

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Застосунок для перенесення обличчя на відео
на основі нейронних мереж

Виконав: студент IV курсу, групи СНЗс-41
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Гудз Я.М.
(підпис) (прізвище та ініціали)

Керівник Матійчук Л.П.
(підпис) (прізвище та ініціали)

Нормоконтроль Шимчук Г.В.
(підпис) (прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Гудзу Ярославу Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи Застосунок для перенесення обличчя на відео
на основі нейронних мереж

Керівник роботи Матійчук Любомир Павлович, д.е.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» 05 2025 року № 4/7-449

2. Термін подання студентом завершеної роботи 18.06.2025 р.

3. Вихідні дані до роботи наукові літературні джерела

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

1. Огляд предметної області.

2. Основи переносу обличчя. Проектування застосунку

3. Реалізація та тестування застосунку

4. Безпека життєдіяльності, основи охорони праці

Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титулка. 2. Актуальність роботи 3. Мета, задачі дослідження.

4 Приклади перенесення обличчя за допомогою різних систем.

5. Етапи перенесення обличчя. 6. Схема перенесення обличчя на основі GAN.

7. Функціональні вимоги до застосунку. 8. Діаграми варіантів використання та розміщення.

9. Програмні засоби, які використані для розробки.

10. Скріншоти сторінок сайту. 11. Топологія нейромережі.

12. Результати роботи застосунку.

13. Висновки. Основні результати проведеного дослідження.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доц. каф. МТ		

7. Дата видачі завдання 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	07.05 – 08.05.25	Виконано
2.	Підбір джерел про перенесення зображення обличчя на відео	09.05 – 12.05.25	Виконано
3.	Опрацювання джерел про системи перенесення обличчя на відео	13.05 – 16.05.25	Виконано
4.	Виконання дослідження щодо розробки програмного забезпечення для перенесення обличчя на відео	17.05 – 19.05.25	Виконано
5.	Розроблення програмного коду	20.05 – 21.05.25	Виконано
6.	Оформлення розділу «Огляд предметної області»	22.05 – 23.05.25	Виконано
7.	Оформлення розділу «Основи переносу обличчя. Проектування застосунку»	25.05 – 27.05.25	Виконано
8.	Оформлення розділу «Реалізація та тестування застосунку.»	28.05 – 30.05.25	Виконано
9.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	31.05 – 03.06.25	Виконано
10.	Оформлення кваліфікаційної роботи	20.05 – 03.06.25	Виконано
11.	Нормоконтроль	01.06 – 05.06.25	Виконано
12.	Перевірка на плагіат	04.06 – 07.06.25	Виконано
13.	Попередній захист кваліфікаційної роботи	07.06 – 09.06.25	Виконано
14.	Захист кваліфікаційної роботи	19.06.25	

Студент

(підпис)

Гудз Я.М.

(прізвище та ініціали)

Керівник роботи

(підпис)

Матійчук Л.П.

(прізвище та ініціали)

АНОТАЦІЯ

Застосунок для перенесення обличчя на відео на основі нейронних мереж // Гудз Ярослав Михайлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2025 // С. – 59, рис. – 27, табл. – 5, слайдів – 13, бібліогр. – 61.

Ключові слова: python, детектування обличчя, нейронна мережа, машинне навчання, обробка зображень

У першому розділі описується предметна область. Також проводиться аналіз існуючих аналогів та методів, що вирішують поставлені завдання.

У другому розділі були розглянуті етапи перенесення обличчя, методи та алгоритми детектування, вилучення інформації про ідентичність. Було розглянуто архітектуру нейронної мережі для перенесення обличчя. Розглянуті основні модифікації зображення, необхідні для перенесення обличчя. Також визначені функціональні та нефункціональні вимоги для системи, представлені діаграми варіантів використання та компонентів. Міститься опис проектування всіх складових системи. В результаті проектування архітектури вся система була розбита на кілька частин, які будуть представляти окремі Docker-контейнери.

У третьому розділі були розглянуті програмні засоби для реалізації, розглянутий процес формування набору даних для навчання, наведено опис процесу навчання нейронної мережі, а також опис реалізації програми та інтерфейсу користувача. розроблено клас, що дозволяє оцінювати якість перенесення осіб та дві функції для підготовки набору даних. Всі компоненти розроблюваної програми були упаковані в Docker -контейнери. Описано процес тестування моделі нейронної мережі за допомогою метрик оцінки якості перенесення осіб. Було проведено функціональне тестування застосунку та тестування інтерфейсу користувача на адаптивність та на коректне відображення в різних браузерах.

ANNOTATION

An application for face transfer to video based on neural networks // Hudz Yaroslav// Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science // Ternopil, 2025 // P. - 58, Fig. - 27, Table - 5, Slide - 13, References - 61.

Keywords: python, face detection, neural network, machine learning, image processing

The first chapter describes the subject area. An analysis of existing analogues and methods that solve the tasks is also carried out.

The second chapter considered the stages of face transfer, detection methods and algorithms, and identity information extraction. The architecture of a neural network for face transfer was considered. The main image modifications necessary for face transfer were considered. Functional and non-functional requirements for the system were also determined, diagrams of use cases and components were presented. The design of all system components is described. As a result of the design of the architecture, the entire system was divided into several parts, which will be represented by separate Docker containers.

The third chapter considered software tools for implementation, considered the process of forming a data set for training, provided a description of the neural network training process, as well as a description of the implementation of the program and the user interface. A class was developed that allows assessing the quality of face transfer and two functions for preparing a data set. All components of the developed program were packaged in Docker containers. The process of testing the neural network model using person transfer quality assessment metrics is described. Functional testing of the application and testing of the user interface for adaptability and correct display in different browsers were conducted.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

CNN (англ. Convolutional Neural Network) – згорткова нейронна мережа.

GAN (англ. Generative Adversarial Network) – генеративно- складна нейронна мережа.

HOG (histogram of oriented gradients) – гістограма напрямлених градієнтів.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Огляд алгоритмів заміни обличь	10
1.1.1 Replacement-based	10
1.1.2 Model-based.....	11
1.1.3 Learning-based.....	11
1.2 Програмні засоби для перенесення обличь	13
РОЗДІЛ 2. ОСНОВИ ПЕРЕНОСУ ОБЛИЧЬ. ПРОЕКТУВАННЯ ЗАСТОСУНКУ	15
2.1 Етапи перенесення обличь	15
2.2 Алгоритми Face Detection.....	16
2.3 Алгоритми Identity Extraction	16
2.4 Алгоритм Face Swap	19
2.5 Обробка зображень	20
2.6 Основні відомості про нейронні мережі.....	22
2.7 Проектування застосунку	24
2.7.1 Визначення вимог до застосунку	24
2.7.2 Варіанти використання застосунку	25
2.7.3 Проектування архітектури системи	26
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ	31
3.1 Програмні засоби реалізації.....	31
3.2 Реалізація Web-interface.....	31
3.3 Реалізація Web-server.....	34
3.4 Реалізація Task executor.....	34
3.5 Реалізація Serving server	37
3.6 Підготовка даних та передобробка набору даних	38
3.7 Вибір та навчання нейронної мережі	40
3.8 Реалізація методів оцінки якості перенесення обличь.....	43
3.9 Контейнеризація системи.....	43
3.10 Тестування застосунку	44

3.10.1 Тестування веб-сайту.....	44
3.10.2 Функціональне тестування.....	44
3.10.2 Тестування нейронної мережі.....	44
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	48
4.1 Навчання працюючих і інструктажі з охорони праці.....	48
4.2 Санітарно-гігієнічні вимоги до умов праці	50
ВИСНОВКИ.....	53
ПЕРЕЛІК ДЖЕРЕЛ	54
ДОДАТКИ	

ВСТУП

Актуальність теми. В даний час на ринку користуються попитом технології з перенесення обличь на відео [9,29,41,50]. Дана технологія використовується для оптимізації витрат на виробництво відеоконтенту, для відмови використання у фільмах дублерів, для локалізації реклами, створення корпоративних амбасадорів, аватарів, лідерів думок з особою, виняткові права на яку завжди залишаються за компанією, для розваги, анонімності в соціальних мережах та відео, приватні користувачі.

Досвідчений дизайнер за допомогою професійного графічного редактора може здійснювати фотореалістичну заміну однієї особи на фотографії на іншу, якщо особи знаходяться в однаковому ракурсі. Але навіть професіонал зазнає труднощів, коли ракурси обличь на двох фото відрізняються. Перенесення особи з фотографії на відео може теоретично здійснюватися покадрово, але практично не здійснимо для людини. Завдання переносу обличь із фото на фото, а також з фото на відео може ефективно вирішувати нейронна мережа. Для нейронної мережі поворот обличчя в потрібний ракурс (у допустимих межах) і покадрове перенесення - це завдання, що легко вирішуються.

Технології заміни обличь існують як веб-сервіси та застосунки. В рамках даної розробки вирішується завдання фотореалістичного переносу обличь на фото та відеоконтент для професійного/напівпрофесійного /приватного застосування. Звідси можна визначити властивості даної розробки, які одночасно є конкурентними перевагами (для цільової аудиторії) порівняно з іншими продуктами ринку:

- висока якість перенесення обличь та створення контенту;
- можливість обробки власного вхідного відео;
- наявність дружнього інтерфейсу користувача;
- онлайн-доступ до застосунку;
- висока швидкість його роботи.

Таким чином, актуальною стає завдання створення застосунку із

перенесення обличь.

Мета роботи – розробка програми для перенесення обличь на відео на основі нейронних мереж.

Для досягнення мети виділено ряд завдань:

- провести огляд систем-аналогів;
- провести огляд методів перенесення обличь;
- виконати проектування застосунку;
- виконати реалізацію застосунку;
- повести тестування розробки.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд алгоритмів заміни обличчя

Існуючі алгоритми заміни обличчя можна умовно розділити на три категорії: replacement-based, model-based and learning-based [6].

1.1.1 Replacement-based

Таки метод, зазвичай замінює, область обличчя в еталонному зображенні вхідною областю обличчя. У роботах [4, 3] автори побудували набір зображень обличчя з різними виразами, формами обличчя та позами, і алгоритм міг автоматично вибирати найбільш схоже еталонне обличчя з набору зображень для заміни обличчя (рис. 1.1).

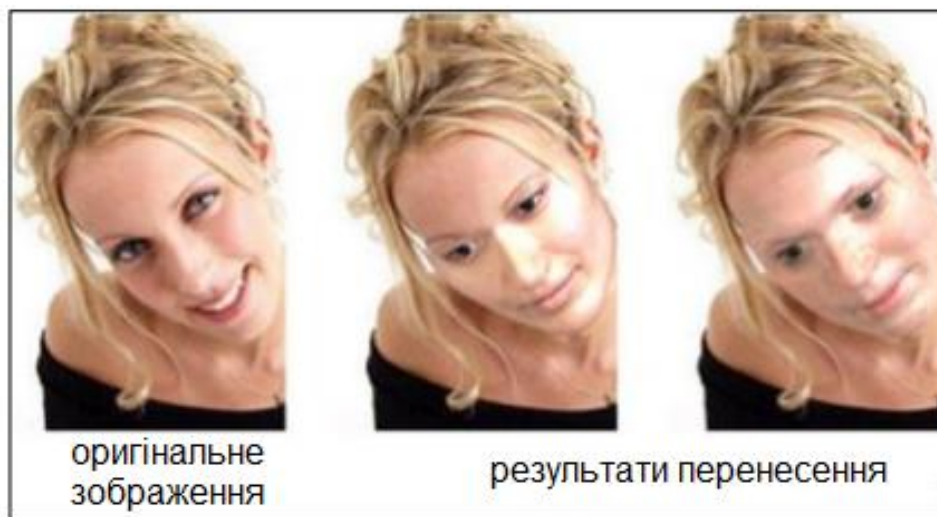


Рисунок 1.1 – Приклад перенесення обличчя методом, описаним у роботі [4]

Використання такого методу обмежене, оскільки користувачі не можуть самостійно обрати своє еталонне зображення обличчя.

У роботах [31, 36] система дозволяє вибирати власне еталонне фото. FSGAN [36] запропонувала двоетапну архітектуру, яка спочатку виконує передачу виразу та пози за допомогою мережі реконструкції обличчя, а потім використовує іншу мережу малювання обличчя для змішування вихідної особи з

цільовим зображенням.

Загальна проблема цієї групи методів у тому, що вони чутливі до вхідного вихідного зображення. Перебільшений вираз чи неприродне становище вихідного обличчя сильно вплинуть на результат заміни обличчя.

1.1.2 Model-based

За такого підходу [37] створюється двовимірна або тривимірна параметрична модель ознак для подання людського обличчя, а параметри і ознаки добре адаптуються до вхідного зображення (рис.1.2) . Потім виконується реконструкція обличчя за еталонним зображенням на основі результату налаштування параметрів моделі.

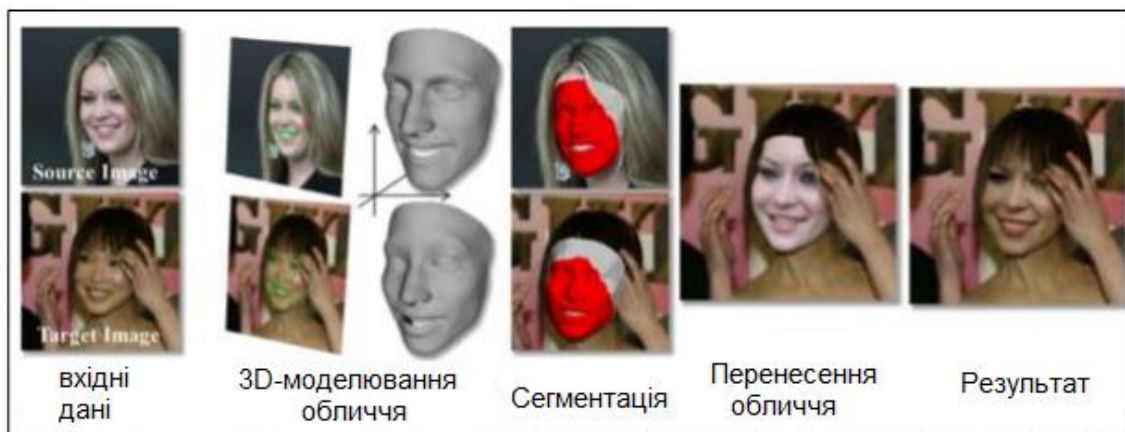


Рисунок 1.2 – Приклад перенесення обличь методом, описаним у роботі [37]

Алгоритм вимагає вхідної 3D-моделі, ручного налаштування для отримання кращого результату та збору даних.

1.1.3 Learning-based

Останнім часом методи learning-based, засновані на нейронних мережах, досягли найкращих результатів.

DeerFaceLab [41] - відоме рішення, засноване на реалізації автоенкодерів. Основна проблема для цього підходу полягає в тому, що модель повинна навчатися на вихідному та цільовому зображеннях обличчя кожного разу, коли

хтось хоче змінити обличчя на відео. При цьому потрібний збір величезної кількості даних. Для якісного перенесення обличь потрібно в середньому 4-5 днів навчання (рис. 1.3).

Такий підхід зручний для багаторазового перенесення обличчя. Це може знадобитися у великих проєктах, наприклад, перенесення обличь у фільмах або серіалах. Однак DeepFaceLab не підходить для багатьох завдань в реальному часі, коли ми не маємо величезних наборів даних для навчання [9] .

У роботі [50] запропоновано підхід, здатний обробляти зображення без перенавчання. Ідея полягає не у створенні нового зображення з нуля, а в перекладі одного кадру в інший за допомогою карт сегментів особи та афінних перетворень, що навчаються.

Обличчя особи в цьому підході розбивається на сегменти, які можна перенести. Даний метод добре справляється з перенесенням окремих частин обличчя – кольору очей, губ, волосся.

Однак сумарне перенесення частин обличчя візуально далеке від досконалості, оскільки сегменти можуть зрушуватися при поворотах (див. рис. 1.3).

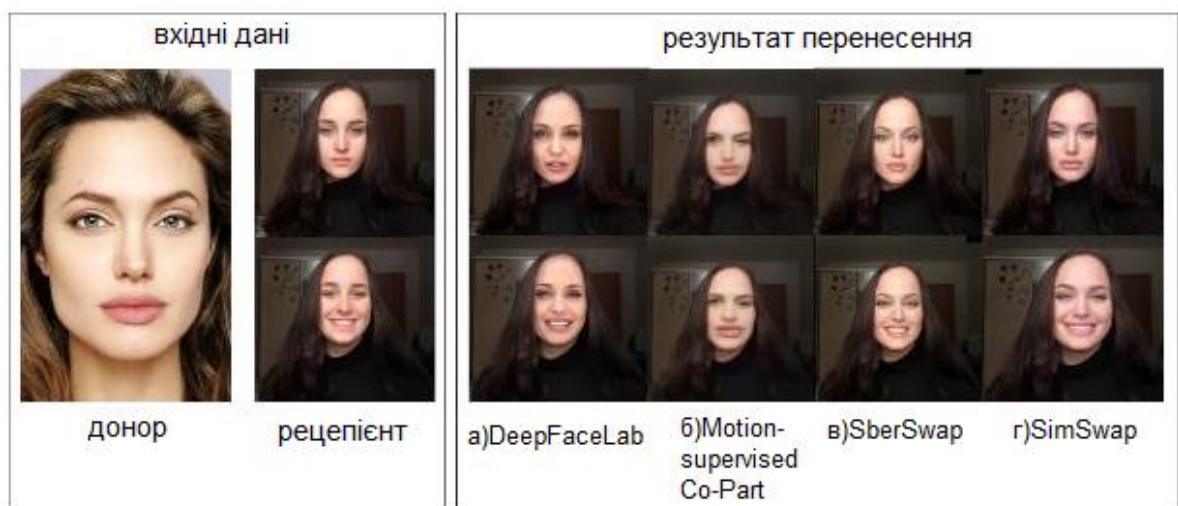


Рисунок 1.3 – Приклади перенесення обличь за допомогою різних систем

У роботі [30] метод FaceSyfter дозволив отримати результати заміни обличчя з високою якістю. У FaceSyfter використовується складна двоетапна

архітектура, що дозволяє досягти високої якості перенесення рис обличчя. Однак, FaceShifter занадто слабо зберігає атрибути реципієнта, тому в результаті часто виходить невідповідність виразів обличчя. Подібний метод використовується в роботі [9] (див. рис. 1.3).

У роботі [7] наведена система ідеологічно схожа на FaceSYfter [30] , відмінність полягає у використанні загальної архітектури моделі нейронної мережі замість двох різних моделей. Ця система найбільш добре зберігає атрибути реципієнта.

На поточному етапі базова версія нейронної мережі реалізує перенесення обличчя із роздільною здатністю не більше 256x256 пікселів, оскільки ця мережа навчалася на наборі даних обличчя розміром 256x256 пікселів, зібраних з кадрів фільмів. При більшому розмірі області обличчя на відео, здійснювався перенос маленького обличчя з подальшим масштабуванням до потрібного розміру . Це не завжди прийнятно, тому збільшення якості обличчя, котрі переносяться, є важливим завданням.

1.2 Програмні засоби для перенесення обличчя

Розглянемо спеціалізовані застосунки, які здійснюють перенесення обличчя – Snapchat [52] , Face App [15] , Faceswap Live [16] . Незважаючи на те, що такі застосунки, на відміну від розглянутих раніше систем, мають дружній інтерфейс, вони мають розважальні цілі і не мають на меті створення реалістичного контенту високої якості.

До лідерів цього напрямку відноситься Reface [49]. Даний проект вже кілька років впевнено тримається на ринку і позиціонується, як Funny face swap videos. Якість перенесення обличчя та максимальна роздільна здатність (для відео до 640x360 пікселів) не дозволяють говорити про її застосування інакше, як для приватного використання в розважальних цілях.

Серед сервісів, що позицінуються, як орієнтовані на гарну якість для професійного / напівпрофесійного / особистого використання можна навести приклад такі, як BestSwap [1] і Face Swapper [53] . Серед ключових особливостей

даних сервісів варто виділити те, що вони не пропонують перенесення обличь на відео, тільки на фотографії. Ці послуги є платними.

Серед сервісів, що здійснюють перенесення обличь на відео як придатну для комерційних цілей, можна виділити DeepCake. Даний сервіс використовувався для заміни обличь у рекламах банків та мобільних операторів. Але за рахунок того, що проект використовує DeepFaceLab, як ключову технологію, процес пересадки обличчя виявляється вкрай тривалим і дорогим для замовника.

Із проведеного аналізу видно, що існуючі рішення не вирішують одночасно такі проблеми: довгий час перенесення, низька якість вихідного відео, відсутність можливості перенесення особи на довільне відео з довільного фото, а також відсутність зручного інтерфейсу. Таким чином додаток, що розробляється, повинен охоплювати всі наявні у аналогів гідності і усувати наявні недоліки.

РОЗДІЛ 2. ОСНОВИ ПЕРЕНОСУ ОБЛИЧЬ. ПРОЕКТУВАННЯ ЗАСТОСУНКУ

2.1 Етапи перенесення обличь

Для перенесення обличь на вхід системі надходить фотографія та відеофайл. Обличчя людини на фотографії називається донором і є обличчям, яке переноситься на відеозображення. Відеофайл містить Обличчя людини (реципієнта), яка замінюється обличчям донора.

Рис. 2.1 ілюструє етапи перенесення обличь. Фотографія донора проходить етап виявлення обличчя (детектування, Face Detection). Мета цього етапу – визначити координати обличчя на вхідному зображенні, а також координати ключових точок обличчя (куточки рота, очей та кінчик носа). Отримані координати використовуються на етапі отримання ідентичності (Identity Extraction). На цьому етапі обличчя представляється як набір векторів ознак, що описує характерні риси обличчя. Інформація про характерні особливості обличчя називається інформацією про ідентичність.

Відеофайл розбивається на кадри. Потім кожен кадр проходить етап Face Detection, аналогічно фотографії з донором.

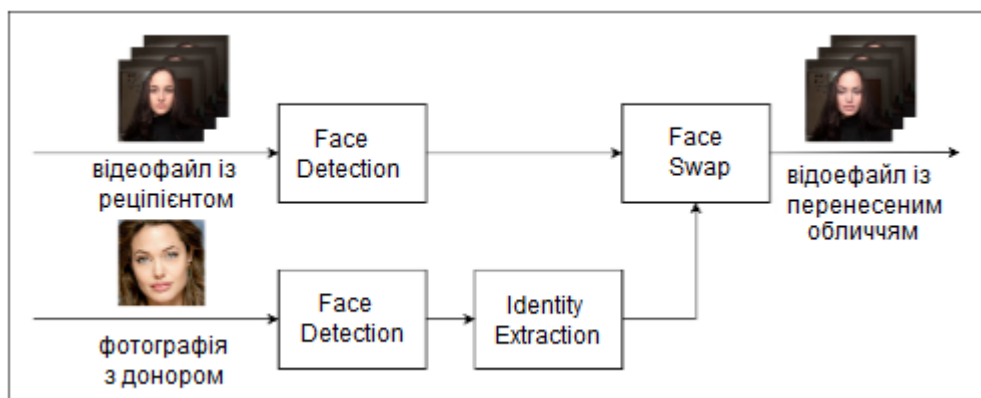


Рисунок 2.1 – Етапи перенесення обличчя

Заключним є етап перенесення обличчя (Face Swap). На цьому етапі інформація про ідентичність обличчя реципієнта замінюється на інформацію

ідентичності донора, зберігаючи при цьому незмінною інформацію, яка описує форму голови, її поворот, нахил, міміку (атрибути обличчя). Результатом етапу перенесення обличчя є кадр із обличчям донора.

2.2 Алгоритми Face Detection

Завдання детектування обличчя є першим етапом у процесі розв'язання задачі розпізнавання особи. Методів вирішення цього завдання дуже багато [29]/

Складність детектування обличчя людей викликана кількома моментами :

- робота системи з особами під різними кутами;
- розпізнавання людей різної зовнішності (стаття, статура, етнічна приналежність тощо);
- робота з різними виразами обличчя та позами голови;
- розпізнавання людей із аксесуарами (головні убори, окуляри, шарфи);
- розпізнавання людей, що мають рослинність на обличчі або сильний макіяж.

В основі багатьох сучасних алгоритмів детектування об'єктів лежить ідея, запропоновані Віолою та Джонсоном на початку 2000 -х років. Дана схема побудови детектора дозволяє суттєво скоротити його обчислювальну складність із збереженням узагальнюючої здатності.

Метод застосовує ковзне вікно, коли рамка, розмір якої, менший, аніж зображення вихідне, переміщається із певним кроком за зображенням, та при допомозі набору слабких класифікаторів встановлює, чи є у вікні обличчя.

Каскадні структури є стандартом у побудові детекторів. Але найпростіших ознак, здобутих функціями Хаара недостатньо для високого рівня виявлення обличчя. Каскадні класифікатор Віоли Джонса і подібні до нього мають пряму залежність часу обробки зображення від його змісту. Крім того, такі методи нестійкі до великого кута повороту голови та різних рівнів освітленості.

Виділяють ще один великий клас, до нього відносяться методи, в яких пошук здійснюється за допомогою порівняння кожної ділянки зображення із заданим шаблоном. Такі алгоритми досить точні, але не підходять для задач

реального часу, оскільки мають дуже низьку швидкість .

Найбільш надійними системами на сьогоднішній день є системи, побудовані на базі CNN. В останні роки CNN демонструють високі результати у вирішенні завдань розпізнавання образів. CNN здатна детектувати особи у широкому діапазоні положення голови, освітлення. Але має високу обчислювальну складність. CNN мають високу гнучкість і дозволяють заздалегідь задавати складність моделей, змінюючи кількість шарів карт і розміри згорткових ядер. Таке тонке налаштування дає змогу отримати високі показники точності при пошуку об'єктів на сильно неоднорідному фоні.

CNN здатні виділяти різні особливості зображення, тим самим переходячи від зображення до абстрактних деталей .

2.3 Алгоритми Identity Extraction

У роботі [28] виділено три групи методів розв'язання задачі вилучення інформації про ідентичність: локальні підходи (local approaches), цілісні підходи (holistic approaches) та гібридні підходи (hybrid approaches) (рис. 2.2) .

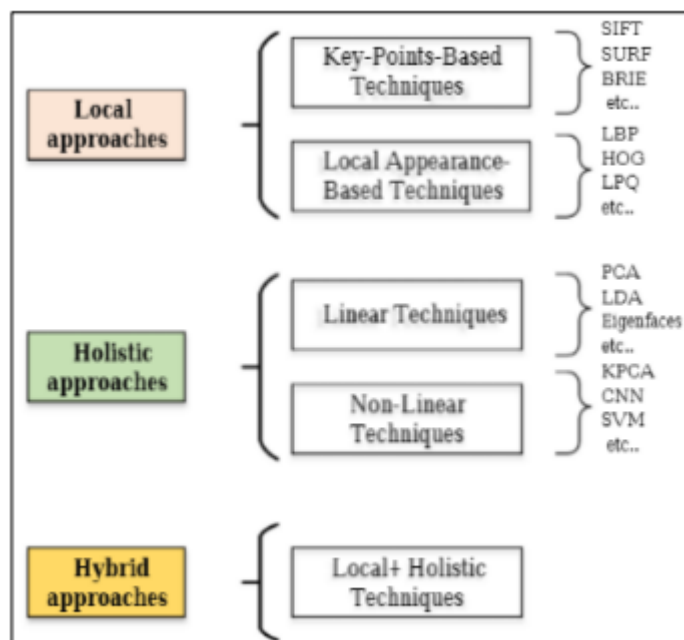


Рисунок 2.2 – Класифікація методів вилучення інформації про ідентичність

Локальний підхід класифікується за певними рисами особи, не враховуючи всієї особи. Основна мета локальних підходів - виявити відмінні риси обличчя [28] . Всі методи націлені на вилучення окремих дескрипторів ділянок обличчя (ніс, губи, лоб, очі нижня частина підборіддя) або точок окремих точок.

Найбільш популярним представником цього підходу є алгоритм HOG. Основна ідея алгоритму у тому, що зображення може бути описано розподілом градієнтів інтенсивності чи напрями країв. Переважно формування цих дескрипторів проходить завдяки розбиттю зображення на осередки, і надання кожному такому осередку HOG для пікселів, котрі містяться і осередку, а вже їх комбінація власне і є дескриптором. Для більшої продуктивності проводиться спочатку локалізація окремих ділянок обличчя, а потім для цих ділянок знаходяться дескриптори. Такий підхід більш надійний, але все ще мають неминуче обмеження стійкості до міміки, оклюзій та пози голови.

Цілісний підхід використовує все обличчя як вхідні дані, а потім проектує його в невеликий підпростір або в площину кореляції. Основна мета цілісних підходів - обробка всього обличчя та представлення зображення обличчя матрицею пікселів [28] .

Одним із яскравих представників такого підходу є метод PCA. Його суть полягає у поданні зображення обличчя вектором малої розмірності (головних компонентів).

Такі методи є відносно швидкими та простими. Але вони не вирішують неконтрольовані зміни особи, що відхиляються від попередніх припущень.

Гібридні підходи засновані на локальних та підпросторових функціях, щоб використовувати переваги як підпросторових, так і локальних методів, які можуть запропонувати кращу продуктивність .

Всі ці методи можна назвати «неглибокими». Такі традиційні методи покращувалися і доповнювалися з часом різними алгоритмами. Але більшість методів до цього дня націлені лише на один аспект необмежених змін обличчя, такий як освітлення, поза, вираження або маскування.

Більш сучасним підходом зараз є використання методів глибинного

навчання, зокрема CNN. Такі методи використовують каскад з кількох рівнів блоків обробки для вилучення та перетворення ознак. Вони вивчають кілька рівнів уявлень, котрі відповідають за різні рівні абстракції. Формується ієрархія абстракцій, що демонструє сильну незмінність пози обличчя, висвітлення та змін виразу (рис. 2.3) [58] :

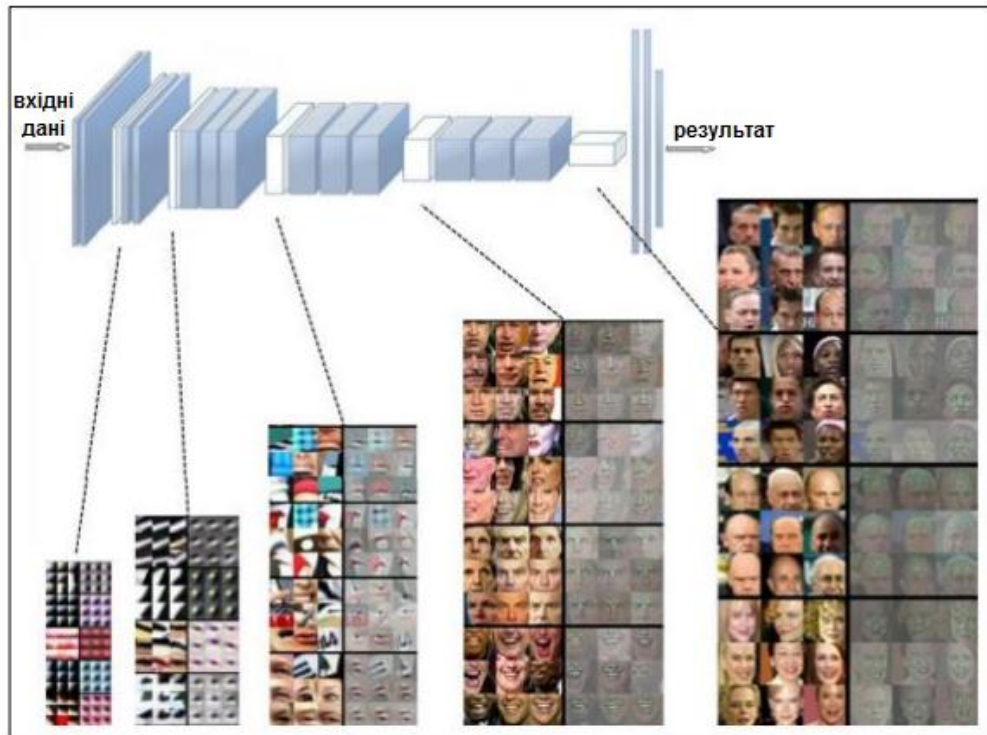


Рисунок 2.3 – Рівні абстракції

З рисунка видно, що зі зростанням рівня нейронної мережі, він вивчає дедалі складніші особливості текстури. Причому кожен шар може використовувати звичні нами «неглибокі» функції, які перевірені роками [59]. Наприкінці ми отримуємо поєднання деяких абстракцій вищого рівня, яке представляє обличчя особистості з безпрецедентною стабільністю .

Безумовно, нейронні мережі так само, як і інші підходи мають недоліки. До головних недоліків можна віднести вимогу до обчислювальної системи. З іншого боку, навчання моделей потрібно збирати величезні набори даних. Більш докладно з недоліками застосування нейронних мереж можна ознайомитись у [55].

2.4 Алгоритм Face Swap

Для основної частини перенесення обличь буде використана GAN. Мережі такого типу складаються з двох різних моделей, генератора (Generator) та дискримінатора (Discriminator). Робота генератора полягає в тому, щоб породжувати підроблені зображення, які виглядають як навчальні зображення. Робота дискримінатора полягає в тому, щоб подивитися на зображення і вивести, чи це реальне навчальне зображення або підроблене зображення з генератора.

Під час навчання генератор постійно намагається перехитрити дискримінатор, генеруючи все кращі та кращі підробки, тоді як дискримінатор працює над тим, щоб стати кращим і правильно класифікувати реальні та підроблені зображення (рис. 2.4). Рівновага цієї гри полягає в тому, що генератор генерує ідеальні підробки, які виглядають так, ніби вони прийшли безпосередньо з навчальних даних, а дискримінатор залишається завжди вгадувати з 50% впевненістю, що вихід генератора є реальним або підробленим [19] .

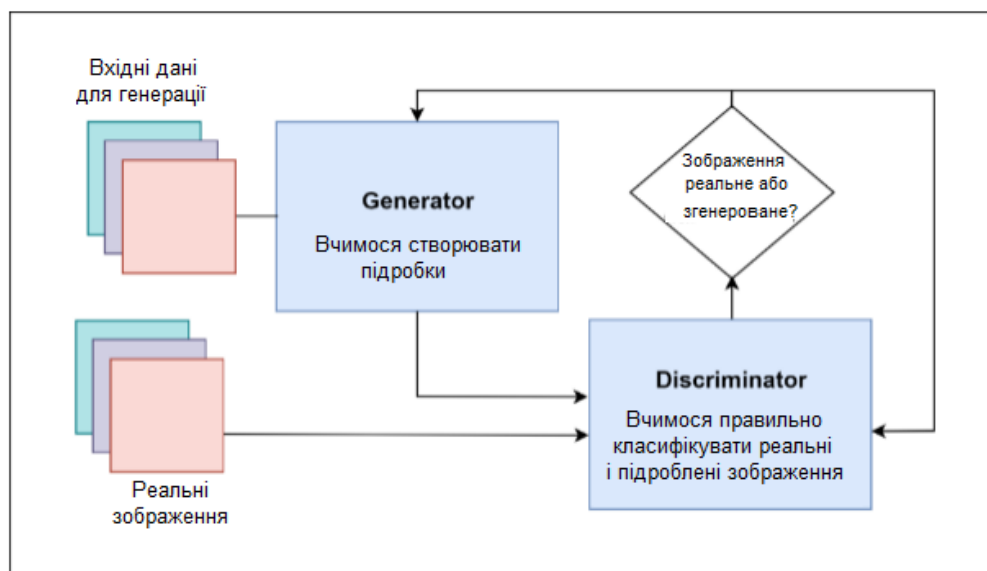


Рисунок 2.4 – Схема GAN

Завдання перенесення обличь полягає в тому, щоб поміняти місцями обличчя реципієнта з обличчям донора. Потрібно замінити інформацію про ідентичність обличчя реципієнта на інформацію ідентичності обличчя донора,

зберігаючи при цьому атрибути обличчя реципієнта незмінними.

Інформація про ідентичність та атрибути тісно пов'язана, і їх важко відрізнити один від одного. Тому при навчанні нейронна мережа вчиться зберігати атрибути цільового обличчя, при цьому не надто модифікуючи ідентичність. Для цього на етапі дискримінатора застосовується функція помилки.

Генератор є основною частиною та дозволяє міняти місцями обличчя. Він містить такі частини: Encoder, ID Injection Module (IIM), Decoder (рис. 2.5).

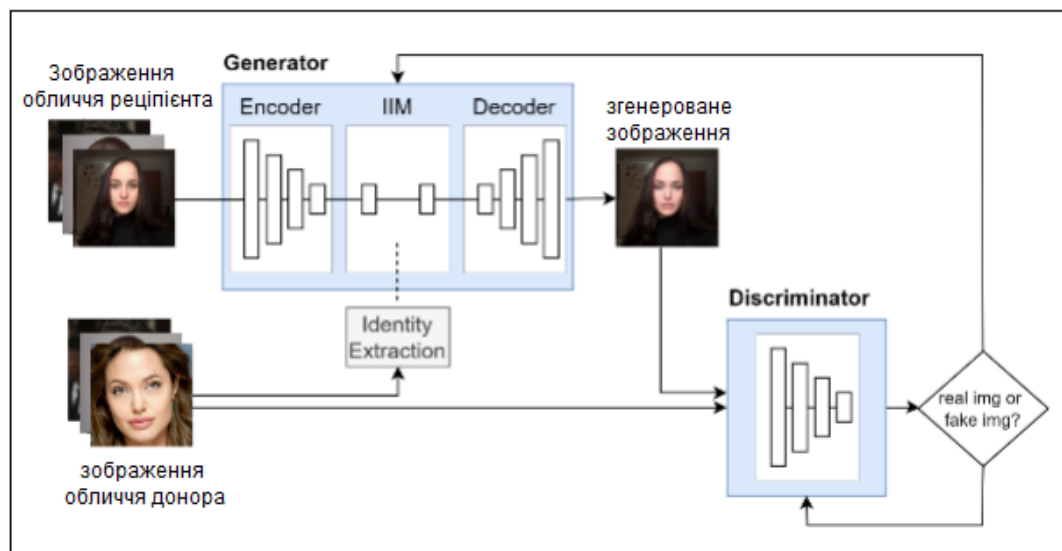


Рисунок 2.5 – Схема перенесення обличчя

Encoder має риси обличчя з цільового зображення.

IIM передає ідентифікаційну інформацію із зображення донора зображенню реципієнта.

Decoder відновлює змінені ознаки у результуючому зображенні.

2.5 Обробка зображень

Після перенесення зображення з обличчям вставляється в початкове місце на зображенні. Однак, якщо вставляти картинку, отриману моделлю, то буде чітко помітна межа накладання на початковий кадр. Цей ефект виникає через неповну відповідність яскравості картинок і можливу розмитість картинки,

згенерованої моделлю. Крім того, при навчанні нейронна мережа крім відновлення обличчя може навчитися відновлювати шум фоновому зображенню. Таким чином, необхідно забезпечити плавний перехід від початкового кадру до отриманого переносу, а також унеможливити перенесення фону навколо обличчя.

Для забезпечення перенесення області обличчя використовуватимуться сегментаційні маски. Сегментація - це розподіл зображень на кілька сегментів або об'єктів [34] (рис. 2.6, а). Додатково маска розмивається біля кордонів, щоб забезпечити плавний перехід.

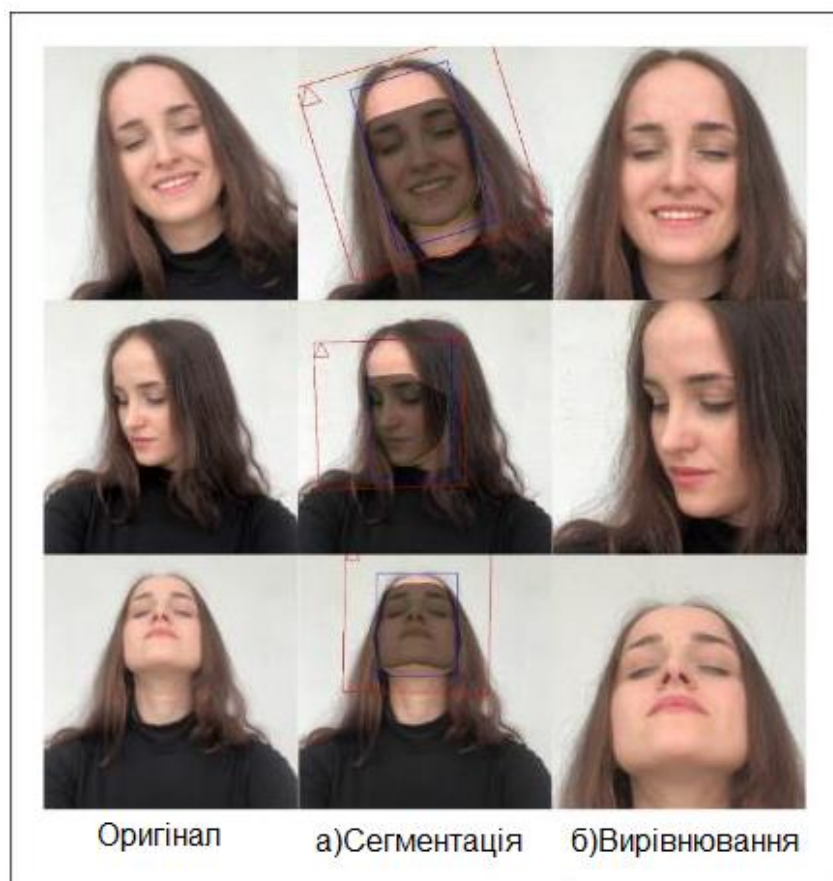


Рисунок 2.6 – Перетворення зображення

Зображення на етап Face Swar наводяться до єдиного розміру. Також здійснюється вирівнювання - зображення повертається так, щоб обличчя знаходилася вертикально (рис. 2.6, б). Після переносу обличчя проходить зворотний процес.

2.6 Основні відомості про нейронні мережі

Нейронна мережа є сукупністю нейронів, котрі визначеним способом поєднані один з одним. Нейрон є тим компонентом, котрий обраховує вихідний сигнал із комплексу вхідних власне.

З'єднання нейронів встановлюється структурою визначеної мережі. Кожен зв'язок властиво у мережі нейронів здатен повністю бути охарактеризованим із застосуванням трьох факторів, а саме: елемент, зв'язок від котрого виходить; елемент, зв'язок до котрого напрямлений; вага зв'язку.

Остання визначає, чи сигнал, котрий який проходить за цим зв'язком, буде посилено або послаблено, Кожен елемент мережі підпадає під ятке правило, згідно з яким обчислення його вихідного значення. Таке правило носить назву функцією активації (рис. 2.7).

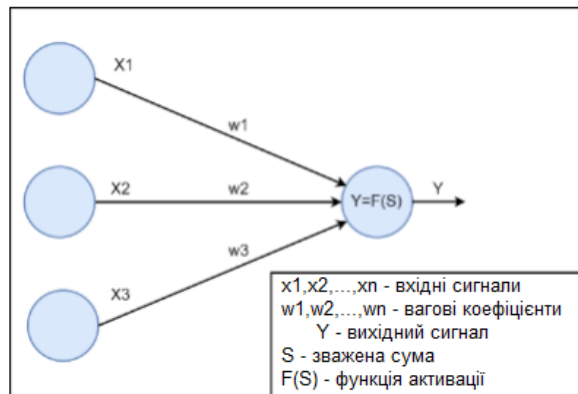


Рисунок 2.7 – Зв'язок між нейронами

Рис. 2.8 відображає функції активації, що найчастіше використовуються .

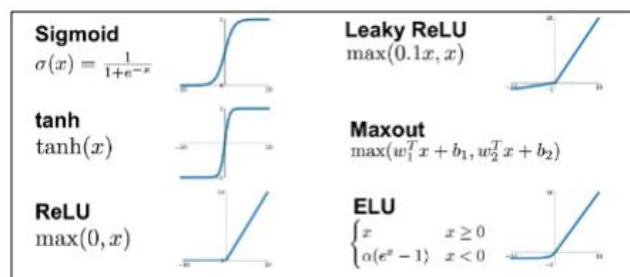


Рисунок 2.8 – Основні функції активації та їх графіки

Проведення навчання нейромережі є відшукуванням самого кращого набору ваг з метою максимізації точності прогнозування. Для такого набору пошуку використовуються оптимізатори. Найбільш поширеними є: стохастичний градієнтний спуск (SGD), середньоквадратичне поширення (RMSProp) та адаптивна оцінка моменту (Adam).

У багатошарових мережах нейрони власне поєднуються у шари. Кожен такий шар має набір нейронів, які мають єдиний вхідний сигнал. Кількість нейронів у шарі властиво може бути будь-якою і жодним чином не перебуває у залежності від числа нейронів у інших шарах. CNN є однією з форм багатошарових нейронних мереж.

Далі будуть розглянуті основні типи шарів, що використовуються в будові нейромереж.

Згортковий шар застосовується для виділення ознак із зображення. Квадрат, названий «ядром», рухається по зображенню зліва направо, переглядаючи його і проводячи скалярне перемноження значень комірок в ядрі та зображенні. Цей процес називається «згорткою». Нижче показано приклад цієї операції (рис. 2.9) .

Кожне ядро послідовно застосовується до всього зображення, складаючи карту ознак. Одна картка для одного ядра. Кількість ядер та розмір задається довільно. Значення в комірках ядра задається випадково самою нейромережею.

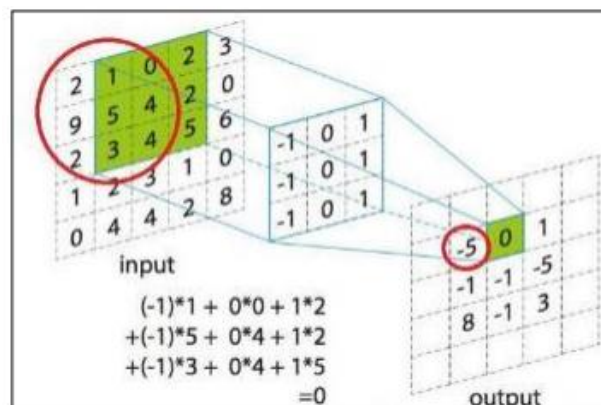


Рисунок 2.9 – Приклад обробки вхідного шару ядром згортки

Шар підвибірки (pooling layer) зменшує розмір зображення, комбінуючи

групу пікселів в один елемент. Для визначення вихідного значення для кожної карти ознак застосовується ковзне квадратне вікно довільного розміру, що перетворює всі пікселі в ньому в один. Як правило, використовується розмір 2x2, що дозволяє вдвічі зменшити зображення.

Значення шарів вибірки може обчислюватися як максимальне або середнє значення. Приклад роботи шару підвибірки з урахуванням типу об'єднання наведено на рис. 2.10.



Рисунок 2.10 – Приклад розрахунку вихідних значень з урахуванням виду об'єднання

Нормалізація (batch-normalization) вхідного шару мережі, як правило, здійснюється через масштабування даних, котрі поступають у функції активації. Для прикладу, якщо наявні ознаки зі значеннями у проміжку від 0 до 1 та якісь ознаки зі значеннями в діапазоні від 1 до 1000, тоді їх треба якимось чином нормалізувати, для пришвидшення навчання.

Також є множина інших шарів, котрі оптимізують навчання нейромережі: шари відображення вхідного кордону (ReflectionPad); шари, котрі підвищують роздільну здатність (Upsample) і т.д.

2.7 Проектування застосунку

2.7.1 Визначення вимог до застосунку

У ході проектування веб-застосунку було визначено функціональні та

нефункціональні вимоги до засобу, що розробляється.

Функціональні вимоги визначають функціональність ПЗ, тобто описують, яку поведінку повинна надавати розроблювана система.

Застосунок, що розробляється, повинен задовольняти такі функціональні вимоги:

- надавати можливість завантаження відеофайлу, розміру не більше 50МБ, у наступних форматах: mp4, mov, webm, gif, avi, m4p, m4v, mpg;
- надавати можливість завантаження зображення, розміру не більше 5МБ, у наступних форматах: jpg, jpeg, png, bmp;
- забезпечувати можливість запуску процесу перенесення обличь для відповідних фото та відео;
- надавати можливість вказувати електронну пошту;
- автоматично надсилати результат на електронну пошту.

До нефункціональних вимог системи відносяться властивості, які вона повинна мати. Наприклад, зручність використання, безпека, розширюваність.

Застосунок повинен задовольняти наступним нефункціональним вимогам:

- написаний мовою Python 3 [45] ;
- використовувати бібліотеку Pytorch [46] для роботи з нейронною мережею;
- реалізований за допомогою технологій HTML [23], CSS [10], та JavaScript [14] ;
- використовувати нейронні мережі для перенесення обличь на відео;
- забезпечувати запуск нейронних мереж на окремому сервері;
- запакований в Docker -контейнери [12] ;
- забезпечувати перенесення обличчя з найкращою якістю в порівнянні з базовою нейронною мережею, при цьому якість необхідно оцінювати за допомогою стандартних метрик.

2.7.2 Варіанти використання застосунку

При проектуванні була використана мова UML. Складено діаграму варіантів використання (рис. 2.11)

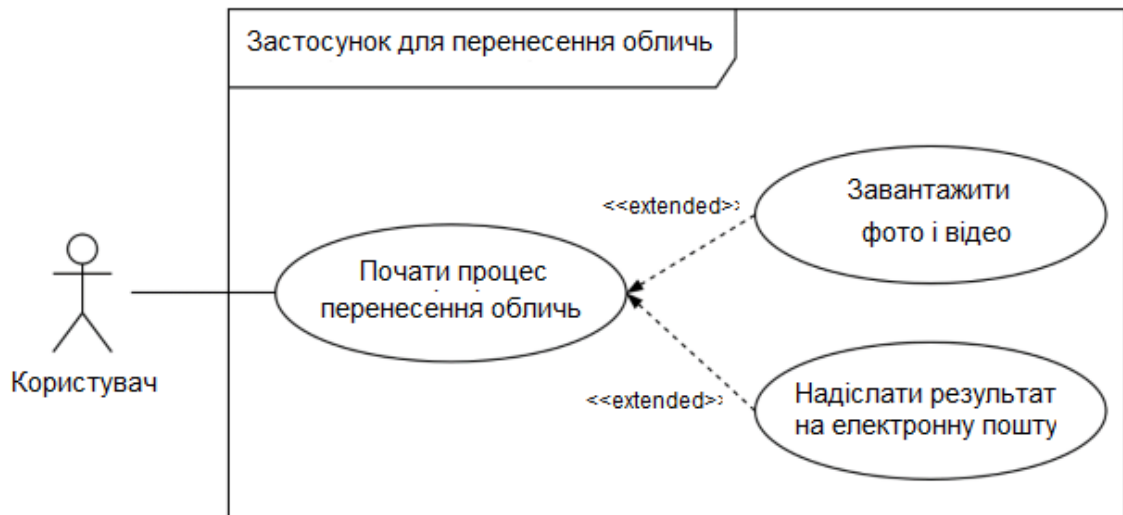


Рисунок 2.11 – Діаграма варіантів використання

Актор - це користувач, який взаємодіє з веб-застосунком. Користувачеві доступний єдиний варіант використання системи «Запустити процес перенесення обличь». Користувач може запустити процес перенесення обличь. Після запуску програма завантажить вказані користувачем фото та відео файли на сервер і почне процес генерації зображення обличчя. Після закінчення процесу генерації результат прийде на вказану користувачем електронну пошту.

2.7.3 Проектування архітектури системи

Система є клієнт-серверним застосунком. Для наочного уявлення архітектури системи було побудовано діаграму розміщення (рис. 2.12).

Користувач взаємодіє з програмою через інтерфейс (Web-interface). Web-server приймає від Web-interface запити, зберігає файли користувача в сховище даних (Data store) і далі відправляє запит виконавцю завдань (Task executor), що складається з брокера зі спілкувань (Message broker) і процесів (Worker), що виконують завдання користувачів. Worker -процеси працюють у нескінченному циклі, очікуючи надходження завдань у чергу. Worker -процеси виконують програмний код, який реалізує етапи перенесення обличь (див. п. 1.1), використовуючи для цього нейронні мережі, запущені на Serving server. Після завершення виконання користувачеві надсилається результат на вказану раніше пошту.

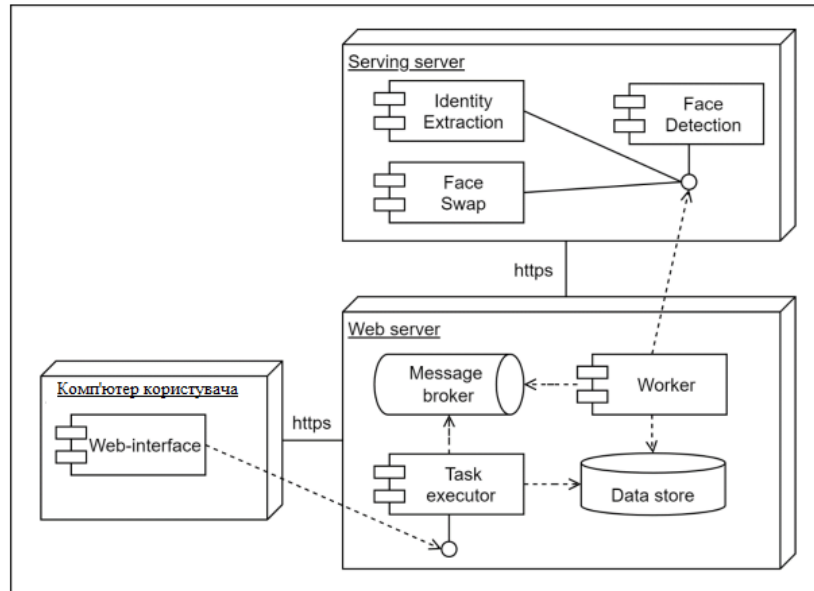


Рисунок 2.12 – Діаграма розміщення

Кожен із перелічених компонентів системи утворює собою окремий Docker -контейнер, що дозволяє розміщувати їх незалежно від розташування інших. Детальний опис кожного компонента описано далі по тексту.

Web-interface. Дозволяє користувачам взаємодіяти із застосунком через інтернет браузер. Web-interface складається з однієї адаптивної веб-сторінки, що містить такі елементи: логотип, назва, заголовок, поля для завантаження фото та відео, поле для введення електронної пошти, кнопка для запуску перенесення обличчя, ілюстрація перенесення обличчя та Welcome -текст. На рис. 2.13 наведено макет веб-сторінки.

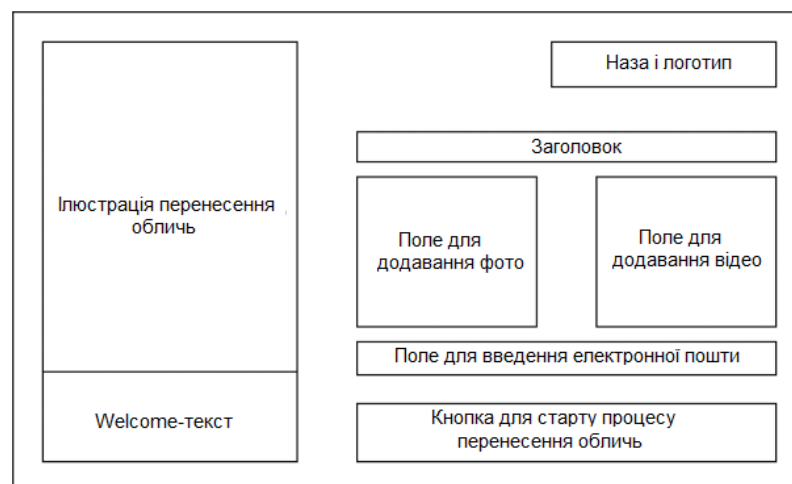


Рисунок 2.13 – Макет сайту для перенесення обличчя

Поля завантаження фото- та відеофайлів повинні надавати користувачеві можливість вибору файлів за допомогою натискання на відповідну кнопку та за допомогою перенесення файлів у область завантаження.

При спробі надсилання порожніх форм, поля для додавання фото- та відеофайлів та поле вводу електронної пошти має виділятися червоною рамкою.

Web-server. Має набір кінцевих точок, які представляють програмний код, який обробляє запит. Табл. 2.1 відображає назви кінцевих точок, назви методів, що вказують на дозволені взаємодії, призначення, вхідні та вихідні дані

Таблиця 2.1 - Список кінцевих точок Web-server

Назва кінцевої точки	Метод	Призначення	Вхідні дані	Вихідні дані
/video_swap	POST	Додати нове завдання перенесення обличчя	- фотографія, - відеофайл, - адреса електронної пошти	Веб-сторінка з подальшими інструкціями користувача
/	GET	Відображення головної сторінки	-	Веб-сторінка

Data store. Одним з компонентів системи, що розробляється, є сховище даних У ньому зберігаються директорії з даними користувачів. Для кожного користувача створюється власна директорія. Назва директорії генерується системою як унікального ідентифікатора з 10 символів. Фото- та відеофайли користувача зберігаються у створену директорію так само з унікальними назвами, що автоматично генеруються, з 6 символів.

Task executor. Для паралельної обробки завдань користувачів необхідно створити кілька окремих процесів (Worker), які прослуховуватимуть чергу повідомлення (Message broker).

Web-server кладе повідомлення у брокер повідомлень, потім Worker - процеси в міру надходження завдань у чергу виконують їх.

Повідомлення для брокера містить наступні параметри: шлях до

директорії користувача в Data store, назва фото- та відеофайлу, E-Mail користувача.

Worker -процес виконує код, що містить у собі всю логіку, необхідну для перенесення обличь, включаючи здійснення етапів, розглянутих у п. 1.1. Для перенесення обличь проводиться передобробка даних: розбиття відео на окремі кадри та вилучення аудіодоріжки. Потім відповідно до алгоритму, описаного в п. 1.1, здійснюється обробка даних. Для отримання інформації про ідентичність, детектування та здійснення перенесення обличь надсилається запит у відповідну модель нейронної мережі на Serving server.

Після процесу перенесення обличь, кадри повертаються на вихідне місце , проводиться постобробка (див. п. 2.5.). Потім кадри збираються в єдине відео і накладається вихідна аудіодоріжка. Результуюче відео надсилається користувачеві на електронну пошту.

Serving server. Розробку застосунків машинного навчання можна розділити на три етапи: model development (розробка алгоритму навчання), training (навчання) та model service (обслуговування моделі) (рис. 2.14).

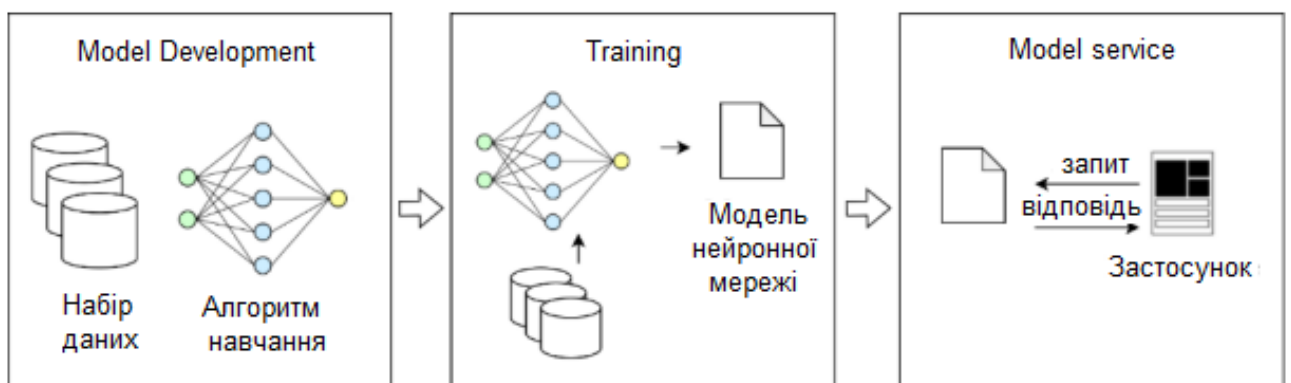


Рисунок 2.14 – Етапи розробки застосунку машинного навчання

Розробка моделі включає в себе збір та підготовку навчальних даних, розробку нових перетворень функцій, вибір архітектури нейронної мережі, а також налаштування різних гіперпараметрів.

Результатом розробки моделі є архітектура мережі, для якої на етапі навчання знаходяться невідомі параметри/ваги. На етапі навчання алгоритм

виконується багато разів в міру надходження нових даних для створення моделі нейронної мережі. Навчання здійснюється за допомогою оптимізації (градієнтного спуску).

На етапі обслуговування модель машинного навчання розміщується в хмарі або локально і через API надає доступ до її функцій.

Кожна з наявних нейронних мереж у застосунку, котрий розробляється, повинна запускатися на сервері в режимі очікування. Serving server має набір кінцевих точок, що є деяким кодом, який обробляє запит. Табл 2.2 відображає назви кінцевих точок, назви методів, що вказують дозволені взаємодії, призначення, вхідні та вихідні дані.

Таблиця 2.2 - Список кінцевих точок Serving server

Назва кінцевої точки	Метод	Призначення	Вхідні дані	Вихідні дані
/face_detector	POST	Отримання координат обличчя та ключових точок	Зображення	Координати обличчя та ключових точок
/identity_extractor	POST	Отримання вектора з інформацією про ідентичність	Координати обличчя та ключових точок	Вектор з інформацією про ідентичність
/face_swap	POST	Отримання зображення з перенесеним обличчям	Координати обличчя та ключових точок, вектор з інформацією про ідентичність	Зображення, маска з виділеним обличчям

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСТОСУНКУ

3.1 Програмні засоби реалізації

Для розробки програмної частини було вибрано високорівневу мову програмування Python версії 3.9.1. Розробка виконувалася із застосуванням редактора PyCharm Community версії 2022.1 [44] .

Інтерфейс користувача був реалізований за допомогою HTML 5 [23] , CSS 3 [10] , і JavaScript [14] . Розробка велася в текстовий редактор з відкритим кодом Notepad++ версії v8.4.1 [38].

Серверна частина застосунку була реалізована з використанням бібліотеки Flask версії 1.1.2 [17]. Для організації черги повідомлень та запуску процесів для виконання завдань використовувалася бібліотека Dramatiq [13]. Для запуску моделей на сервері в режимі очікування використано бібліотеку TorchServe [56].

Для роботи з нейронними мережами були використані такі бібліотеки: numpy версії 1.19.5 [27], PyTorch версії 1.11.0 [46] , torchvision версії 0.2.1 [57] , pillow версії 6.1 і onnxruntime версії 1.1 . Для відстеження ходу навчання нейронної мережі використовувався програмний засіб TensorBoard [54] .

Для автоматичного розгортання та контейнеризації було використано програмний засіб Docker.

3.2 Реалізація Web-interface

Для реалізації Web-interface використовувалася мова розмітки HTML та каскадні таблиці стилів CSS. Розробка велася в текстовий редактор з відкритим кодом Notepad++ версії v8.4.1 [38] .

На рис. 3.1 відображено скріншот сторінки, яка була реалізована для здійснення перенесення обличчя.

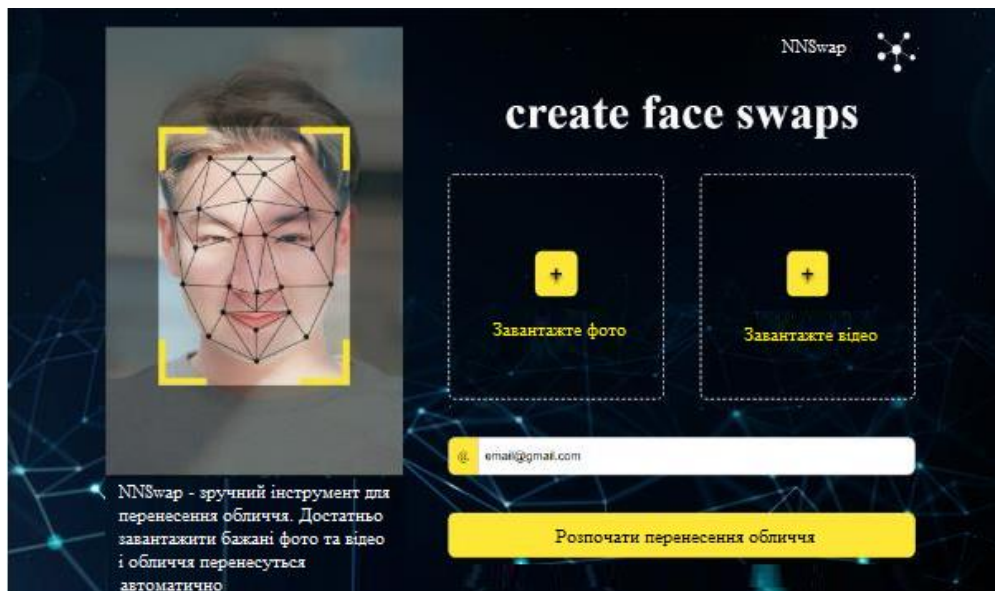


Рисунок 3.1 – Скріншот сторінки сайту

При реалізації Web-interface були використані зображення та відеофайли зі стокового джерела [42]. За допомогою мови JavaScript була реалізована перевірка на валідність даних відповідно до функціональних вимог.

Якщо спробі завантажити файли, які не відповідають вимогам, або у разі надсилання неповністю заповненої форми, на сторінці з'явиться відповідне повідомлення (рис. 3.2). Незаповнене поле підсвічуватиметься червоним контуром (рис. 3.3).

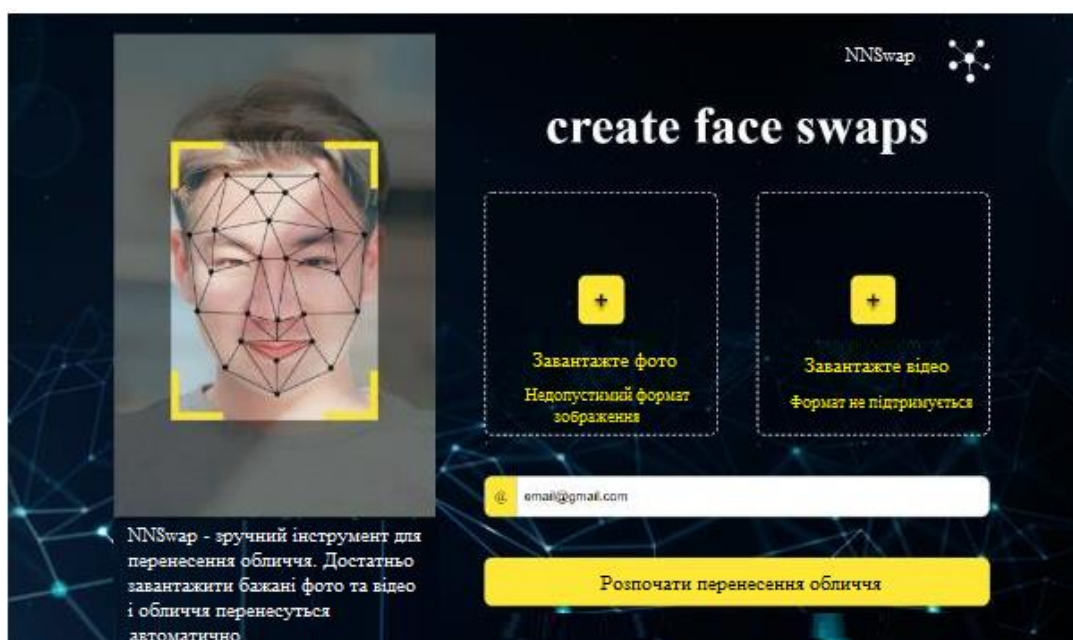


Рисунок 3.2 – Помилка при надсиланні неправильних даних

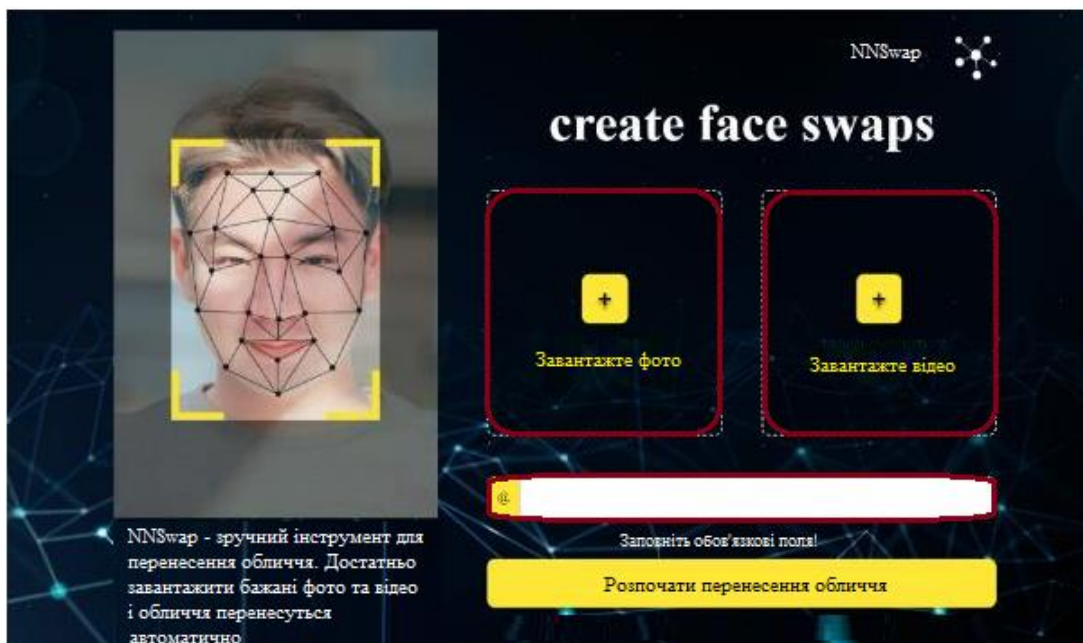


Рисунок 3.3 – Помилка при надсиланні порожньої форми

Користувач має можливість завантаження файлів шляхом натискання на кнопку або за допомогою перетягування файлів у поле завантаження. Ця опція була реалізована за допомогою алгоритму оперування елементами інтерфейсу Drag-and-drop.

При натисканні на кнопку відправки користувач погоджується з умовами конфіденційності, які можна прочитати при переході на посилання. Для надсилання даних на сервер був використаний підхід AJAX.

Веб-сайт має адаптивний дизайн і змінює розміри та розташування контенту в залежності від роздільної здатності екрана пристрою. Рис. 3.4 відображає сторінку веб-сайту на смартфонах.

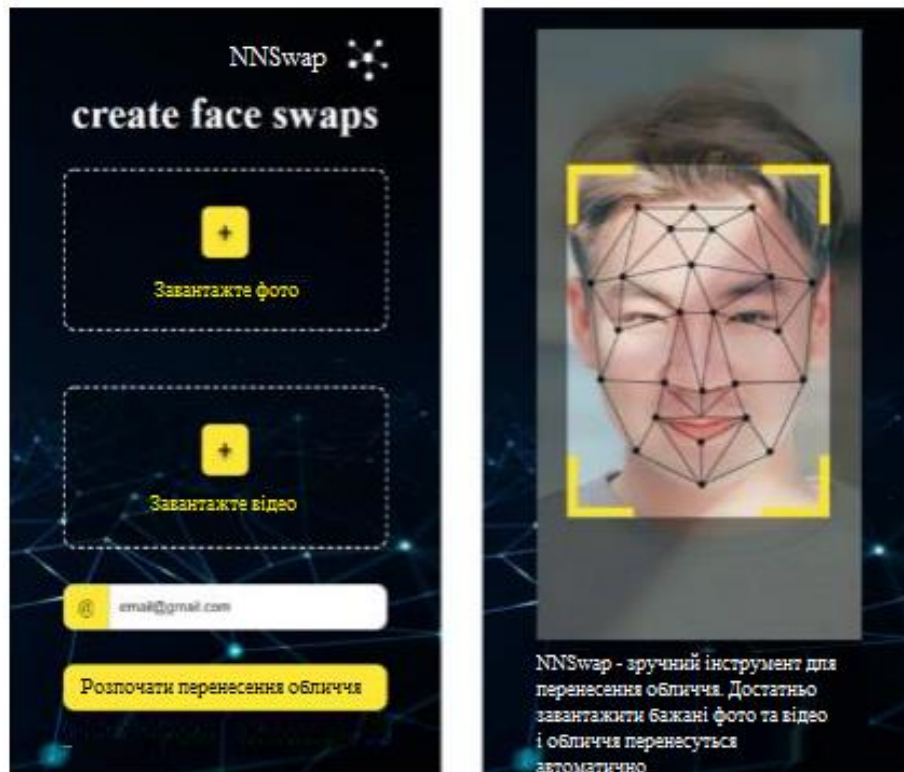


Рисунок 3.4 – Відображення сторінки сайту на смартфоні

3.3 Реалізація Web-server

Web-server є REST API і містить реалізацію методів, описаних у пункті 2.7.3. Для реалізації був використаний фрейм ворк Flask. Лістинг 1 у додатку А відображає реалізацію функції, яка викликається при зверненні до кінцевої точки video_swap.

У методі video_swap відбувається перевірка файлів користувача, створення директорії для завдання, що надійшло, завантаження файлів користувача на сервер у створену директорію. Після чого відбувається відправлення завдання до брокера повідомлень (Message broker).

У разі успішного виконання описаних вище дій, виводиться повідомлення на HTML -сторінку, інакше виводиться повідомлення про помилку.

3.4 Реалізація Task executor

Для реалізації Task executor було використано бібліотеку Dramatiq. Це

бібліотека Python для організації черги завдань за допомогою брокера повідомлень та їх обробки у фоновому режимі робочими процесами. Як брокер повідомлень в Dramatiq використовувалася NoSQL- система Redis [48] .

За допомогою функції `RedisBroker` відбувається ініціалізація брокера повідомлень. Потім за допомогою функції `declare_queue` оголошується повідомлень. Як параметри передається ім'я черги (`queue_name`). На рис 3.5 наведено схему роботи Task executor. Попередньо запускаються Worker -процеси (`worker-1`, `worker-2` тощо). Вони працюють у нескінченному циклі, очікуючи надходження завдання у чергу, яку вони прослуховують. Кількість worker - процесів визначається параметром в Dramatiq і в розробленому застосунку дорівнює кількості ядер процесора. Кількість завдань визначається кількістю надісланих користувачем запитів.

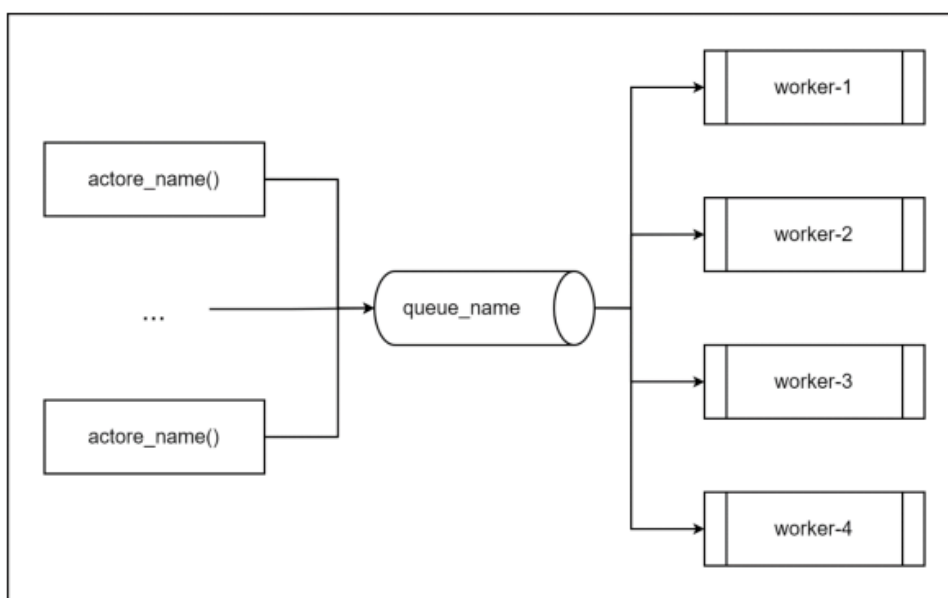


Рисунок 3.5 – Схема роботи Task executor

Для того, щоб поставити завдання в чергу, у Web-server викликається функція `enqueue` (лістинг 3.1). Як аргументи передається назва функції (`actor_name`), ім'я черги (`queue_name`) і асоціативний масив з іменами змінних, необхідних для виконання вказаної функції.

Лістинг 3.1 - Функція enqueue

```
broker.enqueue(Message(queue_name= queue_name, actor_name="processing",  
                        kwargs={"image": image_name, "video": video_name,  
                               "folder": folder_path, "email": email} ))
```

Кожен Worker при отриманні завдання спочатку відправляє запит із зображенням рецепієнта в кінцеву точку identity_extraction і отримує інформацію про ідентичність. З отриманого відео витягується аудіодоріжка і відео розбивається на кадри.

Для здійснення паралельної обробки у кінцеву точку face_detection виправляється кілька зображень. Кількість зазначається константою заздалегідь. Аналогічно надсилаються дані в кінцеву точку face_swap .

Після завершення кадри з'єднуються в єдине відео та накладається аудіодоріжка. Отримане відео надсилається користувачеві на пошту. Функція надсилання повідомлення на електронну пошту реалізована з використанням бібліотеки smtplib [51] .

Лістинг 2 у додатку А відображає реалізацію функції, яку виконують worker -процеси.

3.5 Реалізація Serving server

Для запуску моделей на сервері Serving server у режимі очікування було використано бібліотеку TorchServe. Це бібліотека Python, яка використовує API - інтерфейси на основі REST.

Для запуску TorchServe потрібна наявність файлу з роздільною здатністю ".mar". Це архів, який містить усі необхідні файли для виконання виводу. Основними файлами є сама модель нейронної мережі та написаний для неї обробник (Handlers).

Обробники відповідають за виконання прогнозу з використанням моделі нейронної мережі на основі одного або декількох HTTP -запитів. Обробник є класом, що має три функції: preprocess , inference , postprocess .

Функція preprocess приймає requests (масив запитів). Потім дані, що надійшли, витягуються і перетворюються на вигляд, придатний для передачі в нейронну мережу. Отримані дані передаються у функцію inference, де дані відправляються в нейромережу для отримання результату. Функція postprocess отримує дані від inference, перетворює в придатний для відправки формат і повертає клієнту.

Для кожної нейромережі було написано свій обробник (Handler). Обробник кінцевої точки face_swap крім основного перенесення обличчя, так само здійснює сегментацію обличчя. Лістинг 3 додатка А відображає реалізацію обробника, який викликається при зверненні до кінцевої точки identity_extraction

3.6 Підготовка даних та передобробка набору даних

Розглянуті системи у п. 1.2 використовують нейронні мережі, які навчалися на зображеннях розміром 256x256 пікселів. Даного розміру недостатньо для якісного перенесення обличчя.

Для підвищення якості перенесення обличчя потрібно підготувати набір даних, що складається з зображень 512x512 пікселів.

Крім вимоги до розміру зображення, набір даних повинен враховувати такі фактори [5]: національність; гендерна приналежність; вік; варіації пози голови; освітлення та різноманітність фону; емоції.

Існують відкриті набори даних з високою роздільною здатністю картинок і частково охоплювальні особливості обличчя. Наприклад, у наборі даних ravdess присутні відео 24 молодих чоловіків і жінок з різними емоціями. Набір даних mead охоплює емоції та повороти голови.

Табл. 3.1 відображає аналіз наявності особливостей у різних наборах даних, що підходять для реалізації поставленого завдання.

Таблиця 3.1 - Зведена таблиця аналізу відкритих наборів даних

Набір даних		ffhq [18]	ravdess [47]	hdtf [22]	mead [32]
Фактор					
Національність		+	+	+	+
Гендерна приналежність		+	+	+	+
Вік (років)	до 18	+	-	-	-
	від 18 до 40	+	+	+	+
	від 40	+	-	+	+
Варіації пози голови		+	-	-	+
Освітлення та різноманітність фону		+	-	+	-
Емоції		+	+	-	+

Як видно з таблиці 3.1, існуючі набори даних не повністю охоплюють всі фактори, що потрібні. Набір даних потрібно доповнити різноманітними даними. Для збору даних використовувалися стокові джерела з безкоштовною ліцензією [43, 42].

Підготовка даних проводилася у послідовності, описаній нижче.

1. Зі стокових джерел скачувалися відео, роздільною здатністю не менше HD, оскільки потрібні обличчя 512x512 пікселів. Вибиралися відео, де обличчя людини не сильно пересувалося, щоб уникнути великої кількості «змазаних» кадрів. Також за аналогічними критеріями відбиралися відео з відкритих існуючих наборів даних.

2. Зібрані відео розбивалися на кадри.

3. Видалення повторюваних і «змазаних» кадрів із усіх даних.

4. Виділення обличчя у кадрах. Для виявлення обличчя в кадрі використовувалася нейронна мережа для детектування обличчя SCRFD [21]. Виявлене зображення обличчя вирізалася з кадру, поверталася, так щоб очі були на одній лінії і приводилися до розмірів 512x512 пікселів.

5. Виділення векторів з інформацією про ідентичність. Для кожного зображення виділяється вектор з інформацією про ідентичність, щоб під час

навчання не проводити цю процедуру, тим самим скоротивши час навчання. Для виділення інформації про ідентичність використовувалася нейронна мережа ArcFace [26] .

Лістинг 4 додатку А відображає реалізацію коду, для здійснення пунктів 4 і 5 підготовки набору даних.

При «доборі» даних нахил зроблений на зображеннях, де обличчя перебуває у повороті, зображення з обличчями з різними емоціями і зняті при різному освітленні. Також акцент було зроблено на даних, де особи перебувають на різноманітному тлі. У результаті зібрано близько двох мільйонів зображень обличчя розміром 512x512 пікселів.

На рис. 3.6 представлено приклад 25 зображень із зібраного набору даних, що охоплює всі описані раніше фактори обличчя.



Рисунок 3.6 – Приклад зображень із зібраного набору даних

3.7 Вибір та навчання нейронної мережі

Для кожного етапу, описаного у пункті 1.1. використовується своя нейронна мережа. Таблиця 3.2 відображає інформацію про використані в застосунку нейронні мережі.

Таблиця 3.2 - Список нейронних мереж, що використовуються

Призначення нейронної мережі	Нейронна мережа
Face Detection	SCRFD [25 , 21]
Identity Extraction	ArcFace [25 , 26]
Face Swar	SimSwap [8, 7]
Сегментація	BiSeNet [2,63]

Основною є нейронна мережа для перенесення обличчя. Як архітектура, була обрана архітектура нейронної мережі SimSwap. Топологія цієї нейронної мережі написана за допомогою бібліотеки Pytorch. Рис. 3.7 відображає топологію нейронної мережі, більш детальний опис було наведено у пункті 2.4. "Алгоритм Face Swar". Призначення використовуваних шарів, які відображені в топології, були описані в пункті 2.6. «Основні відомості про нейронні мережі».

Як ініціалізацію ваг генератора і дискримінатора використовувалися ваги з офіційного репозиторію SimSwap [8] . Вони були навчені на наборі даних VGGFace2 [5] , що складається із зображень 256x256 пікселів. Нейронна мережа, навчена на зображеннях низької якості, вже вміє відновлювати обличчя. Тому використання попередньої моделі скоротить процес навчання у кілька разів.

При навчанні генератора та дискримінатора застосовувався оптимізатор Adam [27] при коефіцієнті швидкості навчання 0.0002 та параметрами $\beta_1 = 0$, $\beta_2 = 0,999$. Для навчання використовувалися 99% даних.

Перша частина навчання проходила на персональному комп'ютері з наступними характеристиками: процесор Intel (R) Core (TM) i7-6800K CPU @ 3.40GHz, ОЗУ 64 ГБ, жорсткий диск SSD, графічний прискорювач NVIDIA GeForce GTX 1080. При навчанні розмір batch склав 2 зображення. Навчання проводилося протягом 15 епох із середнім часом навчання 25 годин на епоху.



Рисунок 3.7 – Топологія нейромережі

Друга частина навчання на нейрокомп'ютері. При навчанні розмір batch становив 12 зображень. Навчання проводилося упродовж 40 епох із середнім часом навчання 20 годин на епоху.

Для навчання нейронної мережі на декількох графічних процесорах був використаний інструмент бібліотеки PyTorch - DistributedDataParallel (DDP) [11]. DDP реалізує паралелізм даних на рівні модулів, який може виконуватися на декількох графічних процесорах. При його використанні для кожного доступного графічного процесора породжується свій процес, де ведуться розрахунки градієнтів та помилка. При такому підході кожен процес обробляє тільки виділену йому частину даних. Однак підрахунок метрик та збереження

ваг здійснюється тільки на головному процесі, який має індикатор 0.

3.8 Реалізація методів оцінки якості перенесення обличь

Оцінка візуальної якості не єдиний спосіб оцінити результати перенесення обличь. На основі робіт [9,29,41,7] було виділено список кількісних показників оцінки якості перенесення обличь:

- індекс структурної подібності між двома зображеннями SSIM;
- пікове відношення сигналу до шуму PSNR [24] ;
- евклідова відстань між векторами з інформацією про ідентичність вихідного зображення донора та підсумкового;
- евклідова відстань між кутами Ейлера для вимірювання точності пози голови (обчислена за допомогою FSA-Net).

Для оцінки якості перенесення був розроблений клас Metrics . Він містить функції, що реалізують описані вище методи. Лістинг 3.2 відображає основні функції, що використовуються для реалізації класу Metrics.

Лістинг 3.2 – Основні функції класу Metrics

```
from skimage.measure import compare_ssim
import numpy as np

yaw1,pitch1,roll1 = FSA_net(img1)
yaw2,pitch2,roll2 = FSA_net(img2)

pose = sqrt((yaw1-yaw2)**2+(pitch1-pitch2)**2+(roll1-roll2)**2)
ssim = compare_ssim(img1, img2)
psnr = 20*log10(255 / np.mean((img1 - img2) ** 2))
latent1= netArc(img1)
latent2 = netArc(img2)
ID_dist=np.sqrt(np.sum(np.square(latent1 - latent2)))
```

3.9 Контейнеризація системи

Всі компоненти застосунку, що розробляється, були упаковані в Docker - контейнери. Для організації багатоконтейнерного підходу було використано

спеціальну утиліту Docker Compose. Були виділені такі Docker – контейнери:

- api містить REST API;
- redis - брокер повідомлень, що підтримує чергу прийнятих завдань, а також бекенд, що зберігає результати виконаних завдань;
- torchserve - запуск служби моделей нейронних мереж;
- task - запускаючий процеси worker, що обслуговують чергу.

Лістинг 6 додатка А відображає текст конфігураційного файлу docker-compose.yml, що містить опис контейнерів та опис взаємодії контейнерів між собою.

3.10 Тестування застосунку

3.10.1 Тестування веб-сайту

Проведено тестування сайту на адаптивність. Виходячи з вимог, сайт повинен відображатися коректно при всіх роздільних здатностях екрану. За допомогою зміни роздільної здатності екрана через браузер було проведено шість тестів. Таблиця Б2 додатка Б відображає результати тестування .

Проведено тестування сайту на відображення в різних браузерах. Таблиця Б3 додатка Б відображає результати тестування.

3.10.2 Функціональне тестування

Було проведено функціональне тестування програми з метою перевірки функціональності, що реалізовується. Таблиця Б1 додатку Б представляє набір функціональних тестів.

Тестування проводилося вручну, всі тести пройшли успішно.

3.10.3 Тестування нейронної мережі

Для відстеження процесу навчання нейронної мережі використовувався програмний засіб TensorBoard.

Під час навчання помилка для дискримінатора збільшується (рис. 3.8) , а генератора зменшується (рис. 3.9) . Це говорить про те, що з плином навчання

генератор починає краще переносити особи, а дискримінатор гірше відрізняти реальне зображення і результуюче.

З графіків видно, що помилка на навчальних даних становить 0,65 і 5,3 пункту для дискримінатора та генератора відповідно. Також було проведено тестування на 5% даних, які були відсутні у навчальній вибірці. Середня помилка для дискримінатора становила 0,58 пунктів, для генератора 5,5 пунктів.

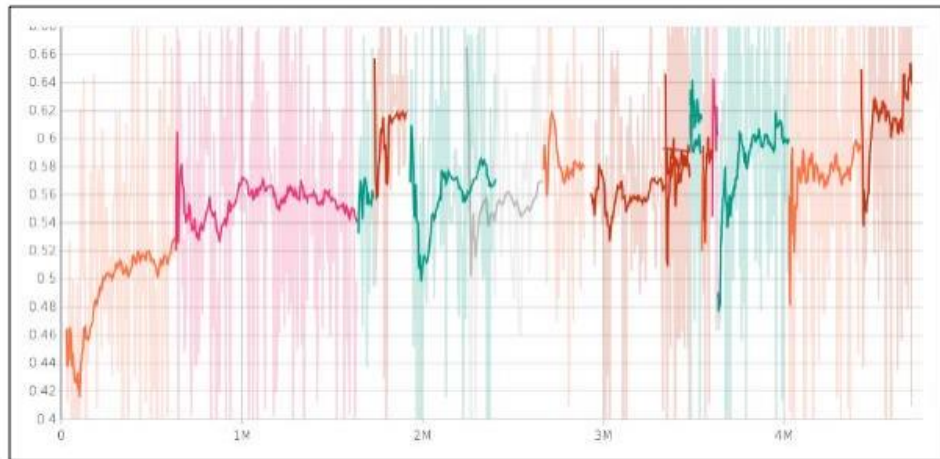


Рисунок 3.8 – Графік помилки дискримінатора під час навчання

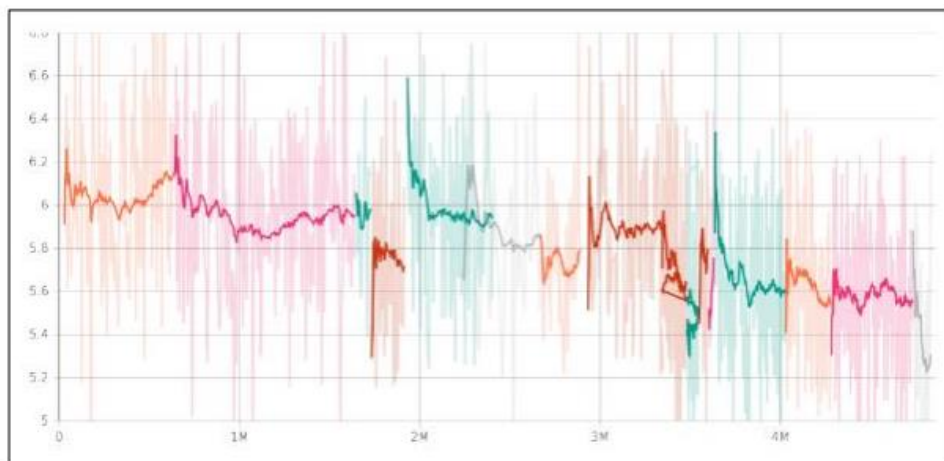


Рисунок 3.9 – Графік помилки генератора під час навчання

Було проведено оцінку якості перенесення обличь за допомогою метрик, описаних у пункті 3.8. Порівняння було проведено із системами [9, 7] . У табл. 3.3 відображено результати оцінки якості перенесення обличь.

З табл. 3.3 видно, що показники, пов'язані з якістю перенесення обличь

(SSIM та PSNR) збільшилися на кілька відсотків. Однак показники, пов'язані з точністю перенесення обличчя (точність пози і збереження ідентичності), трохи нижче за показники аналогічних систем. Це з великим розміром набору даних. Для підвищення цих показників потрібне додаткове навчання нейронної мережі.

Таблиця 3.3 – Результат оцінки якості перенесення обличчя

Система	SSIM	PSNR	Точність пози	Збереження інформації про ідентичність
SimSwap	70%	30%	73%	30%
SberSwap	69%	30%	69%	27%
NNSwap (my)	82%	33%	72%	29%

Рис. 3.10 відображає приклад перенесення обличчя за допомогою навченої нейронної мереж

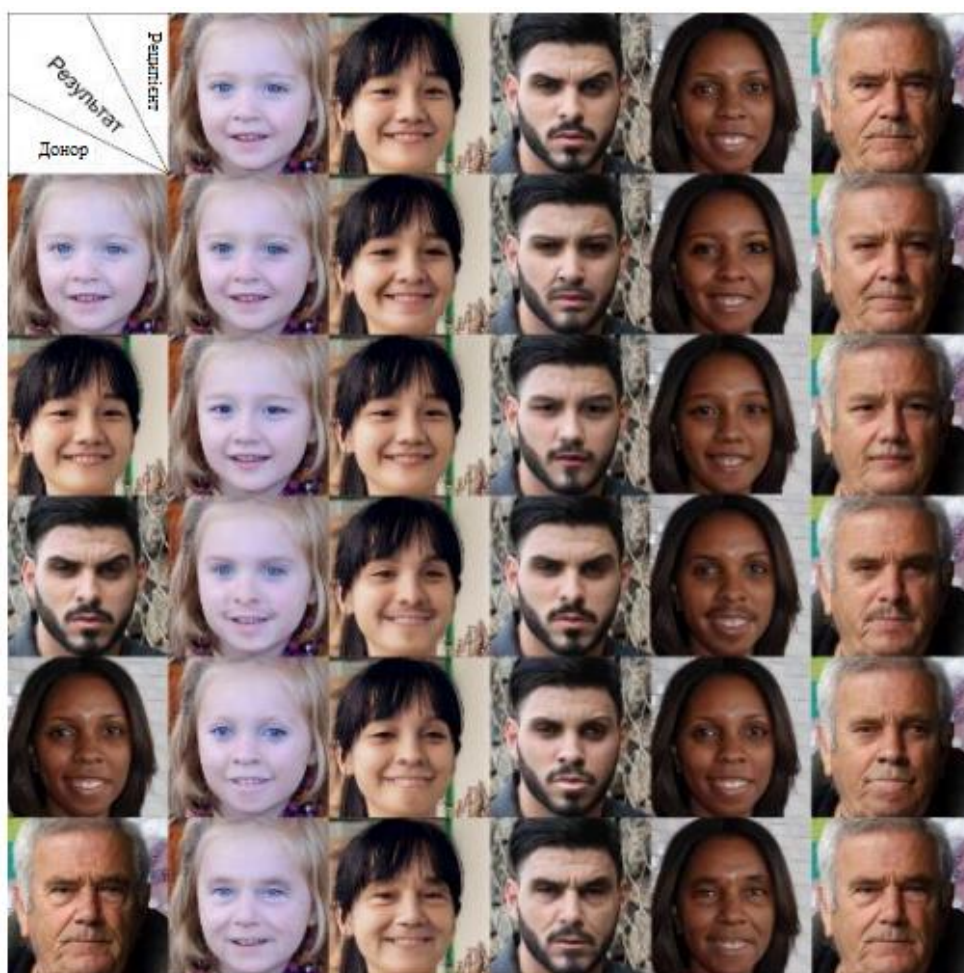


Рисунок 3.10 – Результат навчання нейронної мережі

З рисунку можна побачити, що найбільш ефективно переносяться форми очей, брів, губ та носа. Так само частково переносяться носогубні складки, зморшки, вуса, борода та характерна текстура обличчя. При заміні обличчя зберігається міміка та поза голови реципієнта. Нейронна мережа переносить обличчя з гарною якістю і без сторонніх шумів навколо за рахунок використання сегментаційних масок. Також варто відзначити відсутність явного переходу між фоном і накладеним обличчям. Це обумовлено використанням розмиття меж накладеної маски.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Навчання працюючих і інструктажі з охорони праці

Однією із складових ефективної роботи з профілактики виробничого травматизму є належна підготовка, навчання та підвищення кваліфікації працівників з питань охорони праці. Загальний порядок проведення навчання з питань охорони праці встановлений Законом України «Про охорону праці» (ст. 18. «Навчання з питань охорони праці»).

Виконання вимог Закону України «Про охорону праці» в частині проведення навчання та перевірки знань з питань охорони праці здійснюється відповідно до Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держкомітету України з нагляду за охороною праці 26 січня 2005 р. № 15 (далі — Типове положення).

Нагляд за дотриманням вимог Типового положення здійснюють органи державного нагляду за охороною праці, а координацію та методичний супровід — Головний навчально-методичний центр та навчальні підрозділи експертно-технічних центрів Держгірпромнагляду.

Вивчення предмета «Охорона праці» при підготовці, перепідготовці та підвищенні кваліфікації працівників, які залучаються до виконання робіт з підвищеною небезпекою, на підприємстві регламентується п. 2.3 Типового положення. На підприємствах згідно з п. 1.1 Додатку 3 Типового положення навчання та перевірку знань з питань охорони праці повинні проходити керівники, заступники керівників, головні спеціалісти, керівники основних виробничих та технічних служб, безпосередньо пов'язані з організацією безпечного ведення робіт. Крім цього, згідно з п. 5 Додатку 3, навчання та перевірку знань з питань охорони праці мають проходити керівники, спеціалісти служб охорони праці, члени комісій з перевірки знань з питань охорони праці, особи, відповідальні за технічний стан і безпечну експлуатацію об'єктів

підвищеної небезпеки підприємств.

Типове положення встановлює порядок та місце проведення навчання та перевірки знань з питань охорони праці посадових осіб (п. 5.2 та п. 5.3). Посадові особи, перелік яких наведено в п. 5.2, проходять навчання у Головному навчально-методичному центрі Держнаглядохоронпраці. Перевірка знань цієї категорії посадових осіб проводиться комісією, створеною наказом Держгірпромнагляду.

Організацію навчання та перевірки знань з питань охорони праці працівників на підприємстві здійснюють працівники служби кадрів або інші спеціалісти, яким роботодавець доручив організацію цієї роботи. Навчання та перевірка знань з питань охорони праці працівників (виконавців і посадових осіб), які не залучаються до виконання робіт підвищеної небезпеки, проводиться не рідше ніж один раз на три роки. Посадові особи та працівники, які виконують роботи підвищеної небезпеки, проходять спеціальне навчання та перевірку знань відповідних нормативно-правових актів з охорони праці не рідше одного разу на рік.

Посадові особи малих підприємств (п. 5.4), які не мають можливості створити власні комісії з перевірки знань з питань охорони праці та провести навчання з питань охорони праці, проходять навчання та перевірку знань в навчальних закладах, які мають відповідний дозвіл.

Спеціальне навчання з питань охорони праці може проводитись безпосередньо на підприємстві або навчальним закладом, який має відповідний дозвіл. При проведенні такого навчання на підприємстві навчальні плани та програми розробляються з урахуванням конкретних видів робіт, виробничих умов і функціональних обов'язків працівників і затверджуються наказом керівника підприємства.

Періодичність інструктажів, навчання та перевірки знань з питань охорони праці залежить від видів виконуваних робіт та встановлюється Типовим положенням. Перевірка знань з питань охорони праці після проведення спеціального навчання проводиться комісією підприємства.

Якщо на підприємстві неможливо створити комісію з перевірки знань з

питань охорони праці (п. 4.4 Типового положення), перевірка знань проводиться комісією спорідненого підприємства або Теруправління Держгірпромнагляду.

Всі працівники та посадові особи підприємства, включаючи посадових осіб, відповідальних за виконання робіт підвищеної небезпеки (крім зазначених в п. 5.2 та п. 5.3 Типового положення), проходять навчання та перевірку знань з питань охорони праці на підприємстві. Типове положення не зобов'язує, але й не забороняє проводити навчання всіх виконавців та посадових осіб (особливо тих, що виконують роботи підвищеної небезпеки) в навчальних закладах. У нашій країні є багато підприємств, де таке навчання проводиться, і це має позитивні наслідки. Ті витрати, які при цьому несуть підприємства, перекриваються створенням більш безпечних умов праці і в результаті збереженням життя та здоров'я працівників.

Також в навчальних закладах проходять навчання та перевірку знань із загальних питань охорони праці всі посадові особи та фахівці, які проводять інструктажі або навчання підлеглих працівників з питань охорони праці, виконують роботи з проектування об'єктів, а також інші працівники, незалежно від того, передбачено таке навчання Типовим положенням чи ні.

4.2 Санітарно-гігієнічні вимоги до умов праці

На робочих місцях працівників, які відповідальні за експлуатацію сервісу управління механізмом авторських прав на мультимедійні файли, необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Приміщення для роботи з ЕОМ мають бути обладнані системами опалення, кондиціонування повітря, або припливно-витяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості та рухливості повітря відповідно до норм та правил, а також ДБН В.2.5-67:2013 «Опалення, вентиляція та кондиціонування», затверджених наказом Мінрегіону від 25.01.2013 р. № 24.

Відповідно до санітарних норм мікроклімату виробничих приміщень ДСН 3.3.6.042-99 в офісних приміщеннях, обладнаних ЕОМ, температура повітря повинна становити 22-25°C, відносна вологість повітря – 40-60 %, швидкість руху повітря – не більше 0,1 м/с [60].

Приміщення, призначені для роботи з ЕОМ, повинні мати природне освітлення. У виробничих приміщеннях, обладнаних ЕОМ, необхідно створити належне освітлення. При експлуатації сервісу управління механізмом авторських прав на мультимедійні файли, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18 [61]. Природне світло повинно бути бічним, зорієнтованим, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості не нижче 1,5%. При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами Держпромгірнагляду та органами й установами санітарно-епідеміологічної служби. Вікна приміщень повинні мати регулювальні пристрої для відчинення, а також жалюзі, штори тощо. Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути всеосяжним і рівномірним. У випадку, коли переважають роботи з документами, допускається комбіноване освітлення (додатково до загального освітлення встановлюється світильники місцевого освітлення). Світильники розміщуються збоку від робочих місць (переважно ліворуч), або локально над робочим місцем (при розташуванні відеотерміналів ЕОМ за периметром приміщення). Як джерело світла при штучному освітленні застосовуються, як правило, люмінесцентні лампи. У світильниках місцевого освітлення допускається застосування ламп розжарювання. Рівень освітленості на робочому місці має становити 300-500 лк. При використанні комбінованого освітлення не допускається відблисків на поверхні екрана та збільшення освітлення екрана вище 300 лк.

Орієнтація вікон повинна бути на північ або північний схід, вікна повинні мати жалюзі, які можна регулювати, або штори; не дозволяється розміщувати кабінети обчислювальної техніки у підвальних приміщеннях будинків; кабінети,

обладнані комп'ютерною технікою, в навчальних закладах повинні розміщуватись в окремих приміщеннях з природним освітленням та організованим обміном повітря; стіни, стеля і підлога та обладнання кабінетів комп'ютерної техніки повинні мати покриття із матеріалів з матовою фактурою з коефіцієнтом відбиття: стін — 40- 50 %, стелі — 70 - 80 %, підлоги — 20-30 %, предметів обладнання — 40-60 % (робочого столу — 40-50 %, корпуса дисплею та клавіатури — 30-50 %, стелажів — 40-60 %); поверхня підлоги повинна мати антистатичне покриття та бути зручною для вологого прибирання; забороняється використовувати для оздоблення інтер'єру приміщень комп'ютерних кабінетів полімерні матеріали (дерев'яно-стружкові плити, шпалери, що придатні для миття, плівкові та рулонні синтетичні матеріали, шаровий паперовий пластик та ін.), що виділяють у повітря шкідливі хімічні речовини, які перевищують гранично допустимі концентрації; вміст шкідливих хімічних речовин в повітрі дошкільних та учбових приміщень з комп'ютерною технікою не повинен перевищувати середньодобові концентрації [61].

Організація робочого місця фахівця із експлуатації сервісу повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».

Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Рівень шуму не повинний перевищувати: на місцях, де працюють програмісти та оператори ЕОМ, 55 дБА, у лабораторіях, де складаються алгоритми та ведеться робота з документацією – 60 дБА, у машинному залі – 65 дБА, у приміщеннях, де розміщені гучні агрегати обчислювальних машин – 75 дБА.

ВИСНОВКИ

У рамках цієї роботи було розроблено застосунок для перенесення обличь на відео на основі нейронних мереж. При цьому було вирішено такі завдання.

- проведено огляд наукової літератури;
- здійснено збір та передобробку даних для навчання;
- навчено нейронну мережу та проведено оцінку результатів її роботи;
- розроблено програму для перенесення обличь на відео на основі нейронних мереж;
- проведено тестування програми;
- виконано дослідження ефективності використаних алгоритмів розпізнавання обличь.

Надалі планується покращити якість роботи системи шляхом додаткового навчання нейронної мережі та зміною її архітектури. Наявний набір даних буде доповнюватися новими більш різноманітними даними для покращення якості перенесення. Для зниження міжкадрового переходу планується розробка модуля постобробки.

ПЕРЕЛІК ДЖЕРЕЛ

1. BetSwap. [Електронний ресурс] - Режим доступа: <https://betswap.gg/> (дата звернення: 20.02.2025).
2. BiSeNet. [Електронний ресурс] - Режим доступа: <https://github.com/CoinCheung/BiSeNet> (дата звернення: 10.05.2025).
3. Bitouk D., Kumar N., Dhillon S., Belhumeur P. Face swapping: Automatically replacing faces in photographs // Association for Computing Machinery, New York, NY, USA, August 2008. Vol. 27, no. 3. P. 1–8. DOI:10.1145/1399504.1360638.
4. Blanz V., Scherbaum K., Vetter T. Exchanging Faces in Images // Computer Graphics Forum. 2004. Vol. 23, no. 3. P. 669–676. DOI: 10.1111/j.1467-8659.2004.00799.x.
5. Cao Q., Shen L., Xie W., Parkhi O., Zisserman A. VGGFace2: A Dataset for Recognising Faces across Pose and Age // 13th IEEE International Conference on Automatic Face & Gesture Recognition 13th, 2018, Xi'an, China, May 15-19, 2018. IEEE Computer Society. P. 67–74. DOI:10.1109/FG.2018.00020.
6. Chen D., Chen Q., Wu J., Yu X., Jia T. Face Swapping: Realistic Image Synthesis Based on Facial Landmarks Alignment // Mathematical Problems in Engineering. 2019. Vol. 2019. P. 1–11. DOI: 10.1155/2019/8902701.
7. Chen R., Chen X., Ni B., Ge Y. SimSwap: An Efficient Framework For High Fidelity Face Swapping // Proceedings of the 28th ACM International Conference on Multimedia. Association for Computing Machinery, New York, NY, USA, October 2020. P. 2003–2011. DOI: 10.1145/3394171.3413630.
8. Chen X. SimSwap: An Efficient Framework for High Fidelity Face Swapping. [Електронний ресурс] - Режим доступа: <https://github.com/neuralchen/SimSwap> (дата звернення: 01.02.2025).
9. Chesakov D., Maltseva A., Groshev A., Kuznetsov A., Dimitrov D. A new face swap method for image and video domains: a technical report // arXiv:2202.03046, 2022.

10. CSS. [Электронный ресурс] - Режим доступа: <https://www.w3.org/Style/CSS/> (дата звернения: 11.05.2025).
11. DistributedDataParallel. [Электронный ресурс] - Режим доступа: <https://pytorch.org/docs/master/notes/ddp.html> (дата звернения: 24.05.2025).
12. Docker. [Электронный ресурс] - Режим доступа: <https://www.docker.com/> (дата звернения: 10.05.2025).
13. Dramatiq: background tasks. [Электронный ресурс] - Режим доступа: <https://dramatiq.io/> (дата звернения: 10.04.2025).
14. ECMA. [Электронный ресурс] - Режим доступа: <https://www.ecma-international.org/publications-and-standards/standards/> (дата звернения: 11.05.2025).
15. FaceApp: Most Popular Selfie Editor. [Электронный ресурс] - Режим доступа: <https://www.faceapp.com/> (дата звернения: 20.02.2025).
16. FaceSwapLive. [Электронный ресурс] - Режим доступа: <http://faceswaplive.com/> (дата звернения: 20.02.2025).
17. Flask. [Электронный ресурс] - Режим доступа: <https://flask.palletsprojects.com/en/2.1.x/> (дата звернения: 20.04.2025).
18. Flickr-Faces-HQ Dataset (FFHQ). [Электронный ресурс] - Режим доступа: <https://github.com/NVlabs/ffhq-dataset> (дата звернения: 22.11.2021 г.).
19. Goodfellow I., Pouget-Abadie J., Mirza M. Xu. Generative Adversarial Networks // arXiv:1406.2661, 2014.
20. Google Chrome. [Электронный ресурс] - Режим доступа: https://www.google.com/intl/uk_ua/chrome/ (дата звернения: 24.05.2025).
21. Guo J., Deng J., Lattas A., Zafeiriou S. Sample and Computation Redistribution for Efficient Face Detection // arXiv:2105.04714, 2021.
22. HDTF. [Электронный ресурс] - Режим доступа: <https://github.com/MRzzm/HDTF> (дата звернения: 15.12.2021 г.).
23. HTML. [Электронный ресурс] - Режим доступа: <https://html.spec.whatwg.org/multipage/> (дата звернения: 11.05.2025).
24. Huynh-Thu Q., Ghanbar, M. Scope of validity of PSNR in image/video quality assessment // Electronics Letters. 2008. Vol. 44, no. 13. P. 800–801. DOI:10.1049/el:20080522.

25. InsightFace: 2D and 3D Face Analysis Project. [Электронный ресурс] - Режим доступа: <https://github.com/deepinsight/insightface> (дата звернения: 01.04.2025).
26. Jiankang D., Jia G., Niannan X., Stefanos Z. ArcFace: Additive Angular Margin Loss for Deep Face Recognition // IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16– 20, 2019. Computer Vision Foundation. IEEE, P. 4690–4699. DOI: 10.48550/arXiv.1801.07698.
27. Kingma D., Ba J. Adam: A Method for Stochastic Optimization // Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, CA, USA, May 7-9, 2015. DOI: 10.48550/arXiv.1412.6980.
28. Kortli Y., Jridi M., Atri M. Face Recognition Systems: A Survey. // Sensors. 2020. Vol. 20, no. 2. Article number 342. DOI: 10.3390/s20020342.
29. Liu K., Wang P., Zhou W., Zhang Z. Face Swapping Con-sistency Transfer with Neural Identity Carrier // Future Internet. 2021. Vol. 13, no. 11. Article number 298. DOI: 10.3390/fi13110298.
30. Liu Z., Luo P., Wang X., Tang X. Deep Learning Face Attributes in the Wild // 2015 IEEE International Conference on Computer Vision (ICCV). 2015. DOI: 10.1109/ICCV.2015.425.
31. Mahajan S., Chen L., Tsai T. SwapItUp: A Face Swap Application for Privacy Protection // 31st IEEE International Conference on Advanced Information Networking and Applications. 2017. DOI: 10.1109/AINA.2017.53.
32. MEAD. [Электронный ресурс] - Режим доступа: <https://wywu.github.io/projects/MEAD/MEAD.html> (дата звернения: 20.12.2021 г.).
33. Microsoft Edge. [Электронный ресурс] - Режим доступа: https://www.microsoft.com/uk_ua/edge (дата звернения: 24.05.2025).
34. Minaee S., Boykov Y., Porikli F., Plaza A., Kehtarnavaz N., Terzopoulos D. Image Segmentation Using Deep Learning: A Survey // arXiv:2001.05566, 2020.
35. Mozilla. [Электронный ресурс] - Режим доступа: <https://www.mozilla.org/firefox/new/> (дата звернения: 24.05.2025).

36. Nirkin Y., Keller Y., Hassner T. FSGAN: Subject Agnostic Face Swapping and Reenactment // 2019 IEEE/CVF International Conference on Computer Vision. 2019. DOI: 10.48550/arXiv.1908.05932.

37. Nirkin Y., Masi I., Tran A., Hassner T., Medioni G. Face Segmentation, Face Swapping, and How They Impact Face Recognition // Deep LearningBased Face Analytics. 2021. P. 21–43. DOI: 10.1007/978-3-030-74697-1_2.

38. Notepad-plus-plus. [Электронный ресурс] - Режим доступа: <https://notepadplus-plus.org/> (дата звернения: 10.05.2025).

39. Onnxruntime. [Электронный ресурс] - Режим доступа: <https://onnxruntime.ai/> (дата звернения: 22.04.2025).

40. Opera. [Электронный ресурс] - Режим доступа: <https://www.opera.com/> (дата звернения: 24.05.2025).

41. Perov I. et al. DeepFaceLab: Integrated, flexible and extensible faceswapping framework // arXiv: 2005.05535, 2021.

42. Pexels. [Электронный ресурс] - Режим доступа: <https://www.pexels.com> (дата звернения: 10.11.2021 г.).

43. Pixabay. [Электронный ресурс] - Режим доступа: <https://pixabay.com> (дата звернения: 1.12.2021 г.).

44. PyCharm. [Электронный ресурс] - Режим доступа: <https://www.jetbrains.com/pycharm/> (дата звернения: 25.01.2025).

45. Python. [Электронный ресурс] - Режим доступа: <https://www.python.org> (дата звернения: 19.03.2025).

46. Pytorch. [Электронный ресурс] - Режим доступа: <https://pytorch.org> (дата звернения: 19.03.2025).

47. Ravdess. [Электронный ресурс] - Режим доступа: <https://smartlaboratory.org/ravdess/> (дата звернения: 11.12.2021 г.).

48. Redis [Электронный ресурс] - Режим доступа: <https://redis.io/> (дата звернения: 10.04.2025).

49. Refese. [Электронный ресурс] - Режим доступа: <https://hey.reface.ai> (дата звернения: 20.02.2025).

50. Siarohin A. et al. Motion-supervised Co-Part Segmentation // International Conference on Pattern Recognition 25, 2020. P. 9650–9657. DOI:10.1109/ICPR48806.2021.9412520.
51. SMTP protocol client. [Електронний ресурс] - Режим доступа: <https://docs.python.org/3/library/smtplib.html> (дата звернення: 25.05.2025).
52. Snapchat. [Електронний ресурс] - Режим доступа: <https://www.snapchat.com> (дата звернення: 20.02.2025).
53. Swapper. [Електронний ресурс] - Режим доступа: <https://icons8.com/swapper> (дата звернення: 20.02.2025).
54. Tensorboard. [Електронний ресурс] - Режим доступа: <https://www.tensorflow.org/tensorboard> (дата звернення: 22.04.2025).
55. The Limitations of Neural Networks: 3 Big Issues Dominate. [Електронний ресурс]. - Режим доступа: <https://techmonitor.ai/techonology/emergingtechnology/limitations-of-neural-networks> (дата звернення: 07.04.2021 г.).
56. TorchServe. [Електронний ресурс] - Режим доступа: <https://pytorch.org/serve/> (дата звернення: 22.04.2025).
57. Torchvision. [Електронний ресурс] - Режим доступа: <https://pytorch.org/vision/stable/index.html> (дата звернення: 10.02.2025).
58. Wang M., Deng W. Deep Face Recognition: A Survey. // Neurocomputing. 2018. Vol. 429. P. 215–244. DOI: 10.1016/j.neucom.2020.10.081.
59. Wang Z., Bovik A. C., Sheikh H. R., Simoncelli E. P. Image Quality Assessment: from Error Visibility to Structural Similarity // IEEE Transactions on Image Processing. 2004. Vol. 13, no. 4. P. 600–612. DOI: 10.1109/TIP.2003.819861.
60. Заїкіна Д., Глива В. Основи охорони праці та безпека життєдіяльності. 2019. URL: <https://doi.org/10.31435/rsglobal/001> (дата звернення: 14.04.2025).
61. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання / укл.: Стручок В. С. Тернопіль: ФОП Паляниця В. А., 2022. 156 с.

ДОДАТКИ

Основні функції додатку

Лістинг 1 - Реалізація функції video_swap

```

@blueprint.route('/video_swap',
methods=['POST','GET']) def video_swap():
response = {"status": "Запущено"}
if request.method=='POST':
if not request.files["image"] or not request.files["video"] or
not
request.form.get('email'): response["status"] = "Дані не
вказані"

return jsonify(response)
image = request.files["image"]
, image_ext = os.path.splitext(image.filename)
video = request.files["video"]
, video_ext = os.path.splitext(video.filename)

if (image_ext.lower() not in valid_image_ext) or (video
ext.lower()
not in valid_video_ext): response["status"] = "Неправильний
формат даних"

return jsonify(response)
task_folder = uuid4().hex
save_folder = os.path.join (UPLOAD_FOLDER, task_folder) if not
os.path.exists(save_folder): os.mkdir(save_folder)
image.save(os.path.join(save_folder,

f"{uuid4().hex[:10]}{image_ext}")) video.save(os.path.join(save
folder,
f"{uuid4().hex[:10]}{video_ext}"))

email = request.form.get('email')
broker.enqueue(Message(queue_name=ApiConfig.PROCESSING_QUEUE
NAME,

actor_name="processing", kwargs={"image_name": image
name,
"video name": video name,
"task folder": task folder,
"email": email}, args=(), options={}))

response["status"] = "Відео прийде вам на пошту"
return jsonify(response)

```

Лістинг 2 - Реалізація функції video_swap (фрагмент)

```
def processing(image name, video name, task folder,
email): logging.info("Start processing")
video path = os.path.join(ApiConfig.UPLOAD FOLDER, task
folder,
video name)

try:
driving video = imageio.mimread(video path, memtest =
False) except Exception:
import traceback as tb logging.exception(tb.format
exc()) return False

video forcheck = VideoFileClip(video path)
fps = video forcheck.fps if video forcheck.audio is
None: no audio = True else:
no audio = False del video forcheck

if not no audio:
video clip = AudioFileClip(video path)
frames folder = os.path.join(ApiConfig.UPLOAD FOLDER,
task folder,
"frames")

if not os.path.exists(frames folder): os.mkdir(frames
folder)

bs = ApiConfig.BATCH_SIZE parts = len(driving video) //
bs crop size = 224
r = requests.post(ApiConfig.ARCFACE URL,
files={"body": open(os.path.join(APIConfig.UP LOAD
FOLDER, task folder, image name), 'rb').read()) latent
id = torch.cuda.FloatTensor(r.json()[0])

for i in range(parts + 1): multiple files = [] body num
= 0 images = []
for j in range(i*bs, np.minimum(len(driving
video), (i+1)*bs)): image = driving video[j][..., :3]
success, encoded image = cv2.imencode('.jpg', image)
multiple files.append(('body'+('' if body num==0 else
str(body num)),
('k', encoded image.tobytes()))

body num += 1 images.append(image) if body num == 0:
continue
r = requests.post(ApiConfig.FACE DETECTOR URL,
files=multi
ple files)
result = pickle.loads(r.content) for image, res in
zip(images, result): bbox = np.array(["bboxes"]) kps =
np.array(res["kpss"])
....
```

Лістинг 3 - Реалізація обробника IdentityExtraction (фрагмент)

```
class IdentityExtractionHandler(BaseHandler):
def init (self):
super(). init() self.model = None self.device = None
self.initialized = False

def initialize(self, context):
properties = context.system properties
self.device = torch.device(Config.DEVICE
if torch.cuda.is available() else "cpu")

model dir = properties.get("model dir")

model path = os.path.join(model dir, Config.ARCFACE
MODEL) self.model = model checkpoint['model'].module
self.model = self.model.to(self.device) self.
self.transform = transforms.Compose([
transforms.ToTensor(),
transforms.Normalize([0.485,0.456,0.406],[0.229,0.224,
0.225]) self.initialized = True
...

```

Лістинг 4 - Реалізація функції для підготовки набору даних (фрагмент)

```
def detect():
app = Face detect crop()
app.prepare(ctx id=0, det thresh=0.1, det size=(640,
640)) netArc checkpoint = torch.load(opt.netArc
checkpoint,
map location='cuda')
netArc = netArc checkpoint['model'].module netArc =
netArc.to(opt.device) netArc.eval()
i = 1
images list = glob.glob(opt.dataroot + '\\*.png')
with torch.no grad():
img a whole = cv2.imread(image path) if img a whole is
None:
print('wrong picture:', image path) continue
img a align crop = app.get(img a whole,

opt.extract size)
if not img align crop: continue
img a align crop pil = Image.fromarray(cv2.cvtColor
(img a align crop[0], cv2.COLOR_BGR2RGB)) img a =
transformer Arcface(img a align crop pil)
img id = img a.view(-1, img a.shape[0],
img a.shape[1], img a.shape[2]) img id = img
id.to(opt.device)
img id downsample = F.interpolate(img id, scale
factor=112
opt.extract size)

```

Тестування застосунку

Таблиця Б1 - Функціональне тестування

№	Ціль	Дія	Очікуваний результат	Тест пройдено?
1	Перевірити завантаження файлів за допомогою вибору файлів.	Користувач натискає кнопку «За вантажити фото» або кнопку «Завантажити відео ».	Файл завантажується на сервер. Після закінчення результат приходить на пошту.	Так
2	Перевірити завантаження файлів за допомогою перетягування файлів.	Користувач перетягує файл у область завантаження фото- або відеофайлу.	Файл завантажується на сервер. Після закінчення результат приходить на пошту.	Так .
3	Перевірити завантаження файлів дозволеного розміру.	Користувач завантажує фото менше 5МБ та відео менше 50МБ.	Файли завантажуються на сервер. Після закінчення результат приходить на пошту.	Так .
4	Перевірити за вантаження файлів більше дозволеного розміру .	Користувач завантажує фото більше 5МБ та відео більше 50МБ.	Файли не завантажуються на сервер, виводиться інформуюче повідомлення .	Так .
5	Перевірити завантаження файлів дозволених форматів.	Користувач завантажує фото та відео файли допустимих форматів.	Файли завантажуються на сервер. Після закінчення результат приходить на пошту.	Так .
6	Перевірити завантаження файлів невірних форматів.	Користувач завантажує відмінні від допустимих форматів файли.	Файли не завантажуються на сервер, виводиться інформуюче повідомлення .	Так .
7	Перевірити від редагування порожніх форм.	Користувач не заповнює поля і натискає на кнопку «Запустити перенесення обличь».	Файли не завантажуються на сервер, виводиться інформуюче повідомлення .	Так .
8	Перевірити відправку коректно заповнених форм.	Користувач заповнює поля та натискає на кнопку «Запустити перенесення обличь».	Файли завантажуються на сервер, виводиться інформуюче повідомлення. Після закінчення результат приходить на пошту .	Так .
9	Перевірити від редагування некоректно-веденої БшаП-адреси.	Користувач вводить ЄшаП-адресу, що не відповідає загальному вигляду ел. адрес та натискає на кнопку «Запустити перенесення обличь».	Виводиться повідомлення про те, що адреса введена некоректно.	Так .

Таблиця Б2 – Результат тестування на адаптивність

№	Ширина екрану (пікселі)	Чи сайт відображається коректно?
1	Понад 1200	Так
2	Від 992 до 1200	Так
3	Від 768 до 992	Так
4	Від 575 до 768	Так
5	Від 340 до 575	Так
6	Менш 340	Так

Таблиця Б3 – Результат тестування на відображення у різних браузерях

№	Назва браузера	Версія	Сайт відображається коректно?
1	Opera [40]	75.0.3969.171	Так
2	Google Chrome [20]	101.0.4951.67	Так
3	Mozilla [35]	100.0.1	Так
4	Microsoft Edge [33]	101.0.1210.39	Так