втрату еквіваріантності, щоб додатково включити обмеження якобіанів. Ми припускаємо, що зображення X зазнає відомої просторової деформації TX←Y. У цьому випадку TX←Y може бути афінним перетворенням або деформацією тонкого плоского сплайна. Після цієї деформації отримуємо а нове зображення Y. Тепер, застосувавши нашу розширену оцінку руху до обох зображень, ми отримаємо набір локальні наближення для TX←R і TY←R. Стандартне обмеження еквіваріації записується як:

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}} \equiv \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}} \circ \mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}} \quad (13)$$

Після обчислення розкладень Тейлора першого порядку обох сторін ми отримуємо наступні обмеження:

$$\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p_k) \equiv \mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}} \circ \mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p_k),$$

$$\left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right) \equiv \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}}(p)\Big|_{p=\mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p_k)}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right), \quad (14)$$

Зауважимо, що обмеження Eq. (11) суворо те саме, що стандартне обмеження еквіваріації для ключові моменти. Під час навчання ми обмежуємо кожне розташування ключової точки, використовуючи просту втрату L1 між двома частинами рівняння. Однак, реалізуючи друге обмеження з рівняння. L1 змусить величину якобіанів до нуля і призведе до чисельних проблем. До для цього ми переформулюємо це обмеження таким чином:

$$\mathbb{1} \equiv \left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right)^{-1}\left(\frac{d}{dp}\mathcal{T}_{\mathbf{X}\leftarrow\mathbf{Y}}(p)\Big|_{p=\mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p_k)}\right)\left(\frac{d}{dp}\mathcal{T}_{\mathbf{Y}\leftarrow\mathbf{R}}(p)\Big|_{p=p_k}\right)$$

(15)

де 1 – це ідентична матриця 2 × 2. Тоді втрата L1 використовується подібно до розташування ключової точки обмеження. Нарешті, під час наших попередніх експериментів ми помітили, що наша модель демонструє низьку чутливість до відносної ваги реконструкції та двох втрат еквіваріації. Тому ми використовуємо рівний схуднення в усіх наших експериментах.

3.4 Етап тестування: відносна передача руху На цьому етапі наша мета – анімувати об’єкт у вихідному кадрі S1, використовуючи відео D1,…DT. Кожен кадр Dt незалежно обробляється для отримання St. Замість того, щоб передавати закодований рух у TS1←Dt (pk) до S ми переносним відносний рух між D1 і Dt на S1. Іншими словами, ми застосувати перетворення TDt←D1(p) до околиці кожної ключової точки pk.

Інтуїтивно перетворюємо сусідні околиці кожної ключової точки pk у S1 відповідно до її локальної деформації у відео водіння. Справді, перенесення відносного руху на абсолютні координати дозволяє передавати лише відповідні моделі руху, при збереженні глобальної геометрії об’єкта. І навпаки, при передачі абсолютних координат, як в X2Face, згенерований кадр успадковує пропорції об’єкта ведучого відео. Це важливо зазначимо, що одним з обмежень передачі відносного руху є те, що нам потрібно припустити, що об’єкти у S1 і D1 мають схожі пози. Без початкового грубого вирівнювання рівняння може призвести до абсолютні ключові точки, фізично неможливі для об’єкта, що цікавить.

Ми представили новий підхід до анімації зображень на основі ключових точок і локальних афінних перетворень. Цій. Наша нова математична формулювання описує поле руху між двома кадрами і є ефективно обчислено шляхом отримання наближення Тейлора першого порядку. Таким чином, рух є описується як набір переміщень ключових точок і локальних афінних перетворень. Генераторна мережа поєднує зовнішній вигляд вихідного зображення та відображення руху відеозапису водіння. В Крім того, ми запропонували явно моделювати оклюзії, щоб вказувати на мережу генератора які частини зображення слід розфарбувати. Ми оцінили запропонований метод як кількісно, так і якісно і показав, що наш підхід явно перевершує сучасний рівень за всіма контрольними показниками.

Інший вид алгоритму для створення Deepfake – на основі генеративно-змагальних нейронних мереж. Нейронна мережа повністю змінюює цільове зображення в кожному кадрі відео. Для цього нейронна мережа спочатку має

визначити місцеположення обличчя відео. Далі на основі знайденого обличчя створюється тривимірна маска, на основі якої генератор створює нове обличчя, та замінює старе обличчя на нове. Цей метод дозволяє створювати більш якісний контент, адже нейронна мережа може обробляти великі кути поворотів голови, та майже повністю копіювати рух голови до зміни обличчя. До недоліків можна віднести те, що для кожної окремої людини потрібно вчити модель на основі даних про конкретне обличчя. Для цього потрібно зібрати декілька якісних відео, де буде знаходитись обличчя для генерації, також для створення якісних моделей тренувальний набір даних потрібно збирати максимально різним, з різними рухами голови, в різному освітленні, з невеликою різницею в обличчі (вуса, макіяж). Чим якісніша буде тренована модель, тим стабільніше вона буде працювати. До того ж, модель, що навчилась змінювати обличчя буде працювати з будь-яким обличчям, яке потрібно замінити. Хоча, якщо обличчя, на яке має переноситись обличчя жертви має значно інші параметри від тих, які вивчила модель, кінцевий результат може бути нестабільним. Модель буде давати значно більше «артефактів» на зображенні, за якими можна легко ідентифікувати Deepfake. Артефакти – незвичайні області зображення, в яких нейронна мережа помилилась, та створила неправильні зображення. Так, найбільш частий артефакт може бути помітним при носінні окулярів, сережок, через невелику кількість даних що містять такі предмети, нейронна мережа може мати мало інформації для їх коректного відображення.

Захист від Deepfake вимагає не тільки дослідження виявлення, але й розуміння методів генерації. Однак сучасні методи генерації страждають від неясного робочого процесу та низької продуктивності. Щоб вирішити цю проблему, ми опишемо DeepFaceLab, нині домінуючий фреймворк для заміни обличчя. Він забезпечує корисні інструменти, а також простий у використанні спосіб проведення високої якісна заміна обличчя. Він також пропонує гнучкий і вільний муфтова структура для людей, яким необхідно зміцнити їх конвеєр з іншими функціями без написання складно шаблонний код. Ми детально викладаємо принципи, які керують впровадження DeepFaceLab і введення його конвеєра, за допомогою якого кожен аспект трубопроводу може бути змінений безболісно для користувачів для досягнення мети налаштування. Примітно, що DeepFaceLab може досягти кінематографу. Якісні результати з високою

точністю. Ми демонструємо переваги- рівня нашої системи шляхом порівняння нашого підходу з іншими.

2.3 Генеративно-змагальна нейронна мережа для генерації Deepfake.

Оскільки глибоке навчання розширило сферу застосування комп'ютерного за останні роки, маніпулювання цифровими зображеннями, особливо маніпулювання зображеннями портретів людини, швидко покращилась та досягла фотореалістичних результатів у більшості випадків. Зміна обличчя – це завдання під час створення штучного вмісту шляхом перенесення вихідного обличчя до місця призначення зберігаючи рухи обличчя місця призначення та деформації виразу. Все більше і більше обличь, синтезованих StyleGAN [21], StyleGAN2 [22][23] стають все більш реалістичними і абсолютно нерозрізними для системи зору людини. Численні фальшиві відео, синтезовані на основі GAN методи заміни обличчя публікуються на YouTube і інші відео сайти. Комерційні мобільні додатки, такі як ZAO2 і FaceApp3, які дозволяють користувачам мережі створювати підроблені зображення та відео без особливих зусиль, значно збільшити поширення цих методів обміну, які називаються Deepfake. Ці технології генерації та модифікації контенту можуть вплинути на якість публічного дискурсу та порушити право громадян на персональні дані, особливо з огляду на дипфейк може використовуватися зловмисно як джерело дезінформації, маніпулювання, переслідування та переконання. Дослідження щодо виявлення медіа-підробок активні розвиваються і присвячуючи все більше зусиль для виявлення підробки обличчя. DFDC4 є типовим прикладом конкуренції на мільйон доларів який запустили Facebook і Microsoft. Для моделей виявлення підробок потрібні високоякісні підроблені дані. Дані, згенеровані різними методами, задіяні в DFDC [11] набір даних. Однак виявлення після генерації не є унікальним спосіб зменшення шкідливого впливу deepfake. Це завжди надто пізно виявити розповсюджуваний підроблений вміст.

Успіх DeepFaceLab пов'язаний із створенням попередніх ідеї в дизайн, який поєднує швидкість і простоту використання та бум комп'ютерного зору в розпізнаванні обличь. На відміну від інших систем заміни обличчя, DFL забезпечує повний інструмент командного рядка з кожним аспектом конвеєра які можуть бути реалізовані за бажанням користувачів. До певної міри DFL може функціонувати в режимі реального часу, і за допомогою стороннього

програмного забезпечення згенероване зображення на основі його рухів можна транслювати у сторонні додатки. Деякі практичні заходи були додані для підвищення продуктивності: підтримка кількох графічних процесорів, зменшення точності навчання за допомогою використання половини бітності даних, використання закріпленої пам'яті CUDA [24] для підвищення пропускної здатності, використання кількох потоків для прискорення графічні операції та обробку даних. Навіть а машина з 2GB VRAM також може успішно провести проект заміни обличчя. Щоб посилити гнучкість роботи DFL та залучати інтереси дослідницького співтовариства, користувачі можуть замінити будь-який компонент DFL, який це робить не відповідають їхнім вимогам. Більшість модулів DFL є розроблений як взаємозамінний. Наприклад, люди могли б забезпечити новіший детектор обличчя для досягнення більш високої продуктивності у виявленні обличь з екстремальними кутами або віддаленими ділянками. Наявність хороших наборів даних є важливою для завдання заміни. Як правило, чим більше набори даних, тим краще остаточні результати. Однак результати, які витягуються безпосередньо від вхідного зображення і вихідного зображення завжди з шумами, що значно шкодять кінцевій якості. З огляду на складну ситуацію DFL забезпечує ряд заходів для очищення наборів даних. Завдяки цим заходам DFL має масштабованість і може навіть підтримувати масивні набори даних і проводити заміну обличчя кінематографічної якості на основі великих набори даних.

DeepFaceLab надає набір робочих процесів, які формують гнучкий робочій процес. У DeepFaceLab ми можемо розділити конвеєр на три фази: видобуток, навчання, і перетворення. Ці три частини представлені послідовно взаємно. Крім того, примітно, що DFL відноситься до парадигма обміну обличчями один на один, що означає, що є лише два види даних: вхідні і вихідні.

Фаза екстракції є першою фазою в DFL, метою якої є витягти обличчя з даних. Ця фаза складається багатьох алгоритмів і частин обробки, тобто розпізнавання обличчя, вирівнювання обличчя та сегментації обличчя. DFL забезпечує багато режимів екстракції (тобто, напівлицеве, повне обличчя, ціле-обличчя), що представляє зону покриття обличчя. Як правило, ми використовуємо повний режим за замовчуванням.

Розпізнавання обличчя. Першим кроком на етапі вилучення є пошук цільова грань у наведених даних. S3FD як стандартний детектор обличчя. S3FD змінити на інший доступний в реалізації алгоритм, наприклад RetinaFace [25].

Або застосовувати будь-який власний компонент, що вимагає незначної зміни вихідного коду.

Вирівнювання обличчя. Другим кроком є вирівнювання обличчя. DFL надає два канонічних типи орієнтирів обличчя. Алгоритми вилучення для вирішення цієї проблеми: (a) на основі теплової карти алгоритм орієнтира 2DFAN (для граней зі стандартною позою) і (b) PRNet [26] з попередньою інформацією про тривимірне обличчя (для грані з великим кутом Ейлера (рискання, крок, крен), наприклад, грань з великим кутом відхилення, означає, що одна сторона обличчя знаходиться назовні зору). Після отримання орієнтирів на обличчі ми також навели додаткову функцію з настоюваними кроком часу до плавних лицьових орієнтирів послідовних кадрів, щоб забезпечити подальшу стабільність. Потім ми приймаємо класичне точкове відображення і метод перетворення, запропонований Умеямою для обчислення скласти матрицю перетворення подібності, яка використовується для вирівнювання обличчя мент. Як метод, запропонований для розрахунку потрібні стандартні шаблони орієнтирів обличчя матриця перетворення подібності, DFL забезпечує каноні- вирівняний шаблон обличчя орієнтир. Примітно, що DFL може автоматично передбачити кут Ейлера за допомогою отриманих орієнтирів обличчя.

Сегментація обличчя. Після вирівнювання обличчя папка даних зі стандартним виглядом спереду/збоку (вирівняний вхідний набір або вирівняний вихідний набір). Ми використовуємо дрібнозернисте обличчя. Мережа сегментації через які обличчя з волоссям, пальці або окуляри можуть бути точно сегментовані. Це корисно для видалення неправильних оклюзій, щоб зберегти мережу в процесі навчання більш стійкою до рук, окулярів і будь-які інших предметів, які можуть якось закривати обличчям.

Фаза навчання відіграє найважливішу роль у досягненні фотореалістичні результати заміни обличчям DFL. Немає потреби у виразах обличчям вирівняних у вхідних та вихідних даних, будучи суворо узгодженим, DFL прагне надати парадигму ефективного алгоритму для вирішення цієї непарної проблеми разом із збереженням високої точності та якості сприйняття створеного обличчям. Мотивований попередніми методами, ми пропонуємо дві

структури, структуру DF та структуру LIAE для вирішення цього питання. Узагальнення вхідних та вихідних даних досягається за допомогою спільний енкодеру, який вирішує можливі потенційні проблеми з неузгодженими входами даних. DF структура може закінчити завдання заміни обличчям, але не може успадкувати достатньо інформації від вихідних даних, наприклад освітлення. Щоб ще більше посилити проблему світлової консистенції, ми запропонувати LIAE. Як показано, структура LIAE є більш складною структурою з кодувальником зі спільними вагами, денкодер і два незалежних енкодери. Що стосується функції втрат, то DFL використовує змішані втрати (DSSIM (структурні несхожості)) за замовчуванням. Ця комбінація має отримати переваги від обох: DSSIM швидше вирівнює людські обличчям, а MSE забезпечує кращу чіткість. Ця комбінація втрат слугує для пошуку компромісу. Між узагальненням і наочністю.

Останній розділ є фазою перетворення, коли вихідні зображення обробляються нейронною мережею, знаходяться обличчям і за допомогою тренованої моделі замінюються на вхідні обличчям

Висновки до розділу 2

Розглянуті два основних алгоритми, які використовуються для генерації deepfake відео. Незважаючи на те, що реалізація та принципи роботи в них значно відрізняються, вони показують гарні результати роботи.

Перший алгоритм значно простіше в використанні і не потребує додаткового навчання, що значно полегшує його використання в різноманітних цілях. Але це також накладає певні обмеження до контенту, який може бути згенерований за допомогою нього. Так, як було описано в цьому розділі, незважаючи на покращення роботи алгоритму в зонах, в яких ми не можемо мати явну інформацію про об’єкт (оберти голови), наявний алгоритм лише апроксимізує цю інформацію з інформації, яку він отримав під час єдиного тренування. Тому в таких зонах робота алгоритму незадовільна. В тім, для анімації простих рухів або статичних і майже статичних обличь він підходить краще ніж другий алгоритм.

Другий алгоритм має складнішу будову і застосування. Так він потребує початковий набір даних з обличчям, яке потрібно замінити і обличчям, яке буде замінене. Також він потребує тренування на основі цих обличь, що накладає певні обмеження на пристрої, які мають тренувати це. Так, авторами цієї

роботи було протестовано швидкість тренування на слабкому графічному процесорі. Оцінка часу тренування на основі двох двохвилинних відеозаписів складала близько двох діб. При наявності більш потужних графічних прискорювачів, очевидно, швидкість тренування зросте. Проте, тренування не єдина проблема, яка може з'являтись при використанні цього підходу. Так, якщо користувачі намагаються замінити обличчя, яке за антропоморфними даними значно відрізняються то можуть виникати нестабільні результати. Також нестабільні результати можуть виникати при значній зміні параметрів інших відеозаписів, на які потенційно може бути використана навчена модель. При значній зміні освітлення або недостатній кількості складних прикладів в тренувальному наборі даних (відсутні складні рухи головою, зміна освітлення, зміна зачіски, елементів на обличчі) такий алгоритм теж може показати нестабільні результати.

Таким чином, кожний алгоритм створення deepfake має свої переваги і недоліки. З цих параметрів і випливають сфери застосування, в яких один алгоритм має перевагу над іншим. У випадку потреби швидкої генерації статичних і майже статичних обличь перший метод має більше переваг для використання ніж другий. У випадку коли більшу важливість має якість генерації та потреба застосовувати генерацію для декількох відеозаписів, а також у випадку генерації більш складних прикладів рухів перевагу потрібно віддавати другому методу.

РОЗДІЛ 3. ДОЦІЛЬНИЙ СПОСІБ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ У ФОРМАТІ «DEEP FAKE».

3.1 Теоретичні основи алгоритму визначення Deepfake.

Алгоритми визначення Deepfake контенту почали розвиватись, коли дослідники зрозуміли, що якість згенерованого контенту зросла до такої, що на сьогоднішній час згенерований контент важко відрізнити від реального, якщо спеціально не намагатись знайти невідповідності або артефакти на зображенні. Найкращим алгоритмом з визначення згенерованого штучним інтелектом контенту буде застосування іншого штучного інтелекту. Задачу розпізнавання Deepfake можна вирішувати різними способами, із залученням різноманітних алгоритмів штучного інтелекту, але для кращої роботи може знадобитись використання декількох алгоритмів, які доповнюють один одного. Так, у найпростішому випадку можна тренувати згорткову нейронну мережу для бінарної класифікації зображень. Тоді при підозрілому відео можна запустити класифікацію кадрів, та дізнатись, яка кількість кадрів є згенерованою. Цей метод є найпростішою реалізацією алгоритму детекції, не потребує великих ресурсів для тренування і роботи, проте до недоліків можна віднести необхідність розбивання відео на кадри (можна використовувати не всі кадри, а пропускати певну кількість кадрів, адже частіше всього інформація в сусідніх кадрах мало відрізняється), а також необхідність обробляти кожний кадр, або сукупність кадрів, що може тривати довший час.

Більш просунутим способом можна запропонувати навчити нейронну мережу на невеликих відрізках відео, і класифікувати контент вже за допомогою відео. До переваг такого методу можна віднести те, що відео містить значно більше інформації про обличчям, навіть якщо воно майже статично розмовляє, а також, як ми знаємо з недоліків алгоритмів Deepfake, більша кількість кадрів, що проходять через нейронну мережу класифікації може означати більшу ймовірність того, що на відривку відео генератор контенту помилився, та неправильно намалював контент, тож, ці ознаки класифікуюча нейронна мережа може використовувати як сильну ознаку того, що контент несправжній. До недоліків підходу відносяться більш складна нейронна мережа, що приймає на вхід відео, довша тривалість тренування та більші вимогу до апаратної частини. Також, потрібна більша кількість експериментів, для порівняння якості роботи алгоритму з відео різної тривалості. Залишається також попередній недолік: необхідність повторення

процесу «нарізання» відео на фрагменти для класифікації. Проте, як і в попередньому випадку, при достатньо потужному залізі та достатній оптимізації роботи алгоритмів можна зменшити, проводячи обробку в режимі реального часу, або з мінімальними затримками. Або ж проводити перевірку контенту на етапі модерації, коли автор намагається викласти відео в мережу Інтернет з використанням відомих платформ для перегляду відео.

Іншим потенційним алгоритмом для визначення може стати архітектура, що нагадує генеративно-змагальну мережу, в якості генератора якої буде використовуватись існуюча модель генерації контенту, що також складається з генератора і дискримінатора, а в якості дискримінатора першої мережі – спеціально створений дискримінатор. На відміну від оригінальної задачі тренування генеративно-змагальної нейронної мережі, цю задачу потрібно вирішувати протилежно – тренувати дискримінатор, використовуючи заздалегідь тренований генератор. Експерименти з такою архітектурою виходять за рамки даної роботи через потребу у великій кількості обчислювальних потужностей та високої їх вартості. Проте, можна описати даний підхід теоретично. Так, вважаємо, що в якості генератора у нас готова архітектура з двох нейронних мереж – G. В якості дискримінатора – згорткова нейронна мережа, що на вхід отримує зображення з генератора, або з тренувального набору даних, і на вихід передає ймовірності належності до одного з двох класів – реальне або згенероване зображення.

Тренувальний набір використовується також в генераторі, який змінює реальне обличчя на довільне, завантажуючи різні готові моделі з різними обличчями. Далі використовуємо звичайний процес тренування мережі для заміни обличь. Таким чином, дискримінатор, отримуючи на вхід вже якісну маску обличчя має навчитись розрізняти фейковий та реальний контент. Враховуючи те, що обличчя, які мають подаватись на вхід будуть належати різним людям, це дозволить дискримінатору не підлаштовуватись під певне обличчя, що дозволить використовувати його на довільних прикладах. Для кращої якості також можна використовувати різні приклади генераторів, а також використовувати аугментації зображень, що містять артефакти, подібні до генеративних мереж. Через те, що такі перетворення зображення важко створити програмними методами, можливо, буде потрібно створити окрему генеративну мережу, яка б просто спеціальним чином створювала б специфічні артефакти, що притаманні більшості генеративних мереж. Через те, що поки не

існує відкритих наборів даних для тренування, потрібно буде створювати власний набір даних з ручною розміткою. Тож, прикладне створення подібних алгоритмів, які б показували б гарну якість роботи потребує значних ресурсів, як обчислювальних, так і потенційно людських, для розмітки необхідної кількості даних для тренування додаткових компонентів системи. Тож, в цій роботі ми сфокусуємося на більш простих методах визначення фейкового контенту.

Так, для визначення згенерованого відео можна використовувати властивості людських обличь та рухів м'язів. Встановлено, що люди зазвичай кліпають 15-18 разів на хвилину. Створення простої системи, яка рахує середню кількість закриття очей у людини на відеозаписі може приблизно дати розуміння, що такий контент має викликати підозру, та має бути перевірений більш детально. Deepfake алгоритми часто мають проблеми з відображенням природнього кліпання очей. Це пов'язане з особливостями тренування генеративних мереж – кількість кадрів з відкритими очима значно перевищують кількість кадрів з закритим очима та моментами переходів від закритих до відкритих очей. Тренувальний набір зазвичай створюють з окремих кадрів відео і зазвичай до нього надходять не всі кадри через відсутність великої кількості змін у сусідніх кадрах, тому ефективніше брати більш відмінні кадри. Тому можна вважати, що алгоритми на основі визначення закриття очей може бути ефективним рішенням для визначення ранніх алгоритмів Deepfake. Для більш досконалих алгоритмів генерації такий алгоритм може бути вже не ефективним. Якщо генерація буде враховувати реальну частоту кліпів очима, та правильно її створювати, очевидно, що визначення працювати не буде.

Відстеження рухів м'яз на обличчі, виявлення незвичайних патернів рухів або їх відсутності може допомогти з першочерговим визначенням потенційних кандидатів, які мають бути згенеровані. Досить складно запрограмувати машину на відстеження та аналіз рухів кожного м'язу на обличчі. Алгоритм, який би рахував середню кількість закриття зіниць за хвилину доволі просто реалізувати без штучного інтелекту – достатньо визначити та відстежувати позиції очей та стежити за зміною картинки: при кліпанні різниця між зіницею та віками буде значна, навіть якщо для порівняння фотографій використовувати найпростіший метод суми різниць пікселів. Для визначення рухів м'язів очевидного алгоритму вже не існує, і його створення буде доволі складною

обчислювальною задачею. В цьому випадку розв’язати її допоможе застосування алгоритмів штучного інтелекту. Автоматичний аналіз зображення штучним інтелектом, в якому модель сама буде навчатись розрізняти рухи м’язів обличчя.

Одним з перспективних методів захисту від використання своєї біометрії в deepfake алгоритмах є використання недосконалості нейронних мереж [53]. А саме, нейронна мережа використовує інформацію з пікселів для прийняття рішень. Але якщо спеціальним чином змінити пікселі, для людини це буде непомітним, але нейронна мережа не зможе правильно працювати. Це називається змагальна атака. Використання такого алгоритму очевидно просте – користувач перед завантаженням фото в мережу Інтернет застосовує до нього алгоритм змагальної атаки, змінюючи інформацію в пікселях, таким чином, щоб при використанні розглянутих в розділі 2 алгоритмів генерації deepfake вони не змогли визначити обличчя, або робили це менш якісно. Скоріше всього, це не гарантуватиме 100% захист від створення deepfake, але значно ускладнить його, і зменшить якість вихідного відео, що дасть підстави з більшою ймовірністю розрізнити таке відео не вдаючись до складного аналізу.

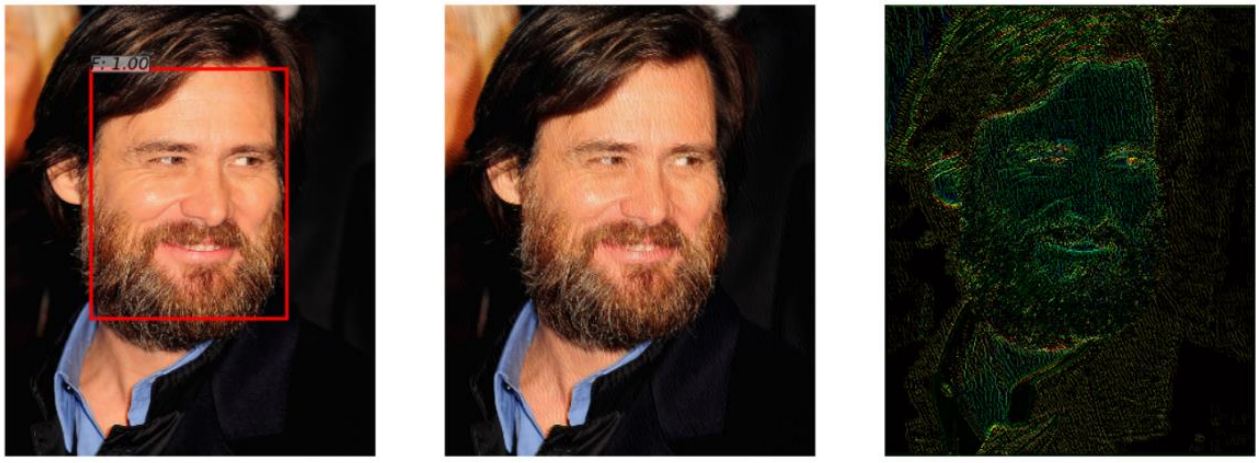

Рисунок 11. Приклад атаки на мережу розпізнавання обличчя

Атаки на нейронні мережі можна також використовувати як звичайні хакерські атаки – для отримання вигоди, або завдання збитків організаціям, які використовують алгоритми штучного інтелекту. Захист від таких атак доволі складний, адже розуміння того, як нейронна мережа приймає рішення ми маємо лише приблизний, тому важко сказати, в якому місці потрібно посилювати захист мережі. Одним з дієвих методів захисту можна вважати аугментації зображення під час тренування з урахуванням можливих змагальних атак. Типовий алгоритм можна описати так:

Розробники тренують необхідну нейронну мережу на основі зібраного набору даних. Далі згідно політики безпеки визначаються напрямки використання і список можливих атак, які можуть бути використані проти готової нейронної мережі. Створюються приблизні алгоритми атаки, на основі того як і де можуть діяти зловмисники. Алгоритми атак створюються як з урахуванням того, що зловмисники мають дані про архітектуру та методи навчання нейронної мережі, яку необхідно захистити (білий ящик) так і коли зловмисники нічого не знають про завищувану нейронну мережу (чорний ящик).

Далі з урахуванням цих атак деяка кількість тренувальних даних змінюється згідно визначених алгоритмів і тренується інша, ідентична до першої нейронна мережа [31]. У випадку подібного тренування, нейронна мережа вивчить можливі патерни атаки та буде менш схильна до помилки у разі реальної атаки [30].

Алгоритм такої атаки дуже схожий на генеративно-змагальні мережі [27] [28]. Хоча, у деяких випадках ГЗМ також можна використовувати для проведення змагальних атак. Хоча більшість таких методів розрахована на те, що ми знаємо яку архітектуру нейронної мережі потрібно атакувати, в реальному світі ми не будемо знати, проти яких архітектури потрібно захистити зображення. Тож, у цій роботі сфокусуємось на атаках на моделі, що є чорними ящиками. Таким чином, лишається метод узагальнення моделі чорного ящика та метод на основі генеративно-змагальних нейронних мереж [29].

Метод узагальнення чорного ящику оснований на тому, що ми маємо доступ до результатів роботи моделі. Відправляючи дані до атакованої моделі та отримуючи результат можна створити набір даних, на якому можливо навчити власну модель, тим самим узагальнюючи атакуючу модель. Після цього можна використовувати атаки на білий ящик узагальненої моделі, адже ми створили схожу модель на основі інформації, яку отримали від невідомої моделі. Ймовірність такої атаки більша, коли ми маємо більше інформації про атакуючу модель.

Метод з застосуванням генеративно-змагальних мереж для захисту фотографій є більш складеним, про і більш ефективним. За допомогою генератора можна змінювати об’єкти, які потім передаються до невідомої моделі, а її відповіді використовують для тренування генератора. Також важливо додати метрику зміни фотографії, яку в процесі навчання генератора

також потрібно мінімізувати. Тобто, завданням генератора буде створення найбільш ідентичної фотографії, яка б значно відрізнялась в розумінні іншої нейронної мережі від обличчя людини.

Такі методи можуть стати більш популярним рішенням, ніж алгоритми визначення через їх більшу простоту для користувача. Простіше сказати користувачу, що «Наш алгоритм здатен захистити вашу фотографію від неправомірного використання», ніж потім наводити докази, що відеозапис з його участю є згенерованим. Те саме стосується і виступів політиків, і публічних діячів, які завантажують свої відеозаписи в мережу Інтернет.

Але в реальному житті цього буде також недостатньо. Існує велика кількість вже відзнятих матеріалів, які все ще доступні для використання нейронними мережами, також можуть бути створені кращі алгоритми для детекції та розпізнавання обличь. Але найбільшою проблемою можуть стати самі deepfake алгоритми. Вони також можуть вільно використовувати змагальні атаки для захисту від алгоритмів визначення deepfake. Тому це ще більше ускладнює роботу дослідників та розробників алгоритмів захисту від згенерованого контенту [32]. Незважаючи на те, що наразі більшість таких алгоритмів знаходиться у відкритому доступі, і у дослідників та розробників є можливості з ними вільно ознайомлюватись та тестувати їх в різних умовах, і застосовувати атаки і захист методами для білого ящика, в найближчому майбутньому можуть з’явитись нові алгоритми з закритим вихідним кодом, які значно перевершать всі існуючі сьогодні рішення.

У випадку застосування алгоритмів класифікації на основі штучного інтелекту важко сказати, за якими ознаками він класифікує зображення. Іншими словами, його рішення неочевидне для людини з точки зору розуміння алгоритму класифікації. Це пов’язане з тим, що людина не приймає жодної участі в створенні алгоритму класифікації, який являє собою складну математичну функції. Вона створюється ітеративним шляхом в процесі тренування нейронної мережі, та апроксимує функцію прийняття рішень на розподілі даних. Ми можемо лише приблизно сказати, які області зображення більш важливі для прийняття того чи іншого рішення, проте можна приблизно дізнатись певні «цікаві» області зображення з точки зору штучного інтелекту. Аналіз цих областей може дати додаткове розуміння задачі, а також методів покращення її розв’язку. Наприклад, видалення областей фотографій в яких нейронні мережі не бачать цікавих областей. Це може покращити результати

роботи, проте зменшення кількості вхідних даних також збільшить швидкість роботи та тренування мережі [13].

Для створення алгоритму протидії на основі інших моделей штучного інтелекту потрібно зібрати набір даних для тренування моделей. В створенні штучного інтелекту одним з найважливіших пунктів є формування початкового набору даних. Він має максимально повно відображати генеральну сукупність даних, з якими в майбутньому буде працювати нейронна мережа. Очевидно, що навчання на генеральній сукупності даних неможливе, тому потрібно створити максимально репрезентативний набір даних, та дати нейронній мережі правильну задачу. Вона являє собою створення правильної функції втрат, яка б давала нейронній мережі змогу вчитися вирішувати певну задачу за допомогою різних методів. З метою покращення результатів навчання існує безліч методів, розгляд всіх із них виходить за межі даної роботи. Але другим важливим методом після створення набору даних є перевірка якості роботи нейронної мережі. Через те, що розробники намагаються використати якомога більше даних для тренування, потрібно також залишити певну кількість даних, для яких доступні правильні відповіді для перевірки моделі перед використанням. Правильна перевірка дає змогу зрозуміти, як буде працювати нейронна мережа перед її реальним застосуванням. Правильно підібрана архітектура, параметри навчання та набір даних дозволяють отримати найкращі результати роботи фінальної версії моделі. Архітектура та параметри знаходяться за допомогою теоретичних та практичних знань інженерів, як і частина параметрів. Інша частина параметрів змінюється після отримання попередніх результатів роботи нейронної мережі. Отримання готової моделі може зайняти чимало часу та забрати велику кількість обчислювальних ресурсів. Тому зазвичай, тренування моделі deepfake для роботи з одним обличчям може займати від доби до тижня на персональному комп’ютері з останніми моделями графічних прискорювачів, або моделями попередніх поколінь. Наразі вже існують відкриті набори даних, які дозволяють використовувати їх для тренування власних моделей для визначення. Проте, зважаючи на легкість та доступність використання існуючих Deepfake-моделей, можна створити власний набір даних, під конкретні випадки, або ж доповнювати існуючі набори даних власними.

3.2 Практична розробка алгоритму визначення згенерованого контенту.

Для розв’язання задачі визначення перенесення обличчя на відео ми будемо використовувати набір даних з Deepfake Detection Challenge. Він представляє собою набір відеозаписів загальним обсягом у 472 Гб. В цій роботі не буде використаний весь набір даних через велику обчислювальну складність у обробці такої кількості даних та високу ціну обчислень. Буде використана частина, розміром близько 10 Гб. Приклад даних з набору зображено на рисунку.

Рисунок 12. Приклад даних для тренування ШНМ

Далі потрібно витягнути фотографії обличчя з відеозаписів, які і будуть використовуватись для тренування детекторів. Для визначення фотографій можна використовувати різноманітні детектори обличь, більшість з яких є програмами з відкритим вихідним кодом та працюють приблизно з однаковою точністю у більшості випадків роботи. В цій роботі буде використано бібліотеку з відкритим вихідним кодом на мові програмування Python [18], яка дозволяє застосовувати dlib [54], який теж є бібліотекою з відкритим вихідним кодом побудована на C++ [55], і дозволяє застосовувати швидкі моделі машинного навчання для різноманітних завдань. Як видно на малюнку, визначення обличчя працює як з реальними відео так і зі згенерованими.

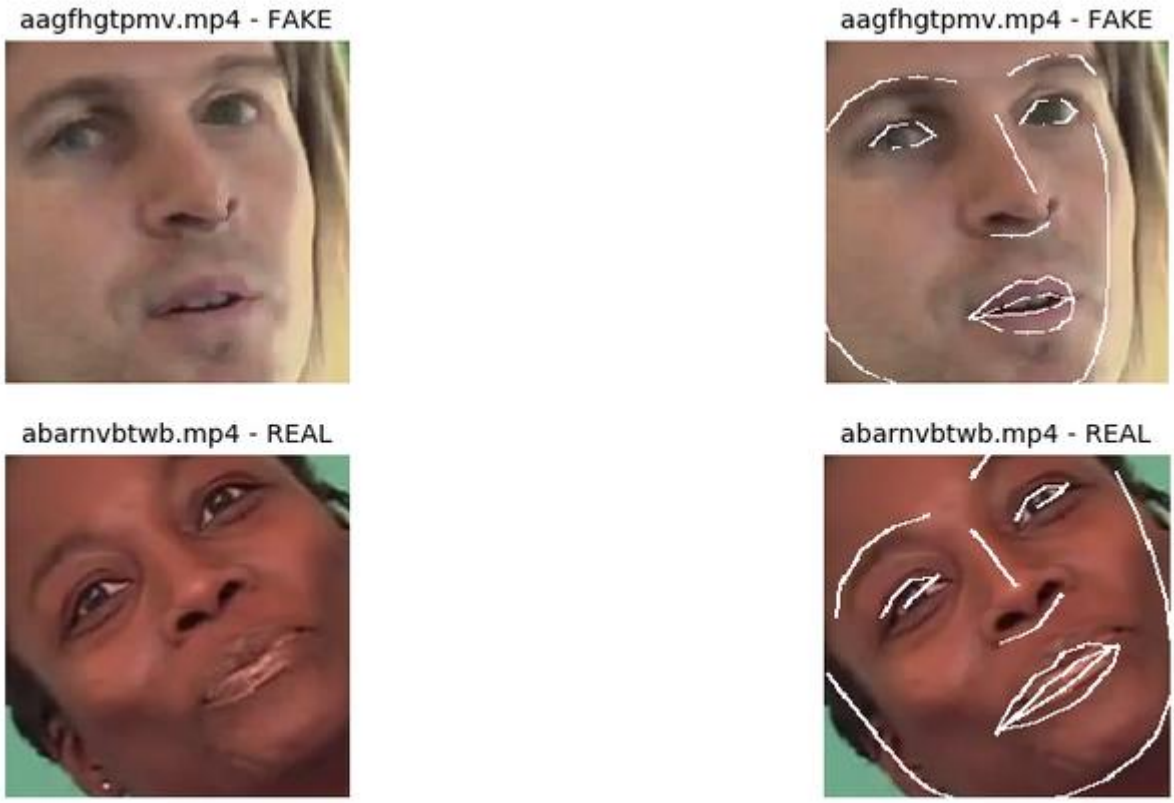

Рисунок 13. Приклад визначення обличчя для згенерованих прикладів

Далі необхідно провести обробку відеозаписів та створити окремий набір даних, який міститиме лише зображення обличчя фіксованого розміру. Таким чином ми будемо вчити бінарний класифікатор, який і буде визначати справжність відеозаписів.

В якості архітектури нейронної мережі були протестовані деякі популярні архітектури. Використовувались також заздалегідь навчені на відкритому наборі даних архітектури, в яких було замінено останній шар і проводилось донавчання на новому наборі даних. Були використані EfficentNet, VGG, ResNet та Xception [14].

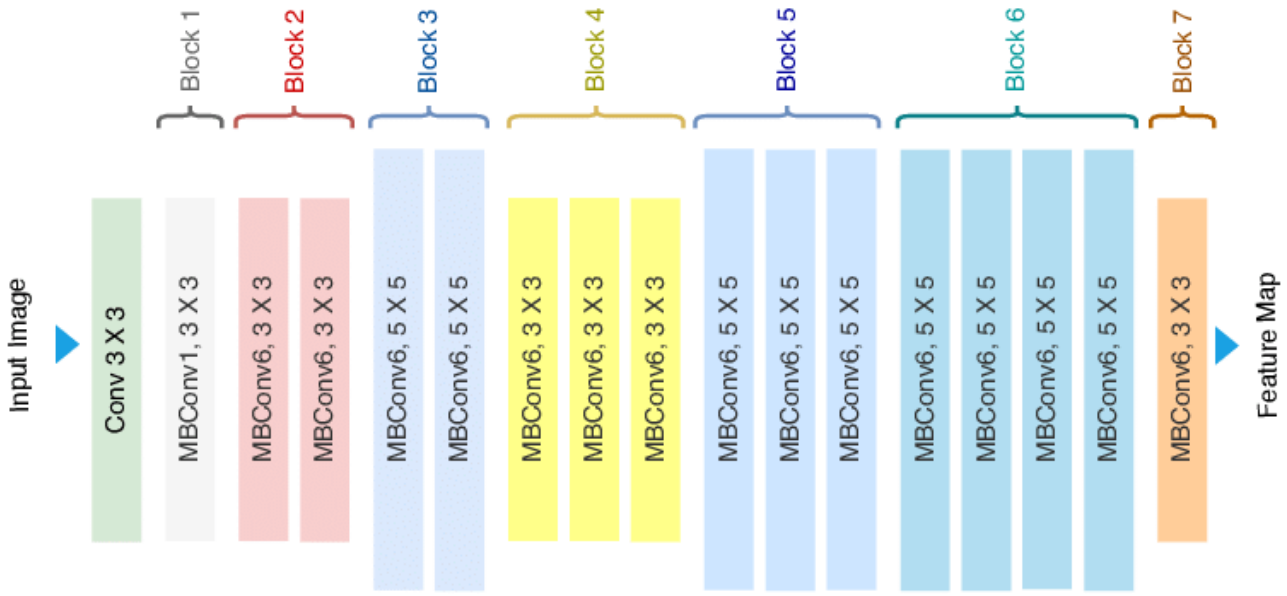

Рисунок 14. Архітектура нейронної мережі

В якості тренувального серверу було обрано хмарне рішення Google Colab Pro [56]. Він дозволяє отримати виділене середовище розробки, в якому є 4-ядерний процесор, до 25 Гб оперативної пам’яті та графічний прискорювач Nvidia Tesla V100 [57] або Nvidia Tesla P100 [58] з 16 Гб пам’яті. Було використано також додаткові перетворення зображення під час тренування, такі як оберти зображення, зміна колірної схеми, вирізання випадкової області. Для тренування не використовувались дані з аудіо доріжок, які були наявні у відеозаписах. Результати тренування моделей наведено в таблиці.

Таблиця 1. Результати тренування ЗГН

Модель	Точність, %
ResNet	62.48
Xception	73.84
EfficentNet	78.05
VGG	59.84

З результатів тренування моделей видно, що найкраще в нашому випадку впоралась модель EfficentNet. Але так як через брак необхідних обчислювальних ресурсів для тренування було використано лише частину набору, яка може бути не репрезентативною відносно всього тренувального набору даних, дані про точність після тренування лише приблизні.

Для перевірки результатів моделі було написано скрипт на мові програмування Python [18], з використанням бібліотек для глибокого навчання Pytorch, Blazeface. Для прискорення обчислювань їх було перенесено на графічний прискорювач за допомогою CUDA – універсальної обчислювальної архітектури, створеної компанією Nvidia для наукових розрахунків та використання в глибокому навчанні на своїх графічних прискорювачах. Це дозволяє значно пришвидшувати навчання та роботу моделей глибокого навчання через їх потребу в великій кількості матричних обчислень. Як відомо, графічні прискорювачі також швидко можуть обробляти матричні дані для зображення, але головна їх перевага перед центральними процесорами в тому, що графічний прискорювач здатен швидко робити велику кількість паралельних обчислень, що корисно при тренуванні моделі та значно пришвидшує швидкість навчання. В якості архітектури нейронної мережі буде використаний EfficentNet [59]. Дані для описаного прикладу було взято з набору даних зі змагання з розпізнавання згенерованих відеозаписів.

В прикладі роботи алгоритму ми використаємо зображення з реальною особою і згенерованою.

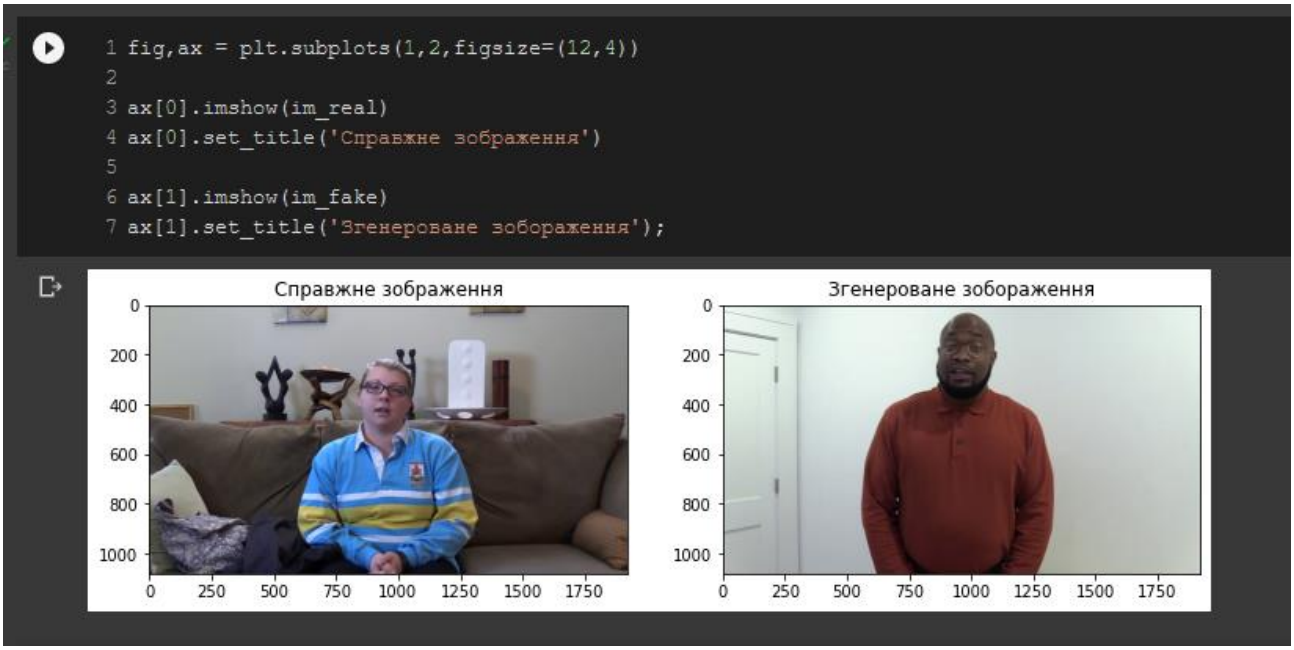

Рисунок 15. Приклад роботи нашого алгоритму

Далі ми визначаємо області в яких знаходяться обличчя та виділяємо їх для подальшого використання, оскільки нейронна мережа для визначення deepfake натренована лише на даних які містять обличчя.

```
1 fig,ax = plt.subplots(1,2,figsize=(8,4))
2
3 ax[0].imshow(im_real_face)
4 ax[0].set_title('Справжне зображення')
5
6 ax[1].imshow(im_fake_face)
7 ax[1].set_title('Згенероване зображення');
```

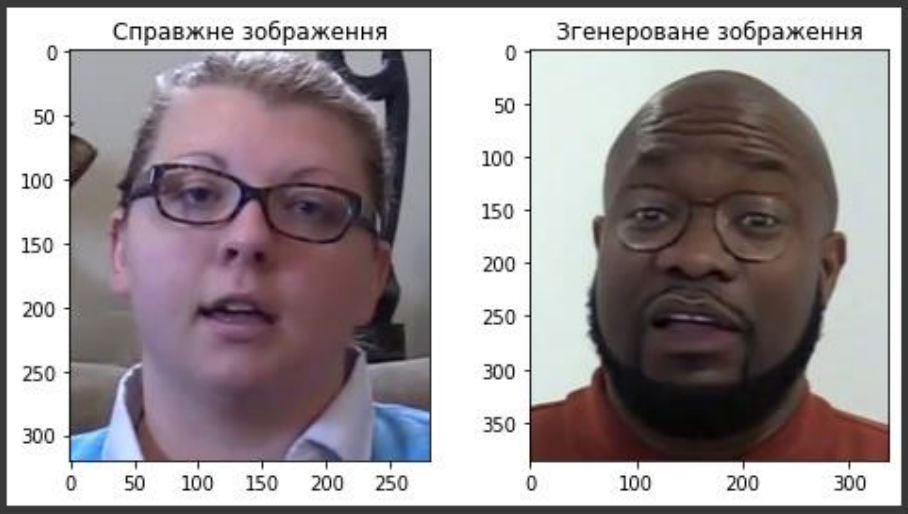

Рисунок 16. Приклад роботи алгоритму

Після цього передаємо обличчя до нейронної мережі, яка на виході має показати ймовірності для класів «Реальне зображення» і «Згенероване зображення».

```
[17] 1 faces_t = torch.stack( [ transf(image=im)['image'] for im in [im_real_face,im_fake_face] ] )
     2
     3 with torch.no_grad():
     4     faces_pred = torch.sigmoid(net(faces_t.to(device))).cpu().numpy().flatten()

[18] 1 """
     2 A score close to 0 predicts REAL. A score close to 1 predicts FAKE.
     3 Результати ближчі до 0 - Реальні, до 1 - Згенеровані
     4 """
     5 print('Значення для реального зображення: {:.4f}'.format(faces_pred[0]))
     6 print('Значення для згенерованого зображення: {:.4f}'.format(faces_pred[1]))

Значення для реального зображення: 0.0107
Значення для згенерованого зображення: 0.8372
```

Рисунок 17. Результати роботи алгоритму

Як бачимо, нейронна мережа правильно класифікувала зображення. Для отримання результатів роботи нейронної мережі ми використовуємо сигмоїдну функцію активації для отримання результатів. Вона використовується як функція активації в останніх шарах нейронних мереж. Результати роботи функції можна вважати ймовірностями того, що дані належать певному класу. Через те, що нейронна мережа має один вихід, сігмоїда дозволяє отримати

значення класу. Чим ближче сігмоїда до своїх крайніх значень, тим ймовірніший клас. А так як значення сигмоїди обмежені інтервалом [0, 1] то отримані значення відповідають правильним класам.

У випадку з обробкою відеозапису, ми обробляємо кожний кадр окремо, отримуючи рахунок для кожного кадру відео. Таким чином можна вивести середню впевненість у тому, що відеозапис є згенерованим, або використовувати медіану в якості єдиного значення, яке відповідає справжності відео.

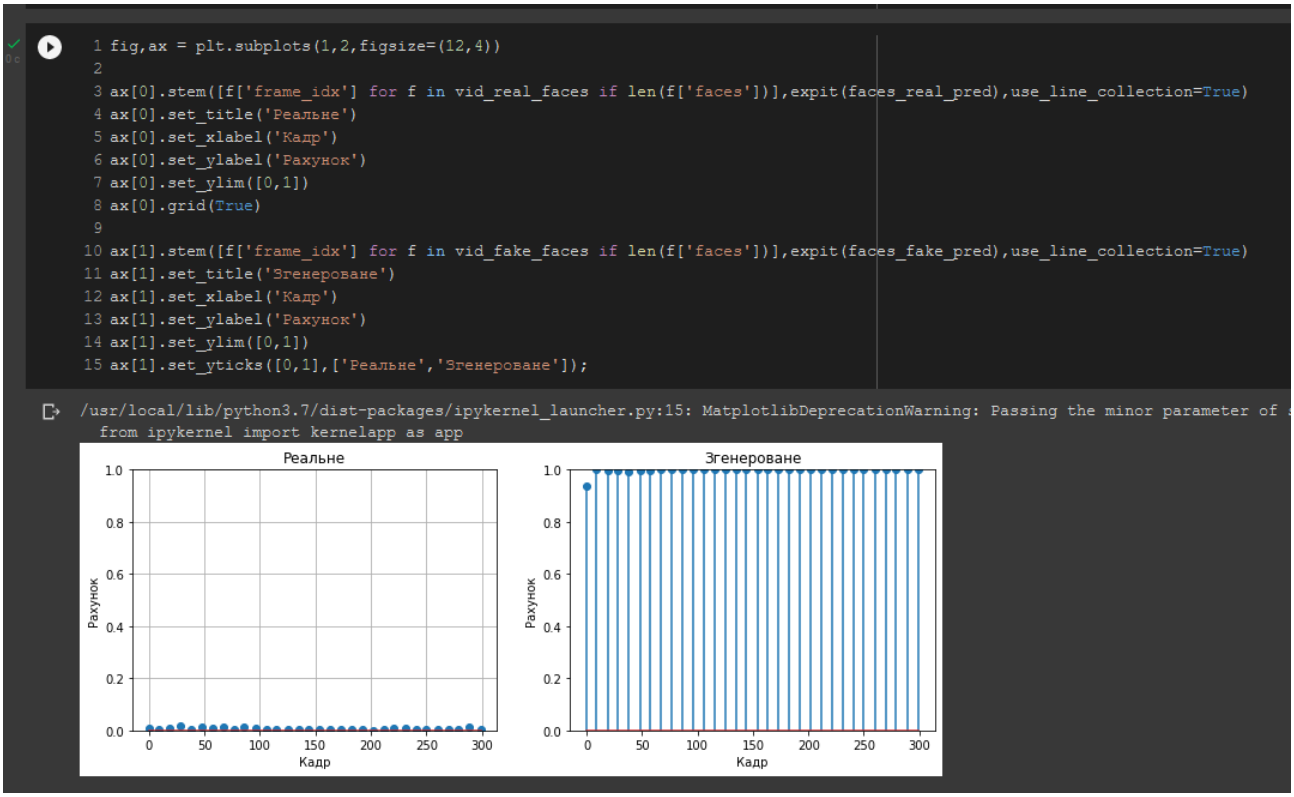

Рисунок 18. Результати обробки відеозапису

Я

к показано на малюнку, всі значення впевненості для реального відео знаходяться ближче до 0. Що означає, що кадр з більшою ймовірністю є реальним. Значення для згенерованого відеозапису знаходяться ближче до 1.

Варто сказати, що реалізація виявилась досить швидкою на серверному графічному прискорювачі Nvidia Tesla V100 [8]. Зазвичай, подібні серверні прискорювачі обмежені у частоті центрального процесору для зменшення виділення тепла в серверних приміщеннях. А тому, відеокарти серії Nvidia GTX та Nvidia RTX можуть бути навіть швидшими за серверні рішення, адже їх частоти не обмежені виробником. Відеокарти інших виробників не підтримуються рішеннями глибокого навчання повністю через відсутність на них рішень CUDA. Також, можлива подальша оптимізація роботи алгоритму

для прискорення та зменшення витрат пам’яті для реальних користувачів, які потенційно будуть використовувати програму на своїх персональних комп’ютерах.

Далі розглянемо рішення переможця конкурсу з визначення Deepfake – Deepfake detection Challenge [10]. Так, переможець використовував для визначення областей обличь MTCNN. В якості моделей також були використані моделі сімейства EfficentNet. Одним з найважливішим підходом в цьому рішенні було намагання автора навчити модель обробляти явні візуальні артефакти на обличчі, які виникають після проходження відеозаписом алгоритмів створення deepfake. Хоча, як заявляв сам автор, і як видно з наших результатів тренувань моделі добре навчаються визначати візуальні артефакти без будь-яких інших доповнень та намагань зі сторони розробника. Але цього теж було недостатньо, тому що моделі можуть перенвачатись на певних візуальних артефактах одного типу, який може бути не досить релевантним відносно всього набору даних, або даних з якими будуть працювати моделі в майбутньому. Тому додатково автор використовував видалення частини обличчя. Для цього потрібно визначити ключові координати обличчя. Далі створюючи повну маску точок обличчя можна випадковим чином видалити або замалювати певну частину обличчя, щоб модель не так фокусувалась на певних рисах обличчя, а намагалась більше генералізувати можливі потенційні артефакти, що можуть виникати в різних областях обличчя. Також, таким чином модель може навчитись виявляти артефакти значно меншого розміру, які можуть навіть не помічати люди. Також це може допомогти моделі навчитись розрізняти інші параметри, які можуть бути неочевидні для людини і які можуть значно відрізняти згенероване і реальне відео. До таких параметрів можна віднести колірну схему реальних відеозаписів та відеозаписів, які пройшли через алгоритми. Також це може бути інші зміни в пікселях обличь, які пройшли редагування алгоритмами deepfake. Приклад створення масок показаний на рисунку.

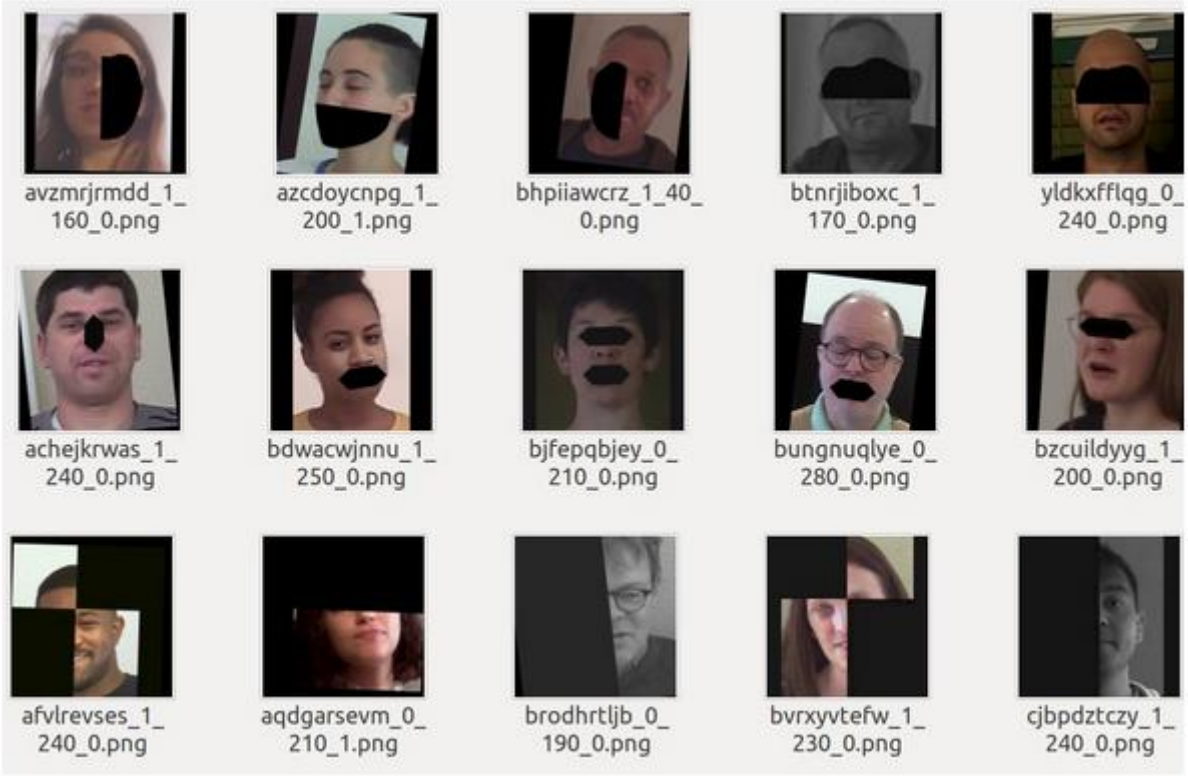

Рисунок 19. Приклад створення масок для навчання

Для тренування переможець конкурсу використовував сумарно 6 графічних прискорювачів Nvidia Titan V.

З цього можна зробити висновок, що не тільки архітектура та кількість даних можуть покращити якість роботи моделей глибокого навчання. Більший внесок можна отримати з розвитком нових архітектур, а також додаткових ідей, які використовують деяке розуміння роботи інших алгоритмів машинного навчання.

Висновки до розділу 3

В цьому розділі було розглянуто теоретичні рекомендації щодо створення алгоритмів визначення Deepfake. Було створено практичну реалізацію алгоритму визначення з наведених теоретичних рекомендації. Також було перевірено різні методи початкового алгоритму, протестовано методи покращення результату та проведене їх порівняння з існуючими моделями.

Визначено оптимальний алгоритм класифікації, створено приклад програмного коду, який можна використовувати як для роботи з зображеннями, так і для роботи з відеозаписами.

ВИСНОВКИ

В магістерській роботі було виконано поставлену мету – визначити способи виявлення дезінформації у форматі «deep fake». Було розроблено та перевірено алгоритм для визначення deepfake контенту на фотографіях та відеозаписах.

При аналізі сфер використання deepfake було визначено що подібні алгоритми використовуються в багатьох сферах життя і для багатьох цілей, як для законних так і для не законних. Також було визначено, що потреба у алгоритмах визначення deepfake наразі велика.

Під час перевірки алгоритмів було визначено, що найкращим алгоритмом визначення штучного інтелекту стає інший штучний інтелект. Під час аналізу робіт на цю тему було визначено, що якість визначення значно зростає при використанні додаткової обробки даних перед навчанням. Вибір правильної архітектури нейронної мережі також покращує результати роботи.

При розробці алгоритму визначення було перевірено різні моделі та методи обробки даних. Були враховані найбільш потрібні сфери застосування таких алгоритмів, надано рекомендації щодо розробки, впровадження та використання алгоритмів визначення.

Було розроблено приклад визначення застосування deepfake на фотографіях та відеозаписах на основі визначених в цій роботі методів та алгоритмів.

Отримані результати можуть бути використані в реальних процесах та програмах виявлення дезінформації, а також для машинного навчання алгоритмів розпізнавання графічної та відеоінформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. https://arxiv.org/pdf/2003.00196.pdf
2. https://arxiv.org/abs/1512.03385
3. https://arxiv.org/abs/1503.03832
4. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
5. https://arxiv.org/abs/1409.1556
6. https://core.ac.uk/display/388651102?source=2
7. https://pytorch.org/
8. https://www.nvidia.com
9. https://github.com/pytorch/pytorch
10. https://www.kaggle.com/c/deepfake-detection-challenge
11. https://www.kaggle.com/c/16880/datadownload/dfdc_train_all.zip
12. https://pytorch.org/vision/stable/models.html
13. https://arxiv.org/abs/1905.11946
14. https://pytorch.org/hub/
15. https://arxiv.org/abs/2103.00484
16. https://arxiv.org/abs/2105.00192
17. https://arxiv.org/abs/1701.08435
18. https://www.python.org/
19. https://paperswithcode.com/dataset/tai-chi-hd
20. http://www.stulyakov.com/papers/monkey-net.html
21. https://github.com/NVlabs/stylegan
22. https://arxiv.org/abs/1812.04948
23. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. CoRR, abs/1711.11585, 2017.
24. https://developer.nvidia.com/cuda-toolkit
25. https://arxiv.org/abs/1905.00641
26. https://github.com/YadiraF/PRNet
27. https://arxiv.org/abs/1812.04948
28. https://arxiv.org/abs/1909.08072
29. https://arxiv.org/abs/2109.06098

30. Alzantot, M., Sharma, Y., Chakraborty, S., and Srivastava, M. Genattack: Practical black-box attacks with gradient-free optimization. arXiv preprint arXiv:1805.11090, 2018
31. Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. The security of machine learning. Machine Learning, 81(2):121–148, 2010.
32. Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: Improving robustness to adversarial examples. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 854–863. JMLR. org, 2017.
33. https://github.com/opencv/opencv/tree/master/data/haarcascades
34. https://github.com/opencv/opencv
35. https://albumentations.ai/
36. https://arxiv.org/pdf/1909.12778.pdf
37. https://arxiv.org/abs/1803.01164
38. https://paperswithcode.com/method/zfnet
39. https://habr.com/ru/post/301084/
40. https://en.wikipedia.org/wiki/Residual_neural_network
41. https://arxiv.org/pdf/1406.2661.pdf
42. Bengio, Y., Yao, L., Alain, G., and Vincent, P. (2013b). Generalized denoising auto-encoders as generative models. In NIPS26. Nips Foundation.
43. Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In NIPS'2012.
44. rizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.
45. https://arxiv.org/abs/1312.6114
46. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
47. https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence
48. https://arxiv.org/abs/1906.02691
49. https://arxiv.org/pdf/1606.05908.pdf

50. https://arxiv.org/abs/2102.04075
51. https://arxiv.org/abs/1904.09658
52. https://arxiv.org/pdf/2106.07822.pdf
53. https://arxiv.org/pdf/1503.03832.pdf
54. http://dlib.net/
55. https://uk.wikipedia.org/wiki/C%2B%2B
56. https://colab.research.google.com/signup
57. https://www.nvidia.com/ru-ru/data-center/tesla-v100/
58. https://www.nvidia.com/ru-ru/data-center/tesla-p100/
59. https://arxiv.org/abs/1905.11946
60. https://www.imdb.com/title/tt0114709/
61. https://www.imdb.com/title/tt0126029/
62. https://play.google.com/store/apps/details?id=io.faceapp&hl=uk&gl=US