

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
МІНІСТЕРСТВА ОСВІТИ І НАУКИ УКРАЇНИ
Кафедра комп'ютерних інтелектуальних систем та мереж

ЛОБОДЮК Олексій Костянтинович

ДИПЛОМНА РОБОТА МАГІСТРА

ІНТЕЛЕКТУАЛЬНА ІНФОРМАЦІЙНА СИСТЕМА
РОЗПІЗНАВАННЯ ВІДОМИХ ОСОБИСТОСТЕЙ ДЛЯ ПОШУКОВОЇ
СИСТЕМИ, ЩО САМОНАВЧАЄТЬСЯ

Спеціальність 123 – Комп'ютерна інженерія
Спеціалізація - Комп'ютерні системи та мережі

Керівник: Копитчук Ігор Миколайович,
к.т.н., доцент

Одеса – 2021

АННОТАЦІЯ

Лободюк Олексій Костянтинович. Інтелектуальна інформаційна система розпізнавання відомих особистостей для пошукової системи, що самонавчається – **Дипломна робота магістра. Одеса, 2021.: 74с. , 25 рис., 20 джерел.**

У даній роботі проведено дослідження на тему систем розпізнавання особистості та виходячи з аналізу інформації, отриманої в ході проведення теоретичного дослідження, було розглянуто найбільш ефективний на сьогодні метод застосовуваний у створенні систем розпізнавання особистості на зображеннях. На основі цього методу було запропоновано оригінальний алгоритм побудови системи розпізнавання відомих для системи особистостей з можливістю так званого самонавчання, а також можливістю масштабування та інтеграції надалі в будь-яких пошукових системах під різні потреби, пов'язані з ідентифікацією та осіб.

Запропонований оригінальний алгоритм автоматизованого розпізнавання обличчя на фотографіях, дозволяє суттєво розширити розв'язання задачі ідентифікації. Використання запропонованого алгоритму дозволяє структурувати інформацію при обробці зображень обличчя та здійснювати пошук та групування фотографій заданої людини.

СИСТЕМА ІДЕНТИФІКАЦІЇ ОСОБИСТОСТІ, НЕЙРОННА МЕРЕЖА, ВІДБИТОК ОБЛИЧЧЯ, ВУЗЛОВІ ТОЧКИ ОБЛИЧЧЯ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, БІОМЕТРИЧНІ ТЕХНОЛОГІЇ.

ANNOTATION

Lobodiuk Oleksii Konstantinovich. Self-learning intelligent information system for recognizing famous people for a search engine. - **Master's thesis. Odessa, 2021.: 74p., 25 figs., 20 sources.**

This paper conducted research on personality recognition systems and based on the analysis of information obtained during the theoretical study, considered the most effective method currently used in the creation of personality recognition systems in images. Based on this method, an original algorithm was proposed to build a system of recognition of famous individuals for the system with the possibility of so-called self-learning, as well as the ability to scale and integrate further in any search engine for different needs related to identification and identity.

The proposed original algorithm for automated facial recognition in photographs, allows to significantly expand the solution of the identification problem. Using the proposed algorithm allows you to structure information when processing facial images and search and group photos of a given person.

PERSONALITY IDENTIFICATION SYSTEM, NEURAL NETWORK, FACE IMPRESSION, FACIAL POINTS OF FACE, FACE RECOGNITION, BIOMETHYRICHIN.

ЗМІСТ

Перелік умовних скорочень	5
Вступ	6
1. Аналітичний огляд поставленої задачі	10
1.1. Переваги та недоліки систем розпізнавання особистостей	10
1.2. Приклади застосування технології.....	12
1.3. Аналіз та порівняння існуючих рішень науково-технічної задачі.....	15
1.4. Нейронна мережа як одна з ключових складових системи	18
1.5. Обґрунтування необхідності вдосконалення існуючого об'єкту	23
1.6. Обґрунтування основних прийнятих рішень та напрямів дослідження	25
1.7. Висновки аналітичного огляду	25
2. Завдання на дослідження.....	26
2.1. Призначення	26
2.2. Вхідні дані.....	27
2.3. Вихідні дані.....	27
2.4. Вимоги до системи.....	28
2.5. Висновки завдання	29
3. Теоретичне дослідження	31
3.1. Оцінка ефективності існуючих методів розпізнавання.....	31
3.2. Аналіз структури згорткової нейронної мережі.....	32
3.3. Аналіз результатів та визначення найбільш ефективного методу	35
3.4. Пропозиції щодо покращення існуючих підходів	35
3.5. Структурне побудування програмної моделі	36
3.6. Визначення функції активації та архітектури мережі	37
3.7. Визначення орієнтирів обличчя.....	40

3.8. Режими функціонування програмної моделі.....	41
3.9. Структурні схеми алгоритмів програмної моделі.....	42
3.10. Висновки теоретичного дослідження	46
4. Результати проведених досліджень	47
4.1. Установки програмної моделі	47
4.2. Моделювання та його результати	50
4.3. Висновки моделювання	56
Загальні висновки.....	57
Перелік використаних джерел	59
Додаток А. Код програмної моделі	61

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

СММ – приховані Марківські моделі

РСА – метод головних компонент

ААМ – активні моделі зовнішнього вигляду

ASM – модель активної форми

ІНС – штучні нейронні мережі

MLP – багат шарові перцептрони

RNN – рекурентні нейронні мережі

CNN – згорткові нейронні мережі

САР – шлях присвоєння кредитів

ВСТУП

Багато напрямів науки, техніки та виробництва значною мірою орієнтуються на розвиток систем, у яких інформація носить характер поля. При обробці такої інформації виникає низка складних наукових, технічних та технологічних проблем. Однією із найскладніших завдань є обробка та розпізнавання зображень. Про важливість цієї проблеми говорить той факт, що дослідження з розпізнавання образів, аналізу зображень та мови включені до переліку пріоритетних напрямків розвитку науки і техніки.

На сьогодні широке поширення набувають біометричні системи ідентифікації людини. Традиційні системи ідентифікації вимагають знання пароля, наявності ключа, ідентифікаційної картки чи іншого ідентифікуючого предмета, який можна забути чи втратити. На відміну від них, біометричні системи ґрунтуються на унікальних біологічних характеристиках людини, які важко підробити та які однозначно визначають конкретну людину.

Розпізнавання особистості – це одна з таких біометричних технологій, це спосіб ідентифікації або підтвердження особистості людини за її обличчям. Системи розпізнавання обличчя можна використовувати для ідентифікації людей на фотографіях, відео або в режимі реального часу. Розпізнавання обличчя - це категорія біометричної безпеки. Інші форми біометричного програмного забезпечення включають розпізнавання голосу, розпізнавання відбитків пальців та розпізнавання сітківки ока або райдужної оболонки ока.

Технології розпізнавання осіб у тому чи іншому вигляді розвиваються вже досить давно, проте за останній приблизно десяток років або близько того стався суттєвий стрибок у галузі розробки та навчання нейронних мереж. Цей напрямок вже є одним з найбільш актуальних та перспективних, поряд з розвитком технологій передачі інформації, різноманітних хмарних сервісів та осмисленого аналізу великих обсягів даних. Сьогодні можна вже з упевненістю сказати, що в деякі системи на основі нейронних мереж та розпізнавання обличчя дуже ефективні, але й вони не позбавлені своїх недоліків.

Об'єктом дослідження є біометричні системи розпізнавання особистостей, а точніше їх підрозділ у вигляді систем розпізнавання обличчя.

Предметом дослідження є методика розробки найбільш ефективних рішень для побудови систем розпізнавання особистостей.

Метою дослідження є виявлення принципів роботи подібних систем, сфер можливого використання, а також недоліки цієї технології, що існують на даний момент, оцінка існуючих рішень та розробка найбільш ефективного методу побудови інтелектуальної інформаційної системи розпізнавання відомих особистостей, здатної до самонавчання.

Підставою щодо дослідження є різні наукові праці та розробки у технології систем розпізнавання особистості на зображеннях. Існуючі методики аналізу зображень, виявлення осіб, а також розпізнавання та запам'ятовування особистості.

Вихідними даними цієї науково-дослідної роботи є різні наукові праці і довідкова і статистична інформація з офіційних сайтів розглянутих у праці технологій щодо основної бази побудови систем розпізнавання особистостей.

Задачі дослідження, які вирішуються для досягнення поставленої мети – це аналітичний огляд теми, виявлення існуючих рішень, переваг та недоліків існуючих систем, виявлення принципу роботи основних складових системи, аналіз результатів та виявлення найбільш ефективного способу побудови подібних систем, а також розробка найбільш ефективного методу побудови інтелектуальної інформаційної системи розпізнавання особистостей, з можливістю самонавчання для пошукових систем.

Дослідження включало в себе збір, обробку та аналіз різної інформації, вивчення актуальності технології, її переваг і недоліків, сфер застосування, а також існуючих рішень. Проводилося глибоке вивчення принципів роботи технології та вивчення принципів побудови її архітектури. Розроблялась методика побудови подібної системи, а також можливість уникнути поширених помилок, що зустрічаються у подібних системах. В рамках цього дослідження було розроблено систему розпізнавання відомих особистостей на зображеннях, здатну до

самонавчання з можливістю масштабування системи для подальшого використання у будь-яких пошукових системах.

У першому розділі цієї дипломної роботи проведено аналітичний огляд наукових робіт, та розкривається актуальність та доцільності обраної теми, розглядаються існуючі рішення та розробки, а також розглядаються їх переваги та недоліки, а також містяться обґрунтування необхідності удосконалення існуючого об'єкта на основі аналізу сучасного стану проблеми та досвіду провідних фірм у відповідній галузі.

У другому розділі визначається завдання на дослідження, призначення, вхідні та вихідні дані, потреби до запропонованого методу.

У третьому розділі було проведено оцінку ефективності існуючих методів, а також запропоновані пропозиції щодо вдосконалення ефективності та оптимізації систем розпізнавання. Зроблено структурне побудування програмної моделі системи, визначення основних складових. Визначені основні режими функціонування програмної моделі, а також приведені структурні схеми алгоритмів.

Четвертий розділ присвячено результатам моделювання, визначення установок програмної моделі, її користувальницького інтерфейсу, а також проведено успішне моделювання системи та продемонстровані його результати.

Важливим фактором дослідження даної тематики є те, що системи розпізнавання осіб з кожним днем все сильніше впроваджуються в наше життя, все частіше дуже багато технологій, компаній, структур використовують подібні системи для безпеки та зручності. Це поширене і стає все більш популярним. Наукові спільноти розширюють обговорення за допомогою нових досліджень, включаючи вплив раси, віку та статі на системи розпізнавання осіб, а також вплив масок на обличчя. Подальше вивчення технології призведе до більш проінформованого обговорення її потенційної цінності.

До елементів наукової новизни варто віднести визначення найбільш ефективних рішень для побудови систем розпізнавання особистості на основі технології розпізнавання обличчя, розробку оригінального методу побудови систем

розпізнавання особистостей, виявлення ймовірності помилок у роботі системи, а також виконання порівняльної оцінки достовірності розпізнавання особистостей.

Практична спрямованість та значення проведених досліджень полягає в оцінці реальних можливостей методів та засобів побудови систем розпізнавання особистостей на основі біометричних систем розпізнавання обличь, а також у запропонованій методиці розробки системи розпізнавання та можливостях її реалізації.

Системам розпізнавання осіб ще належить стати для людей буденністю на кшталт інтернету, хоча перші експерименти в обох областях почалися майже паралельно. З часу, коли американцю Бледсо доводилося лінійкою заміряти обличчя, до появи розумних систем розпізнавання минуло чимало часу — 60 років. Людям ще потрібно було винайти потужні комп'ютери та швидкісні канали зв'язку, щоб ця технологія запрацювала. Але коли розумні камери стали реальністю, людство в короткий термін перетворило їх на інструмент, здатний вирішувати багато насущних проблем.

1 АНАЛІТИЧНИЙ ОГЛЯД ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Переваги та недоліки систем розпізнавання особистостей

Крім розблокування смартфона, розпізнавання обличь дає й інші переваги. На рівні уряду розпізнавання осіб може допомогти ідентифікувати терористів або інших злочинців. На особистому рівні розпізнавання осіб можна використовувати як інструмент безпеки для блокування особистих пристроїв та особистих камер спостереження.

Розпізнавання осіб спрощує пошук грабіжників, злодіїв та зловмисників. Лише знання наявності системи розпізнавання осіб може бути стримуючим чинником, особливо у разі дрібних злочинів. Крім фізичної безпеки, є переваги кібербезпеки. Компанії можуть використовувати технологію розпізнавання обличь замість паролів для доступу до комп'ютерів. Теоретично цю технологію неможливо зламати, тому що нічого вкрасти або змінити, як у випадку з паролем.

Занепокоєння громадськості щодо необґрунтованих зупинок та обшуків є джерелом протиріч для поліції – технологія розпізнавання осіб могла б покращити цей процес. Виявляючи підозрюваних серед натовпу за допомогою автоматизованого, а не людського процесу, технологія розпізнавання осіб може допомогти знизити потенційну упередженість та скоротити кількість зупинок та пошуків серед законослухняних громадян.

У міру того, як технологія стає все більш поширеною, покупці зможуть розплачуватися в магазинах обличчям, а не виймати свої кредитні картки або готівку. Це може заощадити час у черзі до каси. Оскільки для розпізнавання обличь не потрібний контакт, як при знятті відбитків пальців або інших заходів безпеки, розпізнавання обличь пропонує швидку, автоматичну та безпроблемну перевірку.

Процес розпізнавання особи займає лише секунду, що дає переваги компаніям, які використовують розпізнавання обличь. В епоху кібератак і передових інструментів хакерів компаніям потрібні як безпечні, так і швидкі

технології. Розпізнавання обличь дозволяє швидко та ефективно перевірити особистість людини.

Більшість рішень для розпізнавання обличчя сумісні з більшістю програм безпеки. Фактично він легко інтегрується. Це обмежує обсяг додаткових інвестицій, необхідних для його реалізації.

Така технологія також має низку досить істотних недоліків. Деякі побоюються, що використання розпізнавання обличь разом із повсюдними відеокамерами, штучним інтелектом та аналізом даних створює потенціал для масового спостереження, що може обмежити особисту свободу.

Дані розпізнавання осіб не вільні від помилок, які можуть призвести до звинувачення людей у злочинах, яких вони не робили. Наприклад, невелика зміна ракурсу камери або зміна зовнішнього вигляду, наприклад нова зачіска, можуть призвести до помилки. У 2018 році технологія розпізнавання обличь Amazon хибно ідентифікувала 28 членів Конгресу США як людей, заарештованих за злочини.

Питання етики та конфіденційності – найспішніше. Відомо, що уряди зберігають фотографії кількох громадян без їхньої згоди. У 2020 році Європейська комісія заявила, що розглядає питання щодо заборони технології розпізнавання обличь у громадських місцях на строк до п'яти років, щоб дати час на розробку нормативної бази для запобігання порушенням конфіденційності та етичних норм.

Програмне забезпечення для розпізнавання осіб ґрунтується на технології машинного навчання, яка потребує величезних наборів даних для «навчання» та отримання точних результатів. Такі великі набори даних потребують надійного сховища даних. Малі та середні компанії можуть не мати достатніх ресурсів для зберігання необхідних даних.

Проблеми з розпізнаванням обличь є причинами, які знижують точність і швидкість розпізнавання обличь. Цими проблемами є складний фон, занадто багато обличь на зображеннях, дивні вирази, освітлення, менша роздільна здатність, закриття обличчя, колір шкіри, відстань та орієнтація. Суть цих проблем у наступному:

- Незвичайний вираз. Людські особи на зображенні можуть мати несподівані чи дивні вирази обличчя.
- Освітлення. Деякі частини зображення можуть мати дуже сильне або слабке освітлення або тіні.
- Типи шкіри. Розпізнавання обличь різного кольору утруднене і потребує ширшої різноманітності навчальних зображень.
- Відстань. Якщо відстань до камери занадто велика, розмір об'єкта може бути занадто малим.
- Орієнтація. Орієнтація обличчя та кут огляду камери впливають на швидкість розпізнавання обличчя.
- Складний фон. Велика кількість об'єктів у сцені знижує точність та швидкість виявлення.
- Багато людей на одному зображенні. Зображення з великою кількістю людей дуже складно для точного виявлення.
- Приховані частини обличчя. Обличчя можуть бути частково приховані такими предметами, як окуляри, шарфи, руки, волосся, капелюхи та інші предмети, що впливає швидкість виявлення.
- Низька роздільна здатність. Зображення з низькою роздільною здатністю або шум зображення негативно впливають на швидкість виявлення.

1.2. Приклади застосування технології

Існує чимало способів застосування технології розпізнавання обличь у всьому світі, ось деякі з них:

- Стеження за натовпом. Розпізнавання обличь використовується для виявлення натовпів у громадських або приватних місцях із частим відвідуванням.

- Взаємодія людини із комп'ютером. Множинні системи, засновані на взаємодії людини та комп'ютера, використовують розпізнавання обличчя для виявлення людей.
- Фотографія. Деякі сучасні цифрові камери використовують функцію розпізнавання обличчя для автофокусування. Мобільні програми використовують розпізнавання обличчя для визначення сфер інтересу в слайд-шоу.
- Виділення рис обличчя. З зображень можна отримати такі риси обличчя, як ніс, очі, рот, колір шкіри тощо.
- Гендерна класифікація. Програми для визначення гендерної інформації створені з допомогою методів розпізнавання обличчя.
- Розпізнавання особистості. Система розпізнавання обличчя призначена для ідентифікації та верифікації людини за цифровим зображенням або відеокадром.
- Маркетинг. Розпізнавання обличчя стає дедалі важливішим для маркетингу, аналізу поведінки клієнтів чи цільової реклами.
- Відвідуваність. Розпізнавання обличчя використовується для виявлення присутності людей. Це часто поєднується з біометричним виявленням для керування доступом

Існує чимало прикладів технологій розпізнавання обличчя. Amazon раніше просуvala свій хмарний сервіс розпізнавання обличчя під назвою Rekognition для правоохоронних органів. Однак у червневому повідомленні у блозі 2020 року компанія оголосила, що планує запровадити річний мораторій на використання своєї технології поліцією. Причина цього у тому, щоб дати час на прийняття законів із захисту правами людини і громадянських свобод.

Apple використовує розпізнавання осіб, щоб допомогти користувачам швидко розблокувати свої телефони, увійти до програм та здійснювати покупки.

British Airways дозволяє розпізнавати особи пасажирів, що вилітають на рейси. Обличчя мандрівників можуть бути відскановані камерою для перевірки їхньої особи при посадці в літак без пред'явлення паспорта або посадкового талону.

Coca-Cola використовувала розпізнавання обличь по-різному у всьому світі. Приклади включають винагороду клієнтів за переробку в деяких торгових автоматах у Китаї, розміщення персоналізованої реклами на торгових автоматах в Австралії та маркетинг заходів в Ізраїлі.

Facebook почав використовувати розпізнавання обличь у 2010 році, коли він автоматично помічав людей на фотографіях за допомогою свого інструменту пропозиції тегів. Інструмент сканує обличчя користувача та пропонує варіанти того, хто ця людина. З 2019 року Facebook включив цю функцію як частину прагнення стати більш орієнтованим на конфіденційність.

Google включає цю технологію в Google Фото та використовує її для сортування зображень та автоматичної позначки їх залежно від того, кого впізнають.

MAC make-up використовує технологію розпізнавання осіб у деяких своїх звичайних магазинах, що дозволяє покупцям віртуально «приміряти» макіяж, використовуючи дзеркала, що є в магазині, з доповненою реальністю.

McDonald's використовує розпізнавання обличчя у своїх японських ресторанах для оцінки якості обслуговування клієнтів, у тому числі для аналізу того, чи усміхаються їй співробітники, допомагаючи клієнтам.

Snapchat - один з піонерів програмного забезпечення для розпізнавання обличь, він дозволяє брендам та організаціям створювати фільтри, які формують обличчя користувача - звідси усюдисущі фільтри з мордочками цуценят та квітковими коронами, які можна побачити у соціальних мережах.

Найсерйознішими недоліками всіх подібних систем є помилки і хибні спрацьовування. Виявлення обличь на фотографії легко вирішується людьми, хоча історично це було складним завданням для комп'ютерів через динамічну природу обличь. Наприклад, обличчя повинно бути виявлено незалежно від орієнтації або кута, під яким воно знаходиться, рівня освітленості, одягу, аксесуарів, кольору волосся, рослинності на обличчі, макіяжу, віку та іншого.

Людське обличчя є динамічним об'єктом і має високий ступінь варіабельності зовнішнього вигляду, що робить виявлення обличчя складною проблемою для комп'ютерного зору.

1.3. Аналіз та порівняння існуючих рішень науково-технічної задачі

Незважаючи на велику різноманітність існуючих алгоритмів, можна виділити загальну структуру процесу розпізнавання обличчя. На першому етапі проводиться детектування та локалізація обличчя на зображенні. На етапі розпізнавання робляться вирівнювання зображення обличчя, яскравості, геометрії, обчислення ознак і розпізнавання – порівняння обчислених ознак із закладеними основою даних стандартами.

Один з існуючих методів це метод гнучкого порівняння на графах, суть методу зводиться до еластичного зіставлення графів, що описують зображення обличчя. Обличчя представлені у вигляді графів зі зваженими вершинами та ребрами. На етапі розпізнавання один із графів – еталонний – залишається незмінним, тоді як інший деформується з метою найкращого припасування до першого.

У вершинах графа обчислюються значення ознак, найчастіше використовують комплексні значення фільтрів Габора чи його впорядкованих наборів Габорівських вейвлет, які обчислюються у деякій області вершини графа шляхом згортки значень яскравості пікселів з фільтрами Габора.

Ребра графа зважуються відстанями між суміжними вершинами. Відмінність, яка полягає у відстані та дискримінаційній характеристиці, між двома графами обчислюється за допомогою певної цінової функції деформації, яка враховує як різницю між значеннями ознак, обчисленими у вершинах, так і ступінь деформації ребер графа.

Деформація графа відбувається шляхом зміщення кожної з його вершин на деяку відстань у певних напрямках щодо її вихідного розташування та вибору такої її позиції, при якій різниця між значеннями ознак, відгуків фільтрів Габора, у

вершині графа, що деформується, і відповідної їй вершині еталонного графа буде мінімальною. Ця операція виконується послідовно для всіх вершин графа до того часу, поки не буде досягнуто найменша сумарна різниця між ознаками деформуемого та еталонного графів. Значення цінової функції деформації при такому положенні графа, що деформується, і буде мірою відмінності між вхідним зображенням обличчя та еталонним графом. Ця процедура деформації має виконуватися усім еталонним обличчям, закладеним у базу даних системи. Результатом розпізнавання системи буде зразок із найкращим значенням цінової функції деформації.

Одним із статистичних методів розпізнавання обличчя є приховані Марківські моделі з дискретним часом. Приховані Марківські моделі (СММ) використовують статистичні властивості сигналів та враховують безпосередньо їх просторові характеристики. Елементами моделі є безліч прихованих станів, безліч станів, що спостерігаються, матриця перехідних ймовірностей, початкова ймовірність станів. Кожному з цих елементів відповідає своя Марківська модель.

При розпізнаванні об'єкта перевіряються згенеровані для заданої бази об'єктів Марківські моделі і шукається максимальна ймовірність того, що послідовність спостережень для даного об'єкта згенерована відповідною моделлю.

Одним з найбільш відомих і опрацьованих є метод головних компонентів (principal component analysis, PCA), заснований на перетворенні Карунена-Лоева.

Головною метою методу є значне зменшення розмірності у просторі ознак таким чином, щоб воно якнайкраще описувало «типові» образи, що належать безлічі обличчя. Використовуючи цей метод, можна виявити різні мінливості в навчальній вибірці зображень обличчя і описати цю мінливість у базисі кількох ортогональних векторів, які прийнято називати власними.

Отриманий один раз на навчальній вибірці зображень обличчя набір власних векторів використовується для кодування решти зображень обличчя, які у результаті видаються зваженою комбінацією цих власних векторів.

Активні моделі зовнішнього вигляду (Active Appearance Models, ААМ) - це ще один метод, який застосовується для розпізнавання. Статистичні моделі

зображень, які шляхом різноманітних деформацій можуть бути підігнані під реальне зображення.

Активна модель зовнішнього вигляду містить два типи параметрів: параметри, пов'язані з формою, та параметри, пов'язані зі статистичною моделлю пікселів зображення або текстурою. Перед використанням модель має бути навчена на безлічі заздалегідь розмічених зображень. Розмітка зображень виконується вручну. Кожна мітка має свій номер та визначає характерну точку, яку повинна знаходити модель під час адаптації до нового зображення.

Процедура навчання ААМ починається з нормалізації форм на розмічених зображеннях з метою компенсації відмінностей у масштабі, нахилі та зміщенні. З усієї множини нормованих точок потім виділяються головні компоненти з використанням методу РСА.

Далі з пікселів усередині трикутників, що утворюються точками форми, формується матриця, кожен її стовпець містить значення пікселів відповідної текстури. Після знаходження основних компонентів матриці текстур модель ААМ вважається навченою.

При вирішенні задачі локалізації обличчя на зображенні виконується пошук параметрів ААМ, під якими мається на увазі розташування, форма, текстура, які представляють зображення, що синтезується, найбільш близьке до спостережуваного. За ступенем близькості ААМ зображенню, що підганяється, приймається рішення - є обличчя на зображенні чи ні.

Модель активної форми (Active Shape Models (ASM)) діє схожим чином. Суть методу ASM полягає в обліку статистичних зв'язків між розташуванням антропометричних точок. На вибірці зображень обличчя, знятих в анфас. На зображенні експерт розмічає розташування антропометричних точок. На кожному зображенні точки пронумеровані однаково.

Для того, щоб привести координати на всіх зображеннях до єдиної системи, зазвичай виконується так званий узагальнений прокрустовий аналіз, в результаті якого всі точки наводяться до одного масштабу та центруються. Далі всього набору образів обчислюється середня форма і матриця коваріації. На основі матриці

коваріації обчислюються власні вектори, які потім сортуються в порядку зменшення відповідних їм власних значень.

1.4. Нейронна мережа як одна з ключових складових системи

Нейронні мережі є базою для створення системи розпізнавання особистості. Нейронні мережі відображають поведінку людського мозку, дозволяючи комп'ютерним програмам розпізнавати закономірності та вирішувати спільні проблеми в галузі штучного інтелекту, машинного навчання та глибокого навчання.

Штучні нейронні мережі (ІНС) складаються з шарів та вузлів, містять вхідний шар, один або кілька прихованих шарів та вихідний шар. Кожен вузол або штучний нейрон з'єднується з іншим і має пов'язану вагу та поріг. Якщо вихідний сигнал будь-якого окремого вузла перевищує задане граничне значення, цей вузол активується, надсилаючи дані на наступний рівень мережі, а якщо ні, то жодні дані не передаються на наступний рівень мережі.

Нейронні мережі покладаються на навчальні дані, щоб навчатися та з часом підвищувати свою точність. Однак після точного налаштування цих алгоритмів навчання вони стають потужними інструментами в галузі інформатики та штучного інтелекту, дозволяючи нам класифікувати та кластеризувати дані з високою швидкістю.

Кожен окремих шар являє собою модель лінійної регресії, що складається з вхідних даних, ваг, порога і вихідних даних. Формула виглядає приблизно ось так:

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + bias \quad (1.1)$$

де w_i – ваги;

x_i – вхідні данні;

m - кількість відликів;

$bias$ – зміщення.

Після визначення вхідного шару призначаються ваги. Ці ваги допомагають визначити важливість будь-якої даної змінної, при цьому більші з них роблять більш значний внесок у результат порівняно з іншими вхідними даними. Потім всі вхідні дані множаться на відповідні ваги і підсумовуються. Після цього результат проходить через функцію активації, яка визначає висновок. Якщо цей вихідний сигнал перевищує заданий поріг, він активує вузол, передаючи дані на наступний рівень мережі. Це призводить до того, що вихід одного вузла стає входом наступного вузла. Цей процес передачі від одного рівня до наступного визначає цю нейронну мережу як мережу прямого поширення. Формула вихідних даних виглядає наступним чином:

$$output = f(x) = \begin{cases} 1 \text{ if } \sum w_1 x_1 + b \geq 0 \\ 0 \text{ if } \sum w_1 x_1 + b < 0 \end{cases} \quad (1.2)$$

де $f(x)$ – функція активації;

$output$ – вихідні данні;

x_1 – вхідні данні;

w_1 - ваги;

b – зміщення.

Більш практичні варіанти використання нейронних мереж, таких як розпізнавання або класифікація зображень, використовують контрольоване навчання або розмічені набори даних для навчання алгоритму. Під час навчання моделі для того, щоб оцінити точність використовують функцію втрат, у вигляді середньоквадратичної помилки:

$$Cost Function = MSE = \frac{1}{2m} \sum_{i=1}^m (\hat{y} - y)^2 \quad (1.3)$$

де *Cost Function* – функція втрат;

output – вихідні данні;

MSE – середньоквадратична помилка;

m - кількість відликів;

\hat{y} – передбачений результат;

y – фактичне значення;

i – індекс вибірки.

У міру того, як модель коригує свої ваги та зсув, вона використовує функцію втрат та навчання з підкріпленням, щоб досягти точки збіжності або локального мінімуму. Процес, у якому алгоритм коригує свої ваги, здійснюється шляхом градієнтного спуску, що дозволяє моделі визначати напрямок, у якому слід рухатися, щоб зменшити помилки.

Більшість глибоких нейронних мереж мають прямий зв'язок, тобто вони рухаються лише в одному напрямку, від входу до виходу. Однак можливо навчити модель за допомогою зворотного поширення помилки, тобто рухатися у протилежному напрямку від виведення до введення. Зворотне поширення дозволяє обчислити та приписати помилку, пов'язану з кожним нейроном, що в свою чергу дозволяє відповідним чином налаштувати та підігнати параметри моделі.

Нейронні мережі можна розділити на різні типи, які використовуються для різних цілей, потрібно розглянути ці типи для того, щоб з'ясувати який з них найбільш підходить для побудови систем розпізнавання особистостей.

Персептрон – найстаріша нейронна мережа. Вона складається з одного нейрона і є найпростішою формою нейронної мережі (рис.1.1).

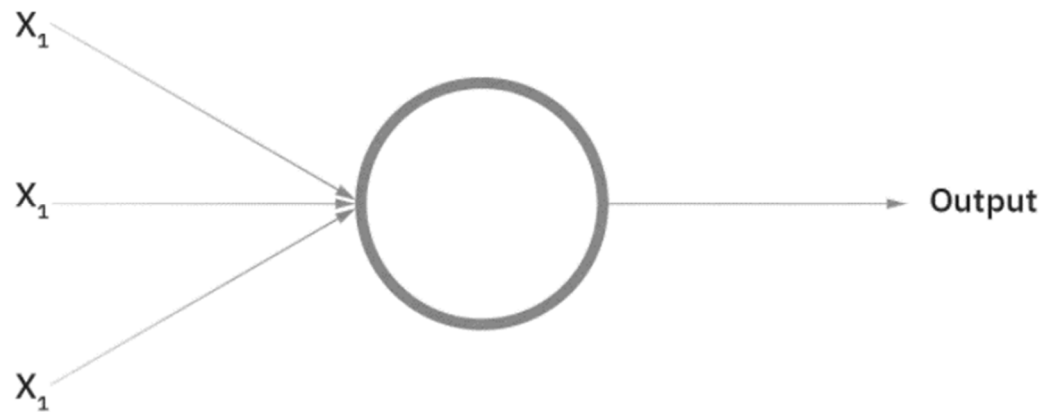


Рисунок 1.1 – Перцептрон

Нейронні мережі з прямим зв'язком або багатошарові перцептрони (MLP) – це класичний основний тип нейронних мереж. Вони складаються з вхідного шару, прихованого шару або шарів та вихідного шару. Структура найпростішого багатошарового перцептрону з одним схованим шаром зображено на рисунку 1.2.

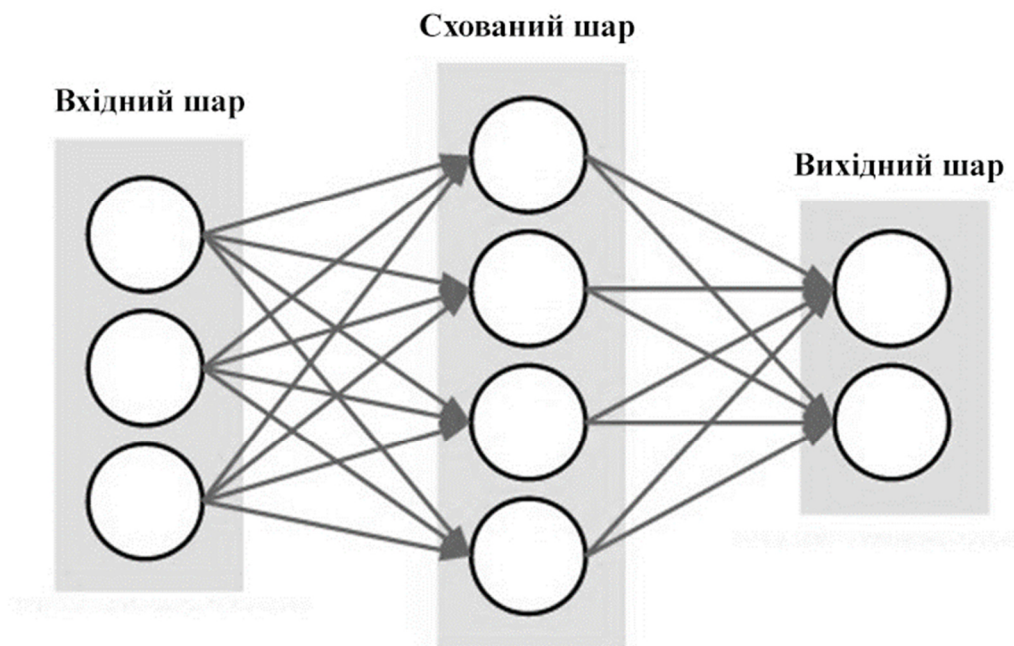


Рисунок 1.2 – Багатошаровий перцептрон

Рекурентні нейронні мережі (RNN) ідентифікуються за їх петлями зворотного зв'язку. Ці алгоритми навчання в першу чергу використовуються при використанні даних часових рядів для прогнозування майбутніх результатів, наприклад, прогнозів фондового ринку або прогнозування продажів.

Ще один тип штучних нейронних мереж – це згорткові нейронні мережі (CNN) схожі на мережі прямого поширення, але зазвичай використовуються для розпізнавання зображень, розпізнавання образів та/або комп'ютерного зору. Ці мережі використовують принципи лінійної алгебри, зокрема, матричне множення, виявлення закономірностей в зображенні.

Штучні нейронні мережі можуть бути як прості, базові, так і складні, глибокі, архітектури глибокого навчання, такі як глибокі нейронні мережі, мережі глибоких переконань, глибоке навчання з підкріпленням, рекурентні нейронні мережі та згорткові нейронні мережі, застосовувалися в таких областях, як комп'ютерний зір, розпізнавання мови, обробка природної мови, машинний переклад. Програми аналізу зображень, перевірки матеріалів та настільних ігор, у яких вони дали результати, порівняні, а в деяких випадках перевершують можливості експертів.

Глибоке навчання пов'язано з необмеженою кількістю шарів обмеженого розміру, що допускає практичне застосування та оптимізовану реалізацію, зберігаючи при цьому теоретичну універсальність у м'яких умовах. У глибокому вивченні шари також можуть бути неоднорідним і сильно відхилитися від біологічно поінформованих моделей, заради ефективності, навчання і зрозумілості.

Глибоке навчання - це клас алгоритмів машинного навчання, який використовує кілька рівнів для поступового отримання високорівневих функцій з необроблених вхідних даних. Наприклад, при обробці зображень нижні рівні можуть ідентифікувати краї, у той час як більш рівні можуть ідентифікувати концепції, що стосуються людини, такі як цифри, літери або обличчя.

Більшість сучасних моделей глибокого навчання засновані на штучних нейронних мережах, зокрема, на згорткових нейронних мережах (CNN), хоча вони також можуть включати пропозиціональні формули або приховані змінні,

організовані на рівні шарів у глибоких генеративних моделях, таких як вузли в глибоких мережах.

У глибокому навчанні кожен рівень вчиться перетворювати свої вхідні дані на дещо абстрактніше і складове уявлення. У додатку для розпізнавання зображень необроблене введення може бути матрицею пікселів, перший репрезентативний шар може абстрагувати пікселі і кодувати краї, другий шар може становити і кодувати конфігурації країв, третій шар може кодувати ніс і очі, а четвертий шар може розпізнати, що на зображенні присутнє обличчя. Різна кількість шарів та їх розміри можуть забезпечувати різний рівень ефективності.

Слово «глибокий» у «глибокому навчанні» належить до кількості рівнів, якими перетворюються дані. Системи глибокого навчання мають значну глибину шляху присвоєння кредитів (САР). САР - це ланцюжок перетворень від введення висновку. САР описують потенційно причинні зв'язки між введенням та виведенням. Для нейронної мережі з прямим зв'язком глибина САР дорівнює глибині мережі і дорівнює кількості прихованих шарів плюс один (оскільки вихідний шар також параметризований). Для рекурентних нейронних мереж, в яких сигнал може поширюватися через шар більше одного разу, глибина САР потенційно не обмежена.

Немає загальновизнаного порога глибини, який відокремлює неглибоке навчання від глибокого, але більшість дослідників згодні з тим, що глибоке навчання передбачає глибину САР вище 20. Було показано, що САР глибини 20 є універсальним наближенням у тому сенсі, що він може імітувати будь-яку функцію. Крім того, більша кількість шарів не збільшує апроксимуючу здатність мережі. Глибокі моделі здатні отримувати кращі функції, ніж дрібні моделі, і, отже, додаткові шари допомагають ефективному вивченні функцій.

1.5. Обґрунтування необхідності вдосконалення існуючого об'єкту

Актуальність і доцільність дослідження полягає в тому, що системи розпізнавання особистостей за допомогою біометричних систем розпізнавання

обличчя все частіше починають використовуватися в повсякденному житті, виходячи з чого потрібен свіжий погляд на побудову цих систем та удосконалення існуючих методів для підвищення ефективності роботи та точності розпізнавання, а також зменшення кількості можливих помилок.

Більшість існуючих методів побудови системи розпізнавання обличчя на зображенні мають ряд істотних недоліків, що робить застосування їх у побудові системи розпізнавання особистостей не досить ефективними, а іноді й зовсім неефективними.

Конкретизація недоліків:

- Метод гнучкого порівняння на графах має високу обчислювальну складність процедури розпізнавання, та низьку технологічність при запам'ятовуванні нових еталонів, а також лінійну залежність часу роботи від обсягу бази даних обличчя.
- Приховані Марківські моделі потребують необхідність підбирати параметри моделі для кожної бази даних окремо, а також не мають здатності розрізняти, тобто алгоритм навчання тільки максимізує відгук кожного зображення на свою модель, але не мінімізує відгук на інші моделі.
- Метод головних компонент вибирає підпростір з такою метою, щоб максимально апроксимувати вхідний набір даних, а не виконати дискримінацію між класами обличчя, що призводить до того, що у тих випадках, коли на зображенні обличчя присутні зміни у освітленості або виразі обличчя, ефективність методу значно падає.
- Головною метою AAM і ASM є не розпізнавання обличчя, а точна локалізація обличчя та антропометричних точок на зображенні для подальшої обробки.

1.6. Обґрунтування основних прийнятих рішень та напрямів дослідження

Більшість існуючих рішень мають суттєві недоліки, які вплинуть на роботу побудованої на їх основі системи, що призведе до помилок, хибних спрацьовувань та/або вимагатиме великих обчислювальних потужностей.

Як основні рішення було прийнято виявити, удосконалити та оптимізувати найбільш ефективний метод для побудови систем розпізнавання особистостей, а також оцінити точність та якість його роботи.

Напрямами проведеного дослідження буде оцінка та виявлення найбільш ефективних та практичних рішень з існуючих, а також описані пропозиції щодо їх удосконалення.

Необхідно, на основі аналізу існуючих рішень, оцінити та дослідити їх точність та ефективність, виявити найбільш якісні, удосконалити та запропонувати оригінальне рішення поставленого завдання.

1.7. Висновки аналітичного огляду

Біометрична технологія розпізнавання особи має досить перспективне майбутнє. Точність і якість розпізнавання в цих системах збільшується з кожним роком, проте переважна кількість цих систем не позбавлені серйозних недоліків.

На етапі аналітичного огляду було розглянуто основні існуючі методи побудови систем розпізнавання обличчя та виявлення їх недоліків.

Було обґрунтовано доцільність та актуальність досліджуваної теми та визначено напрямки дослідження та необхідні задачі.

2 ЗАВДАННЯ НА ДОСЛІДЖЕННЯ

2.1. Призначення

Інтелектуальна інформаційна система розпізнавання відомих особистостей, здатна самонавчатися, призначена для детектування на будь-яких зображеннях обличчя, а також виділення індивідуального для кожної людини відбитка обличчя за допомогою лицьових орієнтирів, що називаються вузловими точками.

Знайдений на зображенні так званий відбиток обличчя перетворюється на числовий код, завдяки якому і відбуватиметься процес розпізнавання особистості. Система розпізнавання обличчя здатна зіставляти людське обличчя на цифровому зображенні зі своєю базою даних обличчя.

Подібна система відноситься до категорії біометричних технологій і є на сьогоднішній день найбільш передовою розробкою в питанні біометричного розпізнавання.

Розроблений метод побудови системи буде можливо використовувати в будь-яких існуючих пошукових системах, або ж розроблених окремо для будь-якого пошуку в базах даних, відкритих джерелах на просторах інтернет мережі, або ж просто для систем біометричної безпеки.

Систему можна буде масштабувати і підлаштувати під будь-яке необхідне завдання.

Розв'язання задачі розпізнавання обличчя на зображенні включає кілька етапів. Передбачається, що попередня обробка зображення, пов'язана з усуненням перешкод, що діють, і спотворень, обумовлених трактом отримання і передачі по каналах зв'язку, виконана.

Висувається гіпотеза, що у зображенні представлена людина з помітними рисами особи. Як правило, зображення представлене на навколишньому фоні, який є не однорідним, тобто предметами інтер'єру або екстер'єру. Перший етап системи розпізнавання обличчя полягає у визначенні локальної області зображення обличчя

людини за її характерними ознаками, якими є колірні складові, локальні особливості точок обличчя та їхнє взаємне розташування, а також форма. Вилучення та оцінка цих ознак лежить в основі цілого класу алгоритмів, спрямоване на вирішення задачі детектування обличчя на статичних зображеннях.

2.2. Вхідні дані

Як вхідні дані в системі будуть використовуватися різні цифрові зображення з обличчями людей, які певним чином оброблятимуться системою.

До того ж у розглянутому прототипі розробленої системи буде демонстраційна можливість пошуку зображень по базі даних.

Якщо розглядати розроблені окремі складові частини системи, що підключаються до неї для повноцінної роботи як вхідні дані, то також можна виділити попередньо навчену модель штучної згорткової залишкової глибокої нейронної мережі, а також попередньо створену модель визначення орієнтирів обличчя н зображеннях за допомогою ключових вузлових точок.

З цієї ж точки зору вхідними даними для розробки можна визначити різні використовувані бібліотеки та методи, необхідні для реалізації поставленого завдання, а також всі необхідні програми та їх компоненти за допомогою яких у рамках дослідження проводитиметься розробка методу побудови нашої системи.

2.3. Вихідні дані

Вихідними даними з точки зору користувача буде інформація про розпізнану особистість, а також набір зображень з розпізнаною особистістю, що зберігається в базі даних. Вихідними даними також буде реалізований графічний додаток що дозволяє за допомогою присутніх у ньому елементів управління проводити пошук і розпізнавання.

З точки зору розробки вихідними даними будуть дескриптори осіб на зображеннях, які будуть порівнюватися між собою для забезпечення розпізнавання особистості.

2.4. Вимоги до системи

Основними вимогами до системи розпізнавання особистостей будуть:

- основна можливість розпізнавання обличчя на зображенні, а також визначення особистості людини на ньому та видача по ньому короткої інформації;
- демонстраційна можливість пошуку інформації по існуючій базі даних з подальшою видачою зображень, що зберігаються в ній, з шуканою особистістю;
- можливість зберігати у свою базу даних зображення притаманні відомої для системи особистості, яких ще немає в базі даних;
- можливість запам'ятовувати невідомі раніше обличчя для подальшого виявлення системою та поповнюючи цим базу відомих системі особистостей;
- демонстраційна можливість так званої верифікації особистості, порівнюючи кілька зображень та визначення чи та сама людина на них зображена;
- однією з важливих вимог також є кросплатформність системи, для можливості її використання на різних пристроях;
- важливою вимогою є можливість масштабування системи та інтеграції її в будь-які інші системи для використання в необхідних потребах;
- забезпечення простоти використання та інтеграції, а також забезпечення можливості редагування без порушення структури системи;
- крім того, одним з найважливіших вимог до системи буде якість і точність розпізнавання.

Розроблений метод побудови системи повинен бути досить актуальним і конкурентоспроможним, а також відрізнятися надійністю і точністю розпізнавання.

2.5. Висновки завдання

У поставленій перед нами задачі ми розроблятимемо метод будівництва інтелектуальної інформаційної системи розпізнавання, розглянемо можливості реалізації даної системи, її складових частин та принципу роботи. На основі зібраної та проаналізованої інформації теоретичного дослідження ми визначимо найбільш ефективні варіанти реалізації, переваги, недоліки, а також існуючі рішення, які використовуються у світі сьогодні.

Виходячи з проаналізованих нами даних ми підберемо рішення, яке максимально ефективно вирішуватиме поставлені перед нами завдання і визначимо метод, що забезпечує найбільшу точність і якість розпізнавання осіб на зображеннях.

Також буде докладно розглядатися низка проблем, помилок та недоліків пов'язаних із системою, а також пояснення їх причин та можливостей їх вирішення, якщо це можливо.

Розроблений метод повинен буде відповідати всім поставленим перед нами вимогам, а також буде універсальною, гнучкою і масштабованою системою, яку можна буде використовувати в реалізації будь-яких потреб пов'язаних з біометричними пошуковими системами або системами безпеки.

Буде здійснено пошук інформації та глибоке вивчення поставленого перед нами питання для виявлення всіх специфічних аспектів роботи подібних систем, завдяки чому буде можливе забезпечення високої якості розробленої системи.

Всі функції та особливості розробленої системи, а також її компоненти та способи реалізації будуть докладно описані та розглянуті.

Необхідно буде глибоко вивчити принцип роботи всіх існуючих можливих методів вирішення поставлених нами завдання та визначити найбільш ефективні складові, що використовуються в розробці. Розглянути та знайти можливі шляхи реалізації та підібрати найкращі. Відобразити основні структури, ідеї та ключові моменти у розробці методу реалізації системи.

В результаті необхідно надати оригінальне рішення автоматизованого розпізнавання обличчя на фотографіях, дозволяючи тим самим суттєво розширити розв'язання задачі ідентифікації. Мати можливість структурувати інформацію при обробці зображень обличчя та здійснювати пошук та групування фотографій заданої людини в розробленій системі.

Необхідно надати програмну модель системи розпізнавання особистостей, її установки, функції та продемонструвати результати її роботи. Виходячи з результатів роботи програмної моделі необхідно зробити висновок вдалося покращити існуючі методи побудови подібних систем чи ні.

3 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

3.1. Оцінка ефективності існуючих методів розпізнавання

Проаналізувавши всі переваги та недоліки існуючих методів, можна оцінити їх ефективність для використання у побудові системи розпізнавання особистості.

Зазначається, що метод гнучкого порівняння на графах має високу ефективність розпізнавання, навіть за наявності різних емоційних виразів та зміни ракурсу, а також має низьку чутливість до рівня освітленості обличчя, що безсумнівно є плюсом. Однак, той факт, що метод гнучкого порівняння на графах має високу обчислювальну складність процедури розпізнавання, зазначається, що для порівняння вхідного зображення особи з 87 еталонними витрачається приблизно 25 секунд, та низьку технологічність при запам'ятовуванні нових еталонів, а також лінійну залежність часу роботи від обсягу бази даних обличь робить його непридатним для використання в побудові систем розпізнавання особистостей, адже основними вимогами до їх роботи є швидкодія та можливість розширення бази даних для будь-яких пошукових цілей.

Метод СММ є непридатним рішенням для побудови систем розпізнавання особистості через свою специфіку у створенні бази даних, завдяки якій для кожної бази даних потрібно окремо підбирати параметри моделі, а це неприйнятно, тому що важливою вимогою побудови системи розпізнавання є легка масштабованість та інтеграція в будь-яку другу систему.

Метод РСА непогано себе зарекомендував у практичних додатках, однак точність розпізнавання – це головна характеристика системи розпізнавання особистостей, а ефективність методу РСА значно падає при зміні освітленості зображення або зміні виразу обличчя, що робить застосування цієї технології недостатньо ефективною для потреб систем розпізнавання особистостей на сьогоднішній день.

Головною метою ААМ і АSM є не розпізнавання обличчя, а точна локалізація обличчя та антропометричних точок на зображенні для подальшої обробки, так що заснувати систему розпізнавання особистості суто на них не є можливим, оскільки система має сама вміти розпізнавати обличчя.

3.2. Аналіз структури згорткової нейронної мережі

Одні з найкращих результатів в області розпізнавання осіб досягається за допомогою використання згорткових нейронних мереж (CNN), які є логічним розвитком таких архітектур як когнітрон і неокогнітрон. Успіх обумовлений можливістю обліку двовимірної топології зображення, на відміну від багат шарового перцептрона.

Згорткові нейронні мережі тепер забезпечують більш масштабований підхід до завдань класифікації зображень та розпізнавання об'єктів, використовуючи принципи лінійної алгебри, зокрема множення матриць, виявлення закономірностей у зображенні. Проте вони можуть вимагати значних обчислювальних ресурсів.

Згорткові нейронні мережі відрізняються від інших нейронних мереж своєю чудовою продуктивністю з вхідними сигналами зображення, мови або звуку.

У згорткових нейронних мереж є три основні типи шарів:

- Згортковий шар
- Рівень об'єднання
- Повнозв'язний шар

Згортковий шар - це перший рівень згорткової мережі. За згортковими шарами можуть йти додаткові згорткові шари або шари об'єднання, повністю пов'язаний шар є останнім шаром. З кожним рівнем складність CNN збільшується, ідентифікуючи великі частини зображення. Більш ранні шари зосереджені на простих елементах, таких як кольори та краї. У міру того, як дані зображення

проходять через рівні CNN, вони починають розпізнавати більші елементи або форми об'єкта, поки, нарешті, не ідентифікують передбачуваний об'єкт.

Згортковий шар є основним будівельним блоком CNN, і саме тут відбувається більша частина обчислень. Для цього потрібні кілька компонентів: вхідні дані, фільтр та карта функцій. Детектор функцій, також відомий як ядро або фільтр, який буде переміщатися по сприймаючих полів зображення, перевіряючи, чи є функція (рис. 3.1).

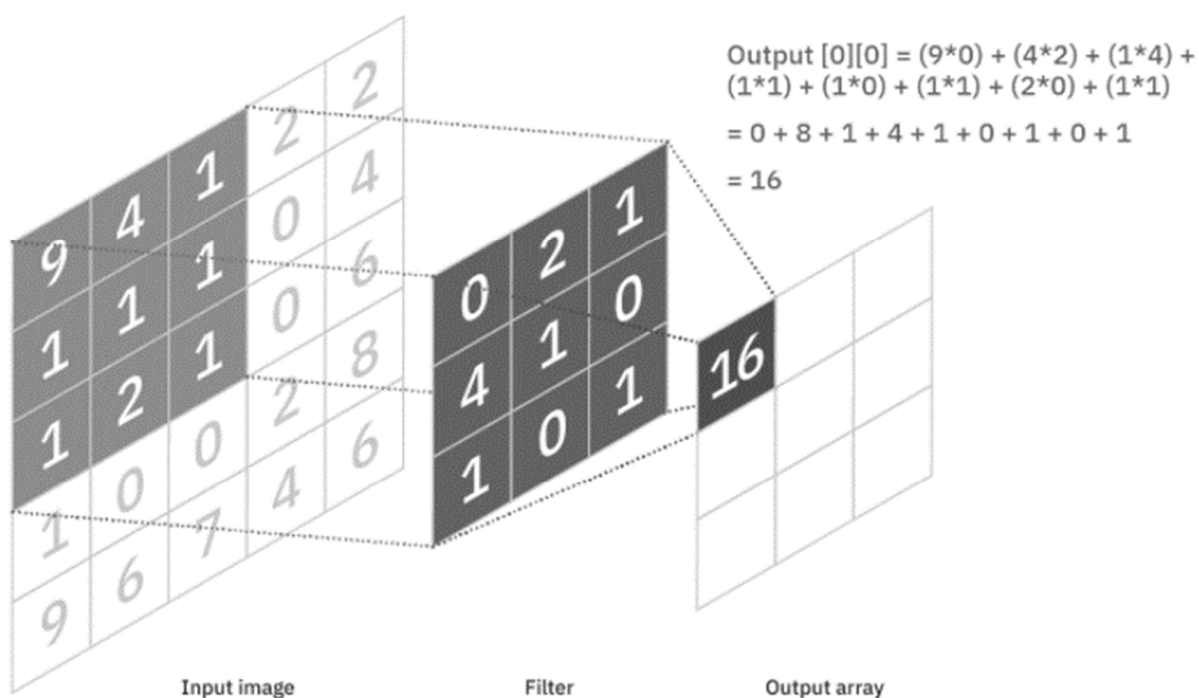


Рисунок 3.1 – Процес згортки

Детектор ознак представляє собою двовимірний масив ваг, який являє собою частину зображення. Розмір фільтра зазвичай є матрицею 3x3, це також визначає розмір рецептивного поля. Фільтр застосовується до області зображення, між вхідними пікселями і фільтром обчислюється скалярне значення. Цей скалярне значення передається у вихідний масив. Фільтр поступово переміщається, повторюючи процес доти, доки ядро не охопить усі зображення. Остаточний

результат серії скалярних значень із входу та фільтра являє собою карту характеристик чи згорнутий елемент.

Ваги в детекторі функцій залишаються фіксованими при переміщенні зображення. Деякі параметри, такі як значення ваги, регулюються в процесі навчання під час зворотного поширення помилки і градієнтного спуску.

Наступний згортковий шар може слідувати за початковим згортковим шаром. Коли це відбувається, структура CNN може стати ієрархічною, оскільки пізніші шари можуть бачити пікселі в межах полів попередніх шарів.

На рівні об'єднання операція об'єднання пропускає фільтр по всьому входу, але цей фільтр не має ваг. Ядро застосовує функцію агрегування до значень у приймаючому полі, заповнюючи вихідний масив. Є два основні типи:

- Максимальне об'єднання: коли фільтр переміщається на вході, він вибирає піксель з максимальним значенням для відправки у вихідний масив.
- Середній пул: у міру того, як фільтр переміщається по вході, він обчислює середнє значення в полі, що приймає, для відправки у вихідний масив.

На рівні об'єднання втрачається багато інформації, але він допомагає знизити складність, підвищити ефективність та знизити ризик перенавчання.

На рівні пов'язного шару значення пікселів вхідного зображення не пов'язані безпосередньо з вихідним шаром у частково пов'язаних шарах, проте у повністю підключеному шарі кожен вузол вихідного шару підключається безпосередньо до вузла попереднього рівня.

Пов'язаний шар виконує завдання класифікації на основі функцій, вилучених за допомогою попередніх шарів, та їх різних фільтрів. Структуру згорткової мережі зображено на рисунку 3.2.

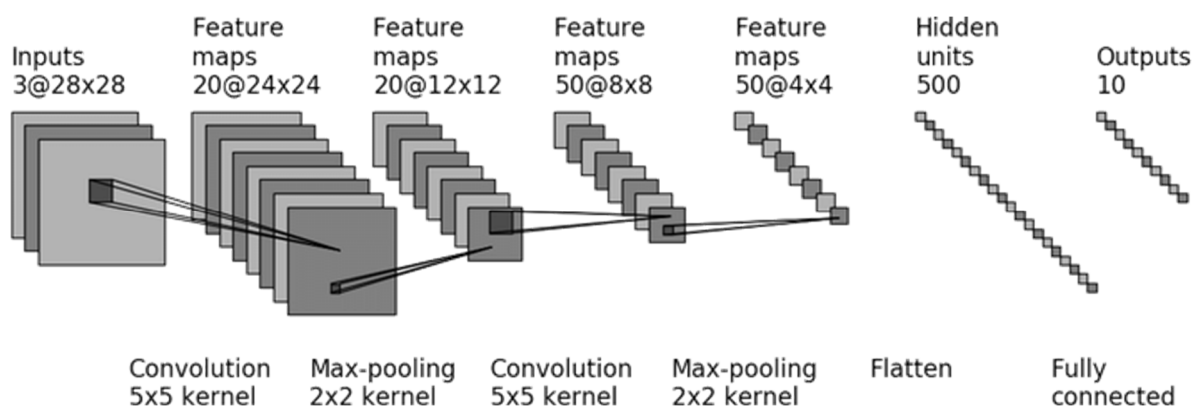


Рисунок 3.2 – Структура згорткової нейронної мережі

3.3. Аналіз результатів та визначення найбільш ефективного методу

Виходячи з отриманих результатів оцінки ефективності методів розпізнавання можна зробити висновки, що не один з розглянутих методів не є достатньо ефективним для побудови системи розпізнавання особистостей. Однак, варто зауважити, що деякі рішення притаманні таким методам як ААМ і АСМ є досить хорошими, якщо використовувати їх як складову для побудови системи.

CNN забезпечує часткову стійкість до змін масштабу, зсувам, поворотам, зміні ракурсу та іншим змінам зображення, але великим недоліком методів, які побудовані на основі нейронних мереж можна зазначити те, що додавання нової особи в базу даних вимагає повного перенавчання мережі на всьому наявному наборі, а це досить тривала процедура, яка залежно від розміру вибірки вимагає дуже багато часу.

3.4. Пропозиції щодо покращення існуючих підходів

Необхідно вдосконалити існуючі підходи, пропозицією покращення є використання згорткової нейронної мережі для розпізнавання особи на зображенні та за допомогою таких методів як ААМ та АСМ локалізувати основні антропологічні вузлові точки на зображенні, що у свою чергу дозволить зчитувати

та зберігати так званий відбиток обличчя певної людини, який також буде служити ключем для його розпізнавання.

Використання подібного поліпшення дозволить отримати всі переваги використання нейронної мережі для розпізнавання обличчя на зображенні та прибере головний недолік у вигляді потреби перенавчання нейронної мережі, для додавання нової особи, оскільки особи будуть закріплюватися за дескрипторами, витягнутими з відбитка обличчя, отриманого за допомогою моделей подібних до AAM і ASM.

3.5. Структурне побудування програмної моделі

Програмна модель за допомогою елементів керування та демонстраційних елементів повинна забезпечувати процес проведення перевірки дієздатності запропонованої системи розпізнавання.

Склад елементів керування визначається функціональністю моделі та включає:

- поля де будуть відображатися обрані зображення та елементи, що дозволяють обрати необхідні зображення;
- поле де буде відображатися інформація про особу розпізнану на зображенні та елементи, що дозволяють виконати функцію розпізнавання;
- поле де буде відображатися інформація стосовно верифікації зображень та елементи, що дозволяють виконати функцію верифікації;
- елементи пошукової строки, та елементи, що дозволяють виконати пошук потрібної особи, та демонстраційні елементи у яких будуть відображатися знайдені зображення.

3.6. Визначення функції активації та архітектури мережі

Існує багато різних архітектур побудови нейронних мереж. Основою для нашої моделі було обрано архітектуру ResNet (рис 3.3). Глибокі нейронні мережі важче навчати. ResNet представляє залишкову структуру навчання, щоб полегшити навчання мереж, які можуть бути значно глибшими. Переформульовані шари як залишкові функції з посиланням на вхідні дані є основою цієї архітектури. Вони надають вичерпні емпіричні дані, що показують, що ці залишкові мережі легше оптимізувати, і вони можуть підвищити точність за можливості значно збільшити глибину.

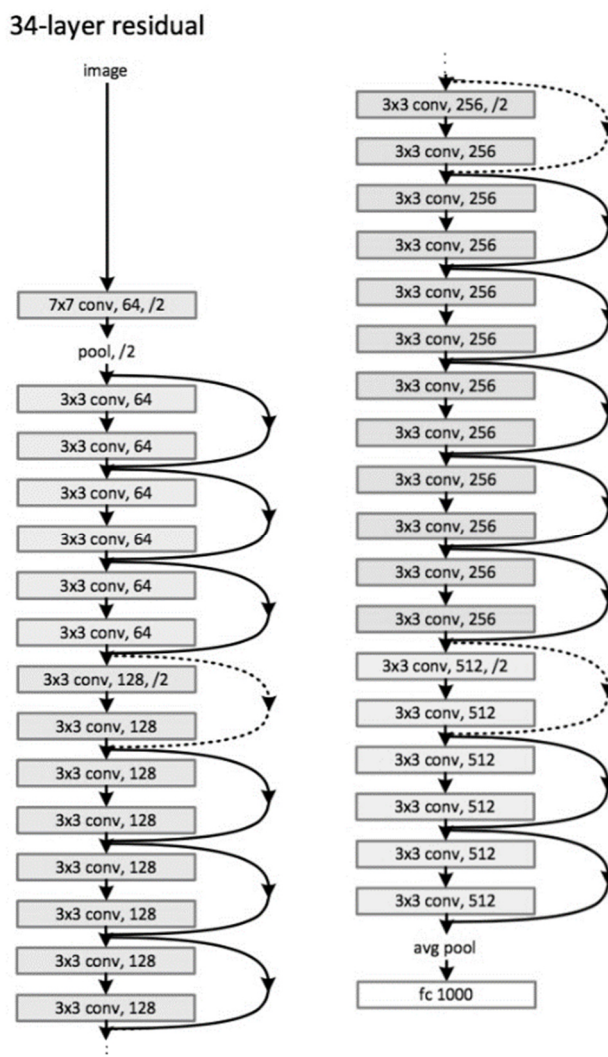


Рисунок 3.3 – Архітектура ResNet

Також необхідно було обрати функцію активації для побудови нейронної мережі. Функція активації визначає вихідне значення нейрона в залежності від результату виваженої суми входів та порогового значення.

Найбільш поширеними функціями активації є східчаста функція активації, лінійна функція активації, сигмоїда та гіперболічний тангенс.

Східчаста функція активації (рис. 3.4) дозволяє отримувати лише бінарну відповідь, що є непригідним для поставленої задачі.

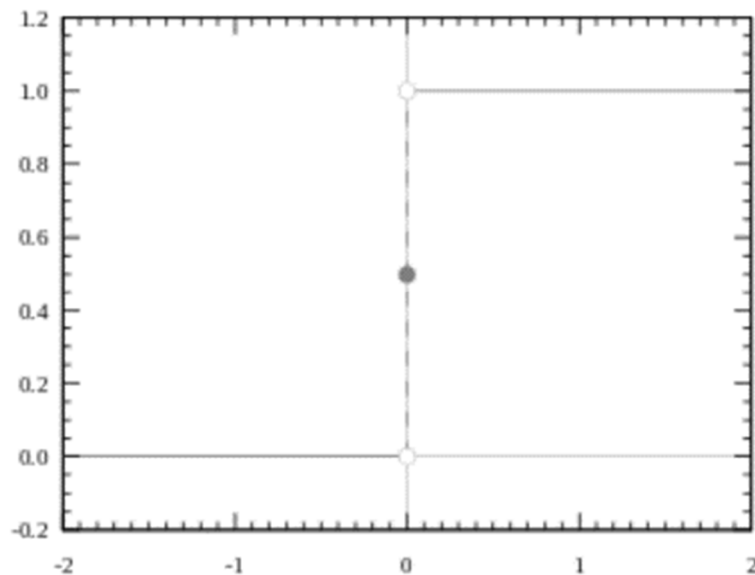


Рисунок 3.4 – Східчаста функція активації

Лінійна функція активації (рис. 3.5) є прямою лінією і пропорційна входу, що дозволяє отримувати спектр значень, але лінійна природа функції означає, що фінальна функція активації в останньому шарі буде просто лінійною функцією від входів на першому шарі, тому ця функція також не є потрібним нам варіантом.

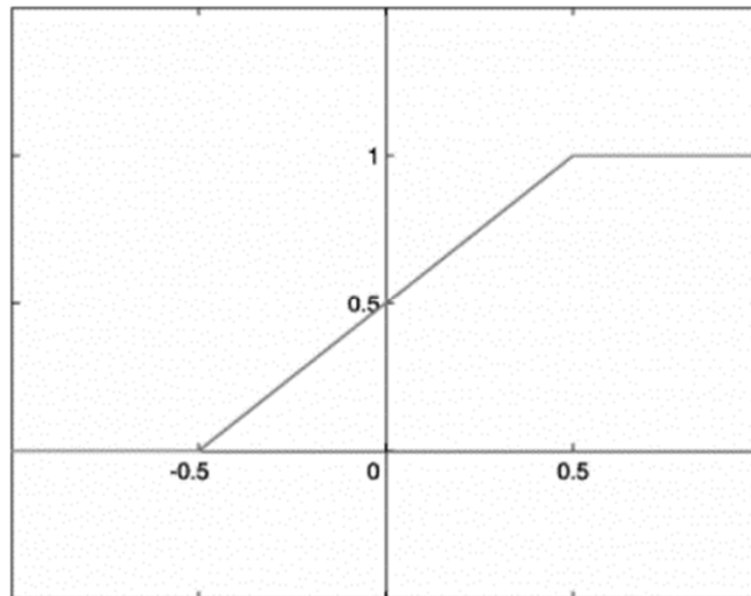


Рисунок 3.5 – Лінійна функція активації

Сигмоїда (рис 3.6) виглядає гладкою і подібна до ступінчастої функції. Сигмоїда — нелінійна за своєю природою, а комбінація таких функцій робить також нелінійну функцію, що дозволяє стікати шари.

Сигмоїда - не бінарна функція, що робить активацію аналоговою, на відміну від ступінчастої функції. Для сигмоїди також характерний гладкий градієнт.

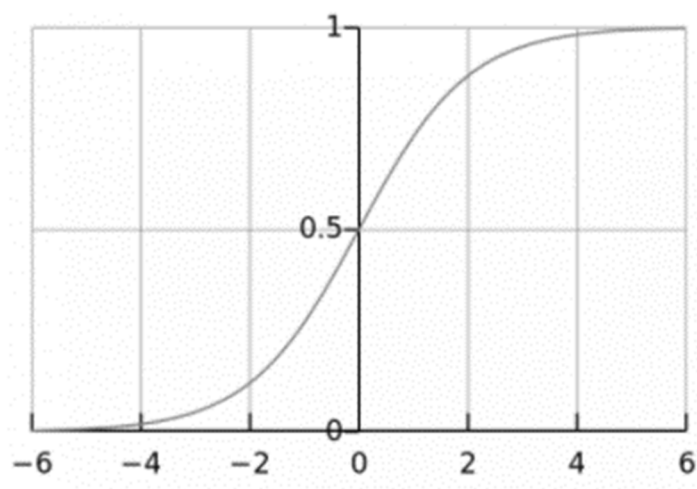


Рисунок 3.6 – Сигмоїда

Гіперболічний тангенс (рис. 3.7) – це скорегована сигмоїдна функція, тому така функція має самі характеристики, як і в сигмоїди. Природа функції також нелінійна та добре підходить для комбінації шарів.

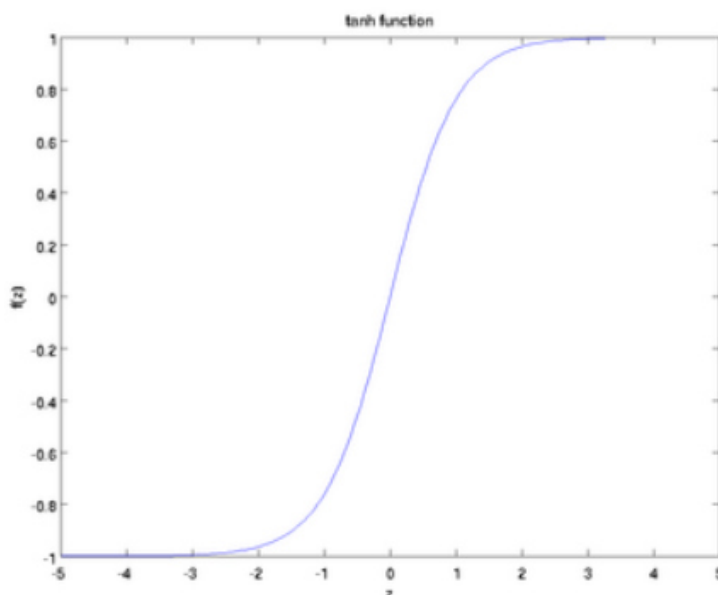


Рисунок 3.7 – Гіперболічний тангенс

Сигмоїда чи гіперболічний тангенс – дуже схожі між собою функції активації та обидві підходять для нейронної мережі за допомогою якої буде здійснюватися побудова системи розпізнавання особистостей.

3.7. Визначення орієнтирів обличчя

Точний орієнтир обличчя та виявлення рис обличчя - важливі операції, що впливають на наступні завдання, сфокусовані на обличчі, такі як розпізнавання обличчя. Визначається орієнтир обличчя як характерна особливість, яка може грати розрізнявальну роль або може бути опорними точками на графі особи. Використовувані орієнтири - це куточки очей, кінчик носа, куточки ніздрів, куточки рота, кінцеві точки дуг брів, мочки вух, ніс та ін., підборіддя тощо. На

рисунку 3.8 зображено приклад розмітки 68 вузлових антропологічних точок для виявлення відбитка обличчя.

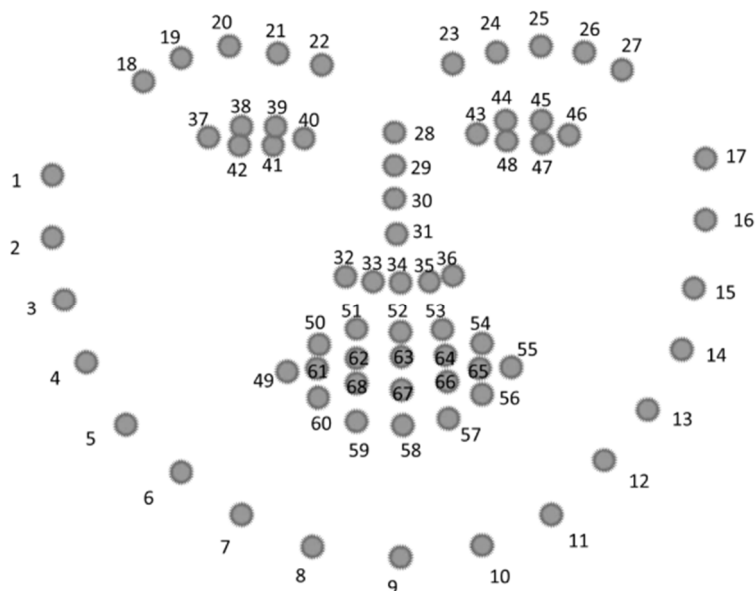


Рисунок 3.8 – 68 вузлових точок обличчя

Вирівнювання граней вирішується з допомогою каскаду функцій регресії. Кожна функція регресії в каскаді ефективно оцінює розріджений набір пікселів, проіндексованих щодо початкової оцінки.

3.8. Режими функціонування програмної моделі

Програмна модель системи розпізнавання особистостей повинна забезпечувати:

- можливість вибору існуючого зображення, його відображення та можливість розпізнати особистість на зображенні;
- можливість обрати декілька зображень та виявити чи одна й та сама людина на них зображена;
- можливість пошуку необхідної особистості та демонстрування знайдених зображень.

Для зберігання інформації про людей, що розпізнаються, а також для зберігання зображень з даними людьми нам знадобиться база даних. За допомогою систем управління базами даних та мови роботи з базами даних SQL ми можемо створити ієрархічну структуру даних, яку можливо буде підключити до нашої системи в програмному коді.

Цю базу даних можна реалізувати за допомогою таких систем управління як Microsoft SQL Server або MySQL.

У процесі самонавчання створена база даних поповнюватиметься зображеннями, що запам'ятовуються системою, а також новими дескрипторами обличь, які раніше були невідомі системі. Мітки нових дескрипторів обличь та прив'язка їх до певних осіб може здійснюватися вручну адміністратором.

3.9. Структурні схеми алгоритмів програмної моделі

Для того, щоб нейронна мережа розпізнавала обличчя на зображення її треба навчити. Навчання нейронної мережі - це процес, у якому параметри нейронної мережі налаштовуються за допомогою моделювання середовища, в яке ця мережа вбудована. Тип навчання визначається способом підстроювання параметрів. Процес навчання (рис. 3.9) являє собою пред'явлення мережі вибірки різних навчальних прикладів.

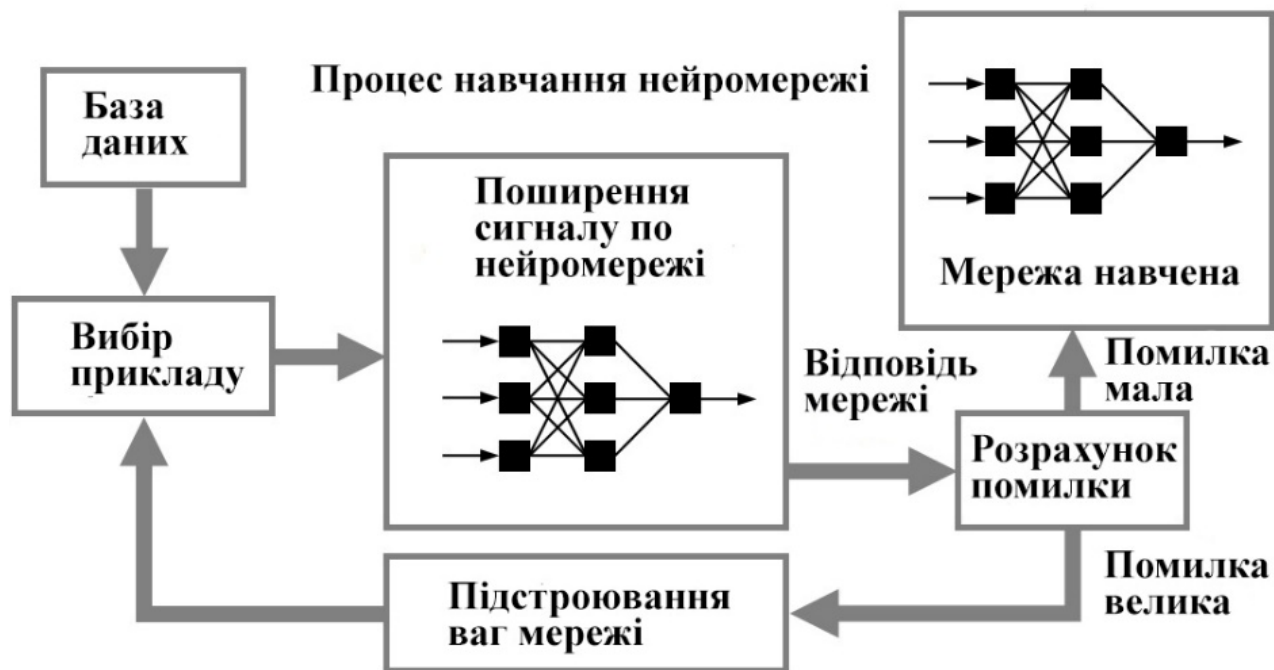


Рисунок 3.9 – Структурна схема алгоритму процесу навчання нейронної мережі

Як одна з основних можливостей нашої системи нам потрібно обрати обличчя та викликати функцію розпізнавання для вилучення дескриптора обличчя координати орієнтирів якого були визначені на зображенні, яке завдяки цьому постане перед нами у вигляді 128-мірного вектора.

Для можливості порівняння нашого зображення з наявними в базі даних або будь-якими іншими, нам потрібно так само зчитати інші зображення, виявити на них обличчя та витягти з них дескриптори, для оптимізації системи можна зберегти ці дескриптори, прив'язані до певних особистостей у базі даних.

При не розпізнанні особи на зображенні система автоматично додає до бази даних дескриптори обличчя на цьому зображенні, що дозволить системі запам'ятовувати нові обличчя і надалі розпізнавати їх. Система буде додатково оснащена функцією, що дозволяє запам'ятовувати та додавати до себе в базу даних нові фотографії особистостей, яких ще немає в системі. Структурна схема алгоритму розпізнавання особистості зображена на рисунку 3.10.

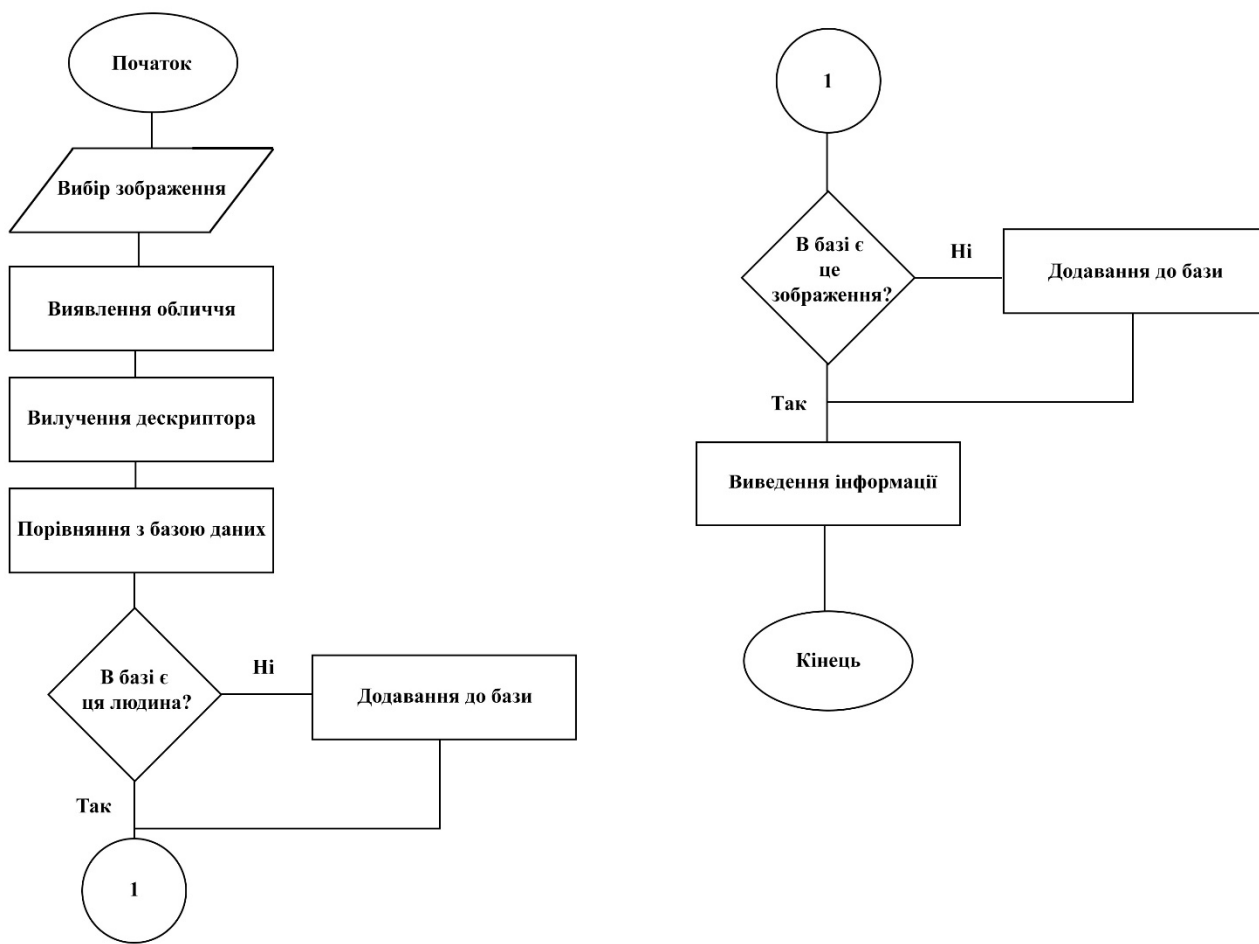


Рисунок 3.10 – Структурна схема алгоритму розпізнавання

Реалізація функції порівняння зображення з іншими буде заснована на обчисленні Евклідової відстані між дескрипторами, яка є геометричною відстанню в багатовимірному просторі. Рекомендується використовувати граничне значення Евклідової відстані між дескрипторами обличчя, що дорівнює 0.6. Якщо Евклідова відстань менша за 0.6, це означає, що фотографії належать одній і тій самій людині.

У системі також буде присутня окрема демонстраційна функція для верифікації особистості шляхом порівняння двох завантажених у програму зображень та визначення одна й та сама на них людина чи ні. Структурна схема алгоритму верифікації зображена на рисунку 3.11.



Рисунок 3.11 – Структурна схема алгоритму верифікації

Додатково розроблено демонстраційну функцію пошуку певної особи в базі даних і видачі нам результатів у вигляді наявних в базі зображень з цією особистістю, а також при необхідності короткої інформації про потрібну особистість. Структурна схема алгоритму пошуку зображена на рисунку 3.12.



Рисунок 3.12 – Структурна схема алгоритму пошуку

3.10. Висновки теоретичного дослідження

У процесі теоретичного дослідження було проведено оцінку ефективності існуючих методів розпізнавання осіб. Було виявлено, що поодинці не одна з цих систем не є придатною для побудови системи розпізнавання особистостей.

Було винесено пропозицію щодо вдосконалення існуючих методів шляхом використання методів подібних до ААМ і АСМ як складової частини системи розпізнавання особистостей основну на згортковій нейронній мережі.

Підібрано найбільш ефективну функцію активації та архітектуру згорткової нейронної мережі.

Було описано структурну побудову програмної моделі, її режими функціонування, а також продемонстровано основні структурні схеми алгоритмів функціонування системи розпізнавання особистостей.

4 РЕЗУЛЬТАТИ ПРОВЕДЕНИХ ДОСЛІДЖЕНЬ

4.1. Установки програмної моделі

Програмний продукт призначений для моделювання роботи системи розпізнавання особистостей реалізовано у доповняльному коді.

Програма написана у середовищі Microsoft Visual Studio на Python 3.9.

Для розробки програмної моделі системи розпізнавання особистостей ми використовуватимемо попередньо навчену модель штучної згорткової залишкової нейронної мережі на основі архітектури ResNet, яку надає бібліотека dlib. Ця конкретна нейронна мережа має 29 згорткових шарів, приймає вхідні дані розміром 150x150x3 та представляє зображення обличчя у вигляді 128-мірних векторів. Модель була кілька разів навчена з використанням різного набору даних, що дозволяє отримати точність розпізнавання обличчя на зображенні 99,38%.

В якості моделі детектора лицьових орієнтирів для розробки програмної моделі було використано модель, яку також пропонує бібліотека dlib, що виділяє 68 основних вузлових точок обличчя на зображенні.

Орієнтири обличчя можна використовувати для вирівнювання обличчя, а потім їх можна деформувати, комбінуючи кілька різних зображень обличчя для створення нового зображення обличчя.

При навчанні подібної моделі основні вузлові точки обличчя помічаються вручну на кожному зображенні. У нашому випадку орієнтирами є щелепа та підборіддя, очі та брови, внутрішня та зовнішня область губ та носа.

Для пошуку орієнтирів на обличчі необхідно спочатку визначити обличчя. Dlib має свій готовий модуль переднього детектора обличчя. Для зчитування зображення нам знадобиться модуль io з бібліотеки skimage, яку потрібно буде імпортувати в наш проект, як і бібліотеку dlib. Детектор орієнтирів обличчя dlib поверне об'єкт форми, що містить 68 координат областей орієнтирів обличчя.

Програма має користувальницький інтерфейс с трьома вкладками під різні режими функціонування.

У першій вкладці яку ми називаємо "Розпізнавання" - буде реалізована можливість додавання зображення за допомогою карти пікселів елемента label, а також методу OpenFileDialog у кодї програми, який дозволить вибрати зображення на комп'ютері натиснувши на кнопку "Додати зображення" при натисканні на яку буде викликатися реалізація функції на основі методу OpenFileDialog. Сама кнопка буде елементом pushButton, який пропонує наш графічний редактор.

Після встановлення потрібного зображення в якості карти пікселів, буде можливість натиснути на кнопку "Розпізнати" яка буде викликати функцію обчислення та порівняння дескрипторів виявленого на зображенні обличчя і видавати інформацію про зображену особу. У разі відсутності інформації про дану особу у базі даних, буде створюватися гілка, в яку будуть запам'ятовуватися обчислені дескриптори. Також буде реалізована функція на основі структури if - else яка дозволить за відсутності схожого зображення в базі даних, додавати його туди і запам'ятовувати. Вся інформація щодо особи буде відображатися в окремому елементі label. Користувальницький інтерфейс вкладки «Розпізнавання» зображено на рисунку 4.1.

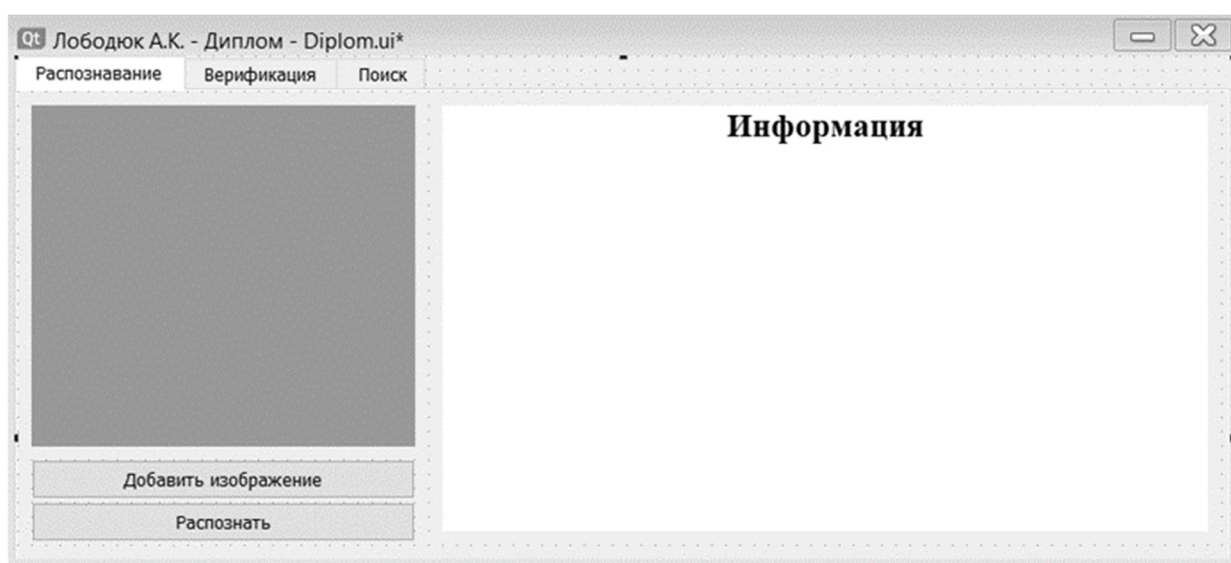


Рисунок 4.1 – Користувальницький інтерфейс вкладки «Розпізнавання»

Друга вкладка, яку ми назвали "Верифікація", матиме можливість завантажити дві різні фотографії та визначити та сама людина зображена на цих фотографіях чи ні. Карти пікселів на основі елемента label також за допомогою схожої кнопки pushButton та методу OpenFileDialog завантажуватимуть потрібні зображення.

Далі буде доступна можливість натиснути на кнопку під назвою "Верифікація", яка буде викликати чергову функцію def, в якій будуть зчитуватися дескриптори обличчя на зображеннях та порівняння їх між собою за допомогою обчислення Евклідової відстані. У разі отримання значення 0.6 або нижче в окремому елементі label буде писатися інформація про те, одна й та ж це людина чи ні. Користувальницький інтерфейс вкладки «Верифікація» зображено на рисунку 4.2.

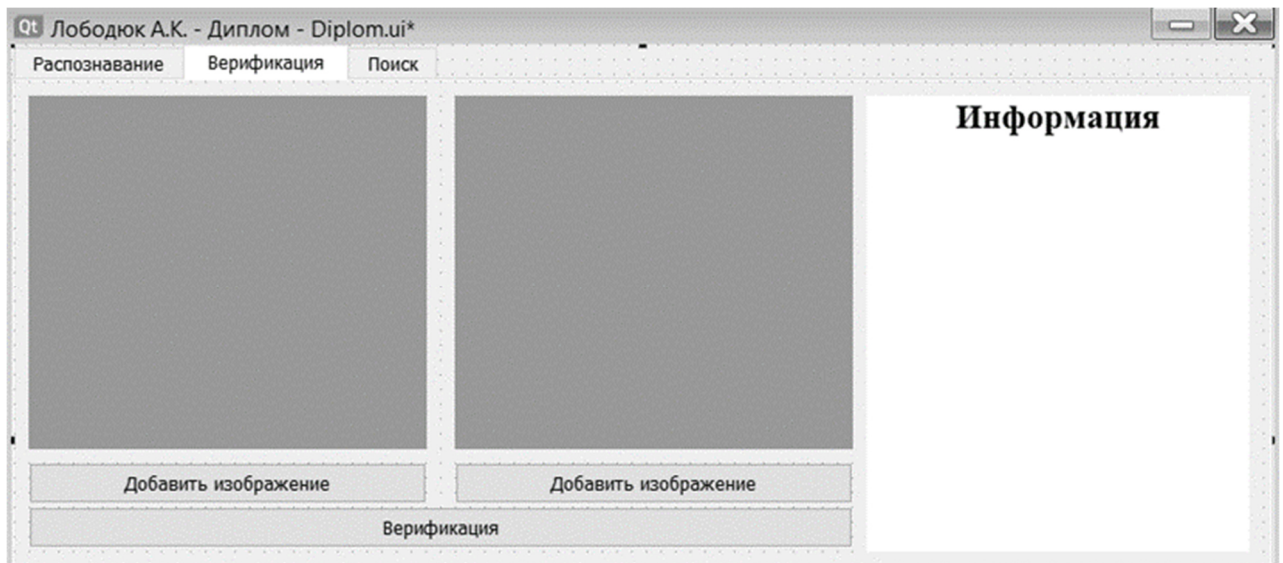


Рисунок 4.2 – Користувальницький інтерфейс вкладки «Верифікація»

Третя експериментальна вкладка демонструватиме можливість використання даної системи в пошукових цілях, в ній буде реалізовано кілька полів під карти пікселів елементів label, елемент lineEdit в якому користувач зможе вписати свій пошуковий запит і при натисканні на кнопку "Пошук" буде викликатись функція яка буде, як наприклад, демонструвати кілька випадкових фотографій цієї особи

наявних у базі даних, якщо таких немає, буде видаватися напис, що нічого не знайдено за цим запитом. Користувальницький інтерфейс вкладки «Пошук» зображено на рисунку 4.3.

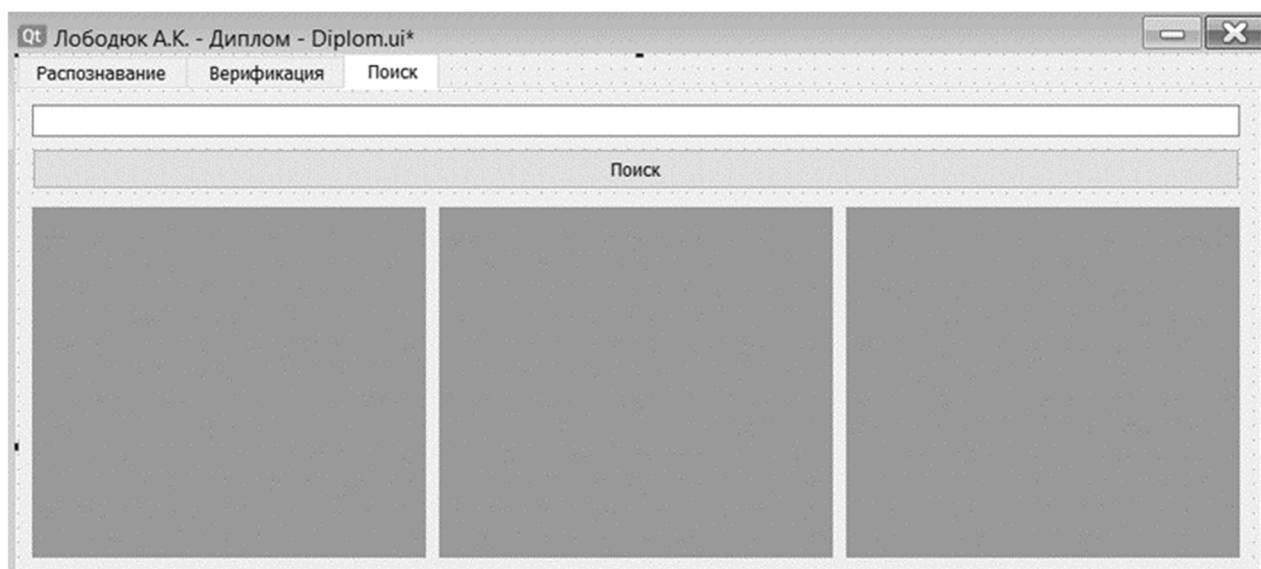


Рисунок 4.3 – Користувальницький інтерфейс вкладки «Пошук»

4.2. Моделювання та його результати

У процесі моделювання роботи запропонованої системи розпізнавання особистостей було проведено оцінку її швидкодії, і навіть продемонстровано її працездатність.

Насамперед було розглянуто вкладку «Розпізнавання» та проведено спробу розпізнавання особи на зображенні, дескриптори якої було вилучено з іншого зображення та попередньо занесено до системи. Результатом розпізнавання було видана правильна інформація системи щодо особистості зображенні. Перший результат продемонстровано на зображенні 4.4.

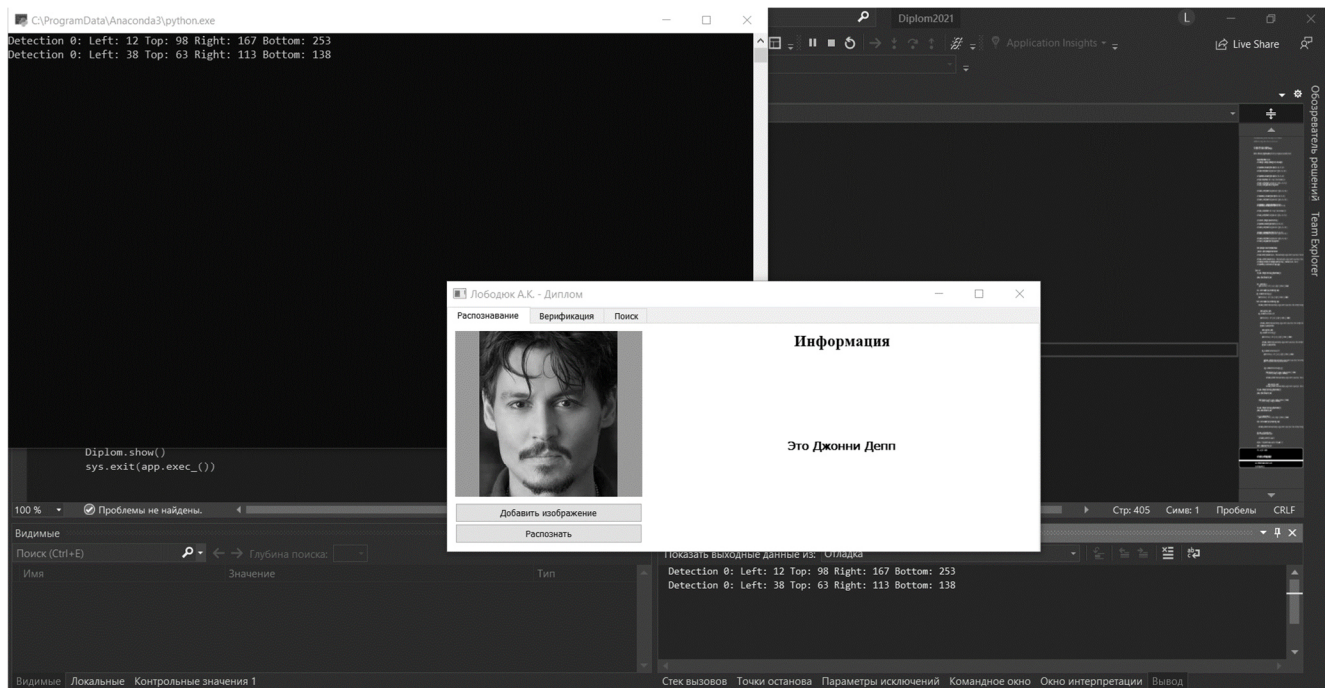


Рисунок 4.4 – Перший результат дієздатності вкладки «Розпізнавання»

Для оцінки точності розпізнавання було обрано інше зображення цієї людини і спроба розпізнавання його особи на іншому зображенні. Другий результат продемонстровано на рисунку 4.5.

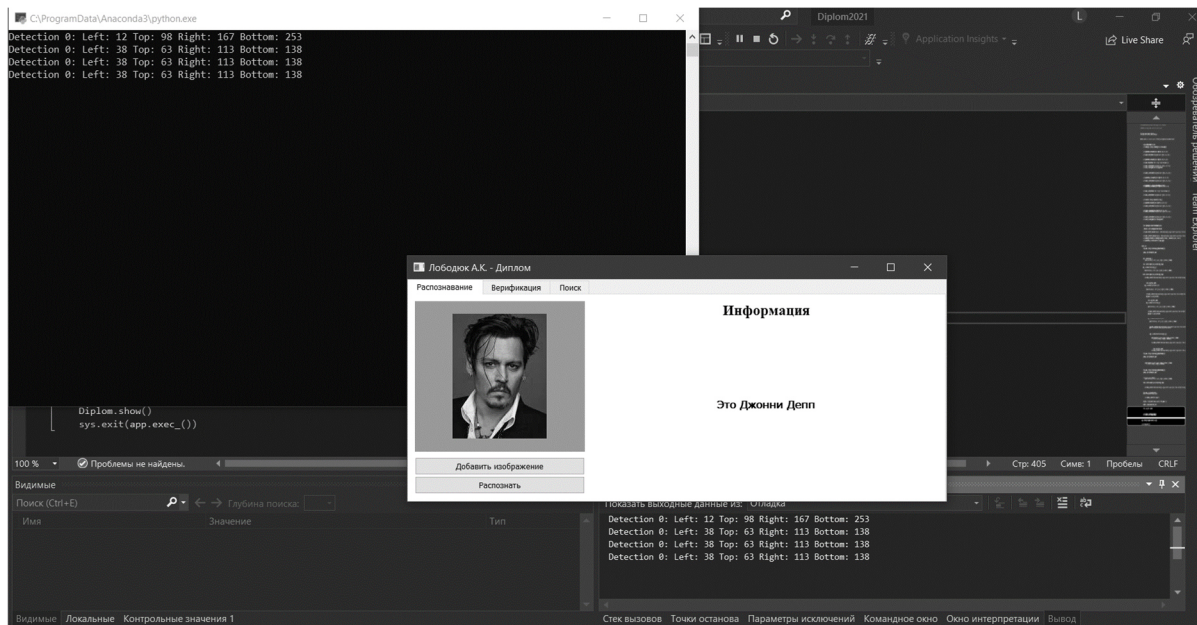


Рисунок 4.4 – Другий результат дієздатності вкладки «Розпізнавання»

Першу особистість, дескриптори якої були занесені до бази даних система розпізнає чітко на більшості наданих зображень, незалежно від ракурсу, пріскі, освітленості та якості зображення. Також система додає до свого каталогу надані їй зображення, яких там ще немає.

Для переконання в працездатності першої вкладки варто перевірити й інші зображення особистостей, дескриптори яких були попередньо додані в базу. Третій результат продемонстровано на рисунку 4.5.

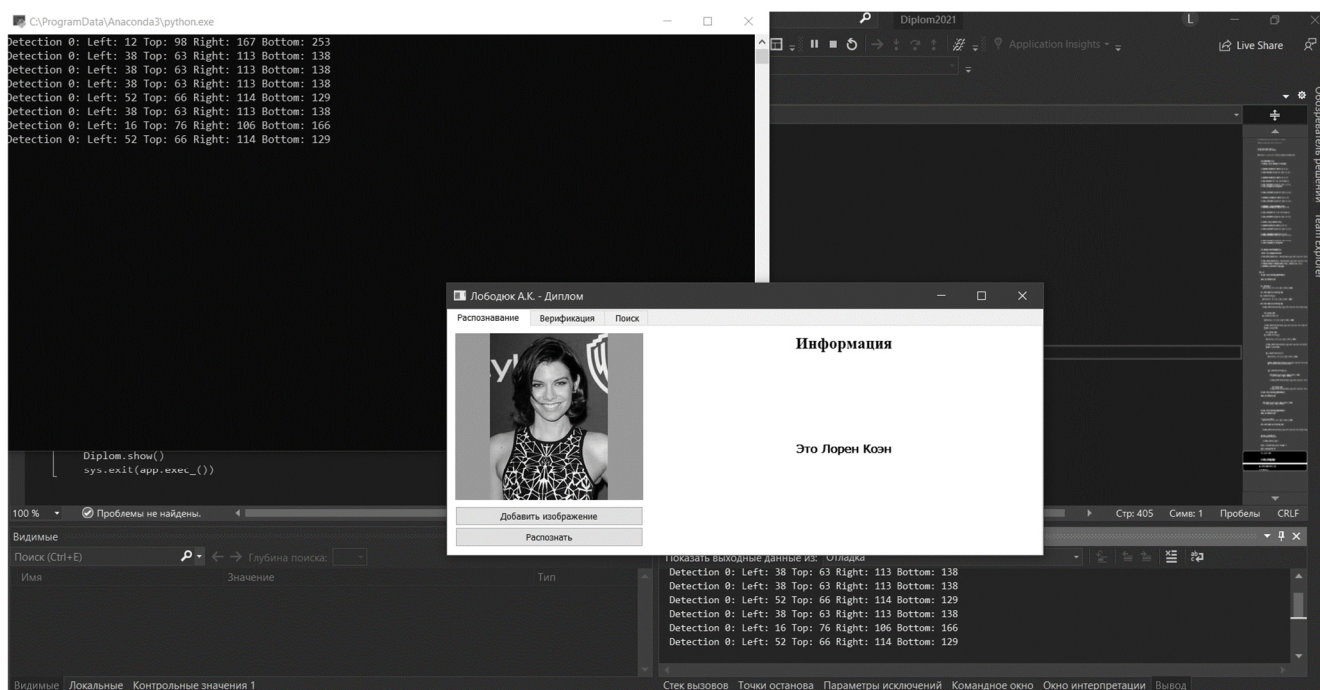


Рисунок 4.5 – Третій результат дієздатності вкладки «Розпізнавання»

Зображення іншої особистості також було чітко розпізнане системою. Далі було розглянуто працездатність вкладки «Верифікація». Для початку було проведено спробу порівняти два зображення різних людей. Перший результат працездатності вкладки «Верифікація» зображено на рисунку 4.6.

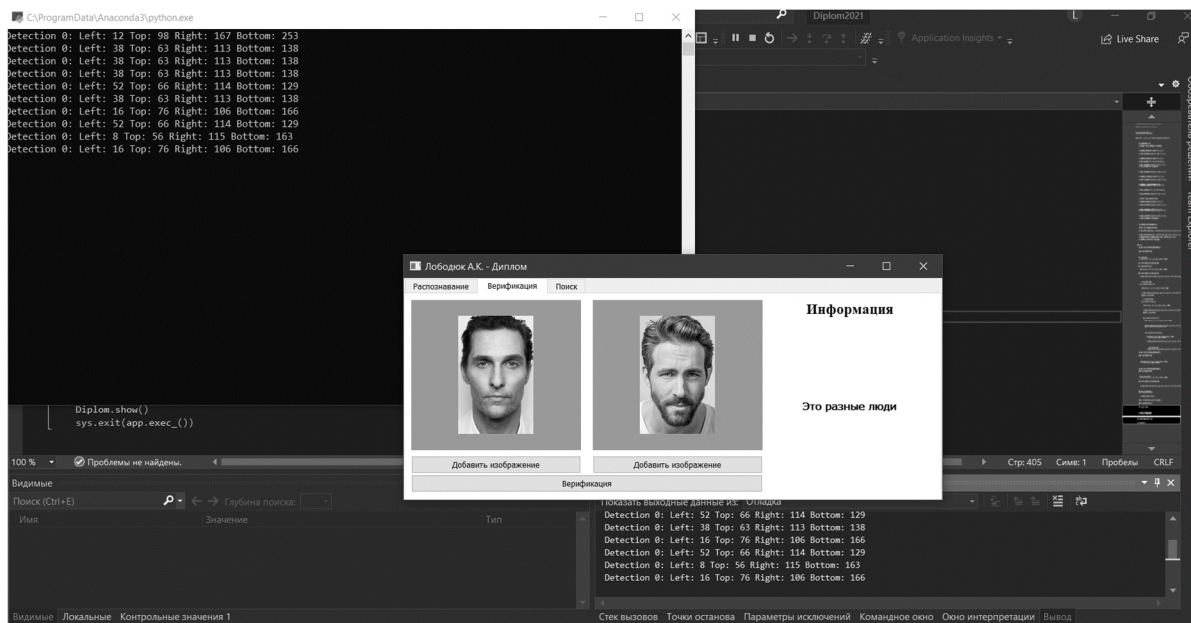


Рисунок 4.6 – Перший результат дієздатності вкладки «Верифікація»

Система вирішила, що на зображеннях знаходяться різні люди, що є правильною відповіддю. Тепер давайте порівняємо зображення двох однакових людей. Другий результат працездатності вкладки «Верифікація» зображено на рисунку 4.7.

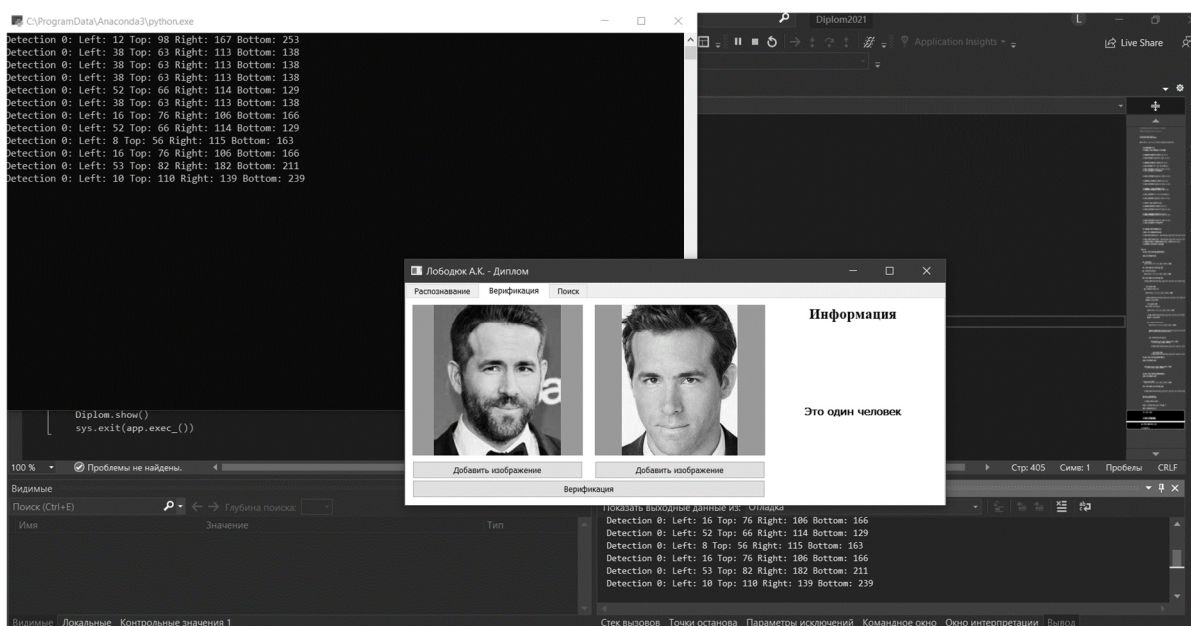


Рисунок 4.7 – Другий результат дієздатності вкладки «Верифікація»

Система прийняла вірне рішення і розпізнала, що людина на обох зображеннях той самий, незважаючи на очевидну різницю між особами на зображеннях, що говорить про те, що система досить точна. Для закріплення проведемо ще одну спробу порівняння. Третій результат працездатності вкладки «Верифікація» зображено на рисунку 4.8.

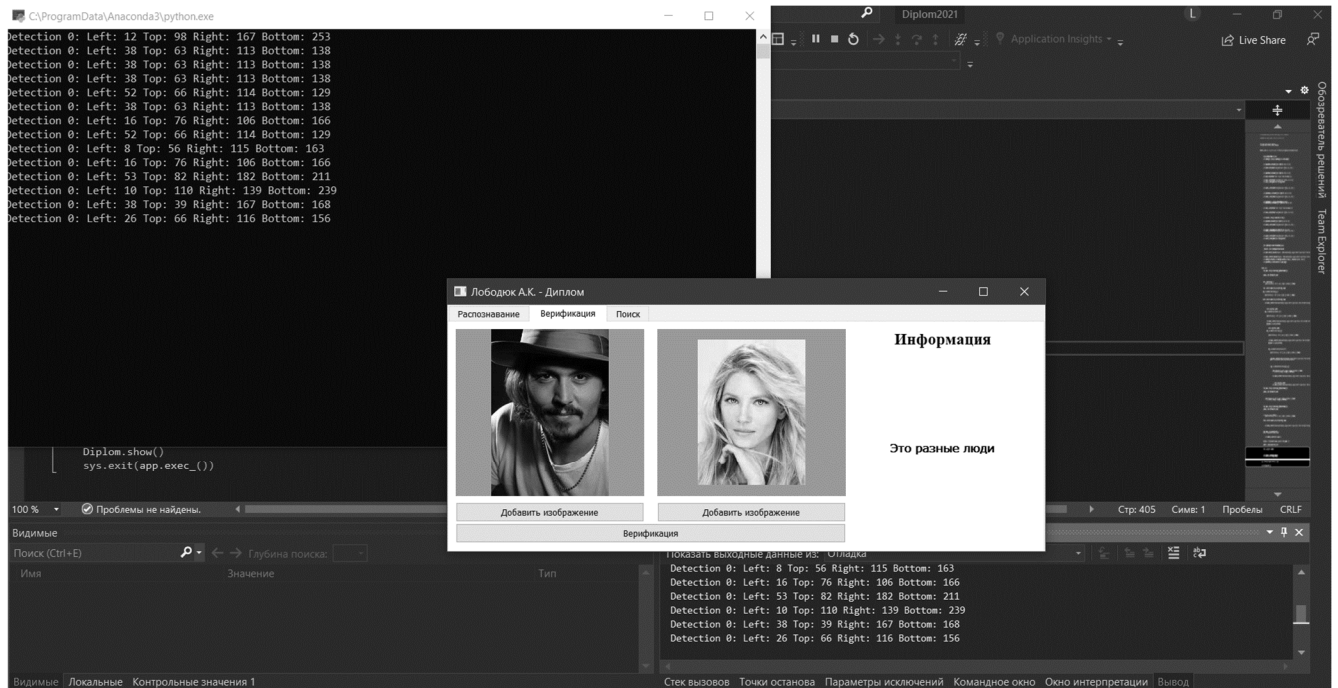


Рисунок 4.8 – Третій результат дієздатності вкладки «Верифікація»

Система знову дала правильну відповідь. Перейдемо до останньої демонстраційної вкладки «Пошук». Спробуємо знайти кілька зображень особи, яку ми введемо у пошуковому рядку. Перший результат працездатності вкладки «Пошук» зображено на рисунку 4.9.

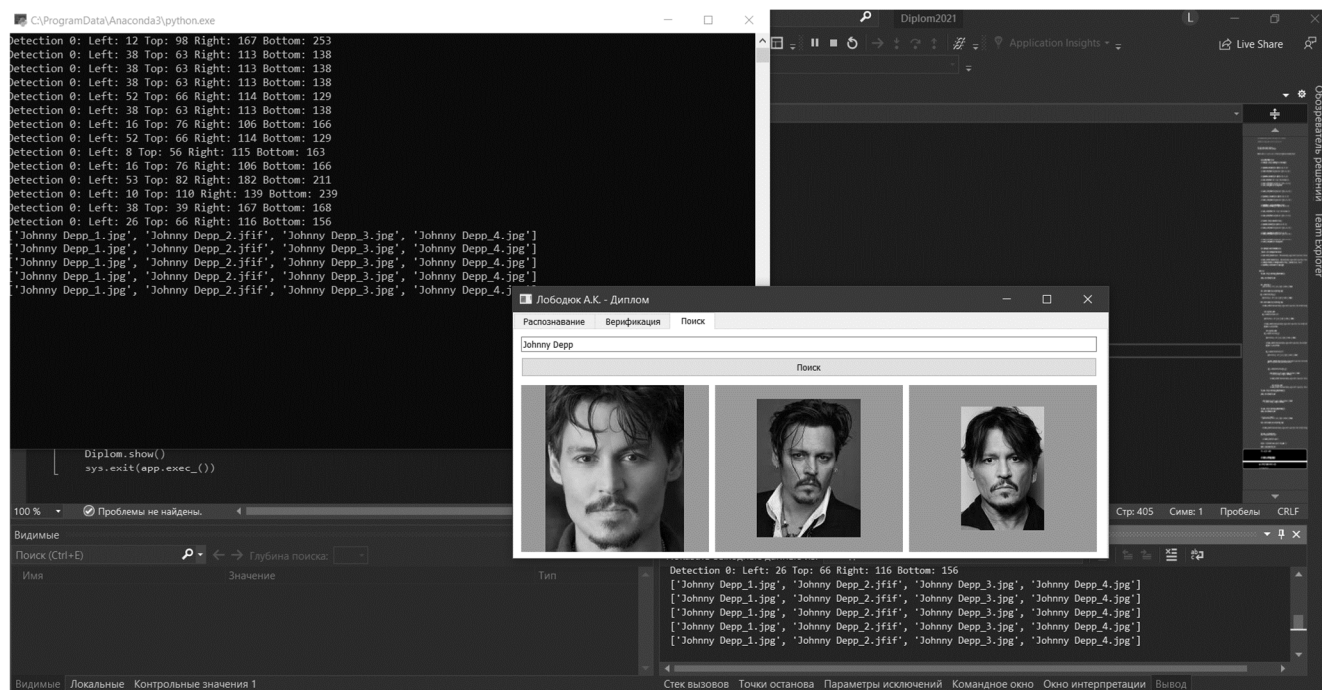


Рисунок 4.9 – Перший результат дієздатності вкладки «Пошук»

Система дійсно продемонструвала нам зображення людини, яку ми шукали, що є у ній у базі. На рисунку 4.10 зображено другий результат дієздатності.

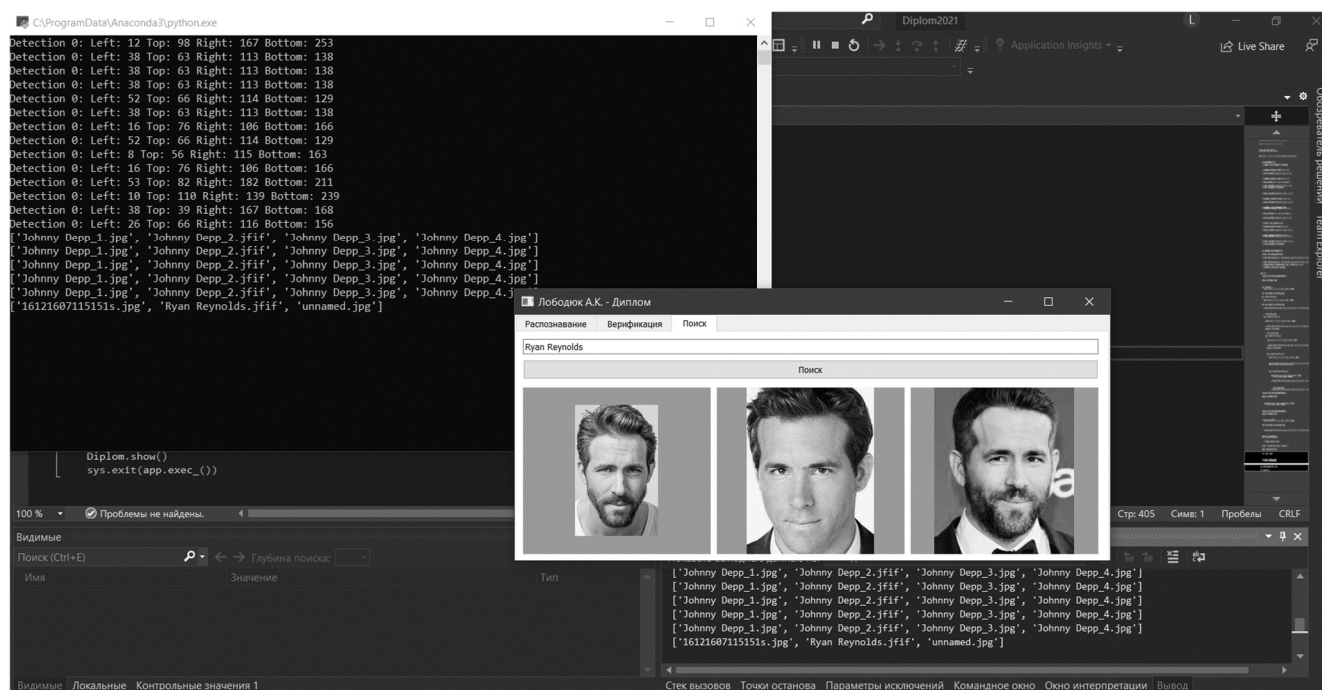


Рисунок 4.10 – Другий результат дієздатності вкладки «Пошук»

На рисунку 4.11 продемонстровано третій результат працездатності вкладки «Пошук», в якому ми намагаємося знайти зображення, яких немає в базі даних.

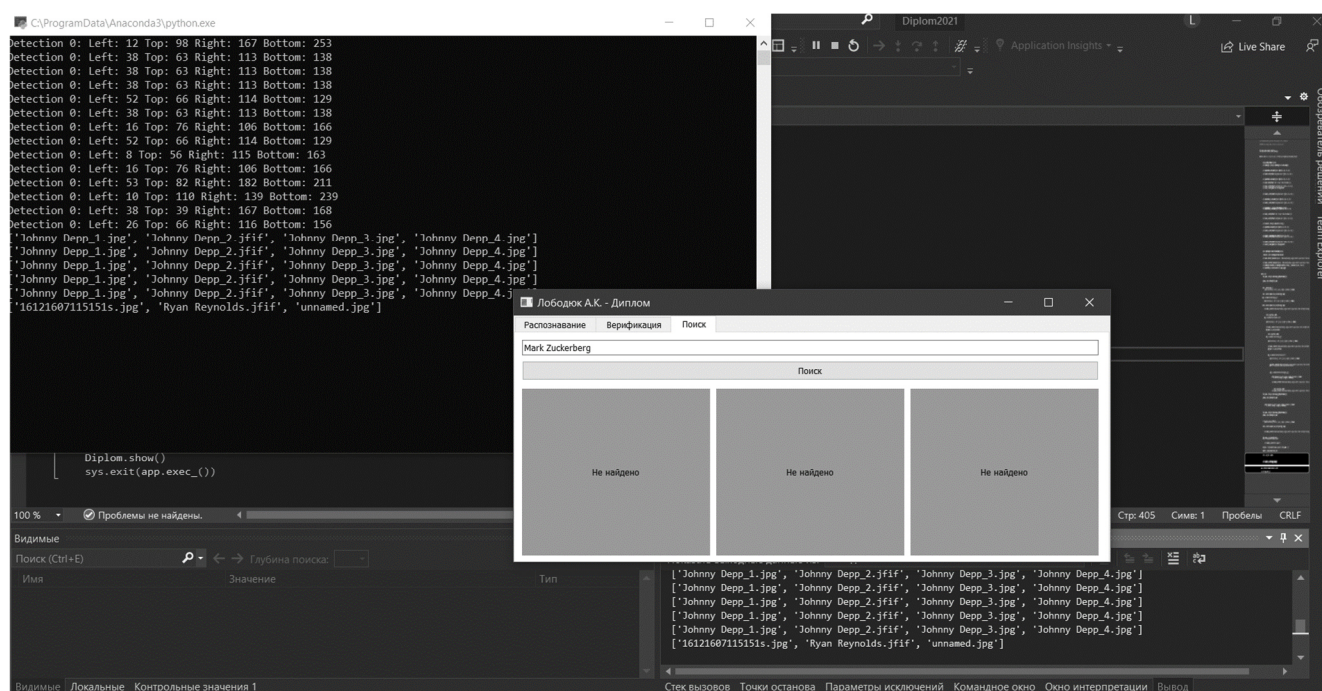


Рисунок 4.11 – Третій результат дієздатності вкладки «Пошук»

Система повідомляє нам, що результатів за даним пошуковим запитом не було знайдено.

4.3. Висновки моделювання

Змодельована система проявила себе дуже добре у процесі випробувань та перевірки точності та якості виконання її основних функцій. Розпізнавання обличь працює досить точно, щоб розрізняти особи при різних змінах зображення.

Виходячи з результатів моделювання можна сміливо сказати, що нам вдалося вдосконалити та оптимізувати існуючі методи побудови систем розпізнавання особистостей за допомогою біометричних систем розпізнавання обличь.

ЗАГАЛЬНІ ВИСНОВКИ

У цій роботі було досліджено питання інформаційних інтелектуальних систем - систем розпізнавання особистості. Проведене теоретичне дослідження та запропоновано вдосконалення існуючих рішень, обґрунтована потреба проведення дослідження.

Виходячи з аналізу інформації, отриманої в ході проведення теоретичного дослідження, було запропоновано оригінальний ефективний метод для побудови систем розпізнавання особистості на зображеннях. Було описано алгоритм побудови системи розпізнавання відомих для системи особистостей з можливістю так званого самонавчання, а також можливістю масштабування та інтеграції надалі в будь-яких пошукових системах під різні потреби, пов'язані з ідентифікацією та осіб. Використання запропонованого алгоритму дозволяє структурувати інформацію при обробці зображень обличчя та здійснювати пошук та групування фотографій заданої людини.

Виконано моделювання системи розпізнавання особистостей яка базується на запропонованому методі. Успішне моделювання системи та перевірка дієздатності її функцій, а також продемонстрована точність розпізнавання означає, що запропоновані покращення існуючих методів вдалися.

Результатом виконаної роботи є розроблений метод побудови інтелектуальної інформаційної системи розпізнавання відомих особистостей, що самонавчається, яку можна використовувати в будь-яких пошукових системах.

Використаний метод є універсальним рішенням, і дуже ефективним рішенням розробки таких систем нині. Система розпізнавання з досить великою точністю може визначити особистості на зображенні, також завдяки функціям вирівнювання орієнтирів необов'язково, щоб обличчя на зображенні було розташовано суворо у анфас.

Розроблений метод побудови подібних систем має можливість масштабування та розширення своїх можливостей, а також своєї бази даних до будь-яких необхідних розмірів.

Результат даної роботи є кросплатформним рішенням, завдяки чому з ним можна працювати на будь-якій існуючій операційній системі.

Завдяки використаній штучній згортковій залишковій нейронній мережі архітектури ResNet система забезпечує один з найвищих результатів точності розпізнавання, що дозволяє даній розробці конкурувати з будь-якими існуючими рішеннями.

Графічний інтерфейс, який розглядається в розробці даного методу, дозволяє нам наочно і зручно продемонструвати функції та можливості нашої системи, а також можливе її використання. Розроблений графічний інтерфейс досить просто змінити і оснастити практично будь-якими функціями, а також змінити його візуально для найбільш прийняттого під ті або інші потреби естетичного рішення.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Л. Шапиро, Дж. Стокман. Компьютерное зрение = Computer Vision. — М.: Бинوم. Лаборатория знаний, 2006. — 752 с.
2. Желтов С.Ю. и др. Обработка и анализ изображений в задачах машинного зрения. — М.: Физматкнига, 2010. — 672 с.
3. Друки А. А. Система поиска, выделения и распознавания лиц на изображениях // Известия Томского политехнического университета [Известия ТПУ]. — 2011. — Т. 318, № 5. — С. 64—70.
4. Dreyfus, Stuart E. "Artificial neural networks, back propagation, and the KelleyBryson gradient procedure". Journal of Guidance, Control, and Dynamics. —1990. —13 (5): — 926–928.
5. Gates, Kelly. Our Biometric Future: Facial Recognition Technology and the Culture of Surveillance. NYU Press. pp. - 2011 - 48–49.
6. McCulloch, Warren; Walter Pitts. "A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics. —1943. —5 (4): —115–133.
7. Stan Z. Li, Anil K. Jain. Handbook of Face Recognition, Stan Z. Li, Anil K. Jain, Second Edition, Springer, 2011. — 699 p.
8. Le Cun Y., Bengio Y. Convolutional networks for images, speech and time series // The handbook of brain theory and neural networks. — 1998. — V. 7. — № 1. — P. 255–258.
9. Viola, P.; Jones, P. : Robust real-time object detection. Int. J. Comput. Vis. (IJCV), 57 (2) (2004), 137–154.
10. Takahashi, K.; Mitsukura, Y.: Adaptively designed shape regression model for facial point detection. J. Inst. Image Inf. Telev. Eng., 69 (3) (2015), J126–J132.
11. P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In CVPR, pages 545–552, 2011.

12. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.
13. D. Cristinacce and T. F. Cootes. Boosted regression active shape models. In *BBCV*, pages 79.1–79.10, 2007.
14. X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012.
15. Zeiler, F., Krishnan, D., Taylor, G., and Fergus, R. (2010). Deconvolutional networks. In *CVPR'2010*.
16. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, 1(4):541-551, Winter 1989.
17. Ji, Shuiwang; Xu, Wei; Yang, Ming; Yu, Kai (2013-01-01). "3D Convolutional Neural Networks for Human Action Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 35 (1): 221–231.
18. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2015-12-10). "Deep Residual Learning for Image Recognition".
19. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). "Deep Residual Learning for Image Recognition" (PDF). *Proc. Computer Vision and Pattern Recognition (CVPR)*, IEEE. Retrieved 2020-04-23.
20. Nakashima, Yuta, Noboru Babaguchi, and Jianping Fan. "Detecting intended human objects in human-captured videos." *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010 IEEE Computer Society Conference on. IEEE, 2010. Nakashima, Y.; Babaguchi, N.; Fan, J. (2010). "Detecting intended human objects in human-captured videos". 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. p. 33.

ДОДАТОК А

Код програмної моделі

```
import dlib
import numpy
import os
import shutil
import os.path
import random
from skimage import io
from scipy.spatial import distance
from PyQt5 import QtCore, QtGui, QtWidgets

sp = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
facerec =
dlib.face_recognition_model_v1('dlib_face_recognition_resnet_model_v1.dat')
detector = dlib.get_frontal_face_detector()

class Ui_Diplom(object):
    def setupUi(self, Diplom):
        Diplom.setObjectName("Diplom")
        Diplom.resize(860, 355)
        self.centralwidget = QtWidgets.QWidget(Diplom)
        self.centralwidget.setObjectName("centralwidget")
        self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)
        self.tabWidget.setGeometry(QtCore.QRect(0, 0, 861, 541))
        self.tabWidget.setObjectName("tabWidget")
        self.tab = QtWidgets.QWidget()
        self.tab.setObjectName("tab")
```

```
self.pushButton1 = QtWidgets.QPushButton(self.tab)
self.pushButton1.setGeometry(QtCore.QRect(10, 260, 271, 28))
self.pushButton1.setObjectName("pushButton1")
self.label1 = QtWidgets.QLabel(self.tab)
self.label1.setGeometry(QtCore.QRect(10, 10, 271, 241))
self.label1.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1.setText("")
self.label1.setScaledContents(False)
self.label1.setAlignment(QtCore.Qt.AlignCenter)
self.label1.setWordWrap(False)
self.label1.setObjectName("label1")
self.pushButton2 = QtWidgets.QPushButton(self.tab)
self.pushButton2.setGeometry(QtCore.QRect(10, 290, 271, 28))
self.pushButton2.setObjectName("pushButton2")
self.label = QtWidgets.QLabel(self.tab)
self.label.setGeometry(QtCore.QRect(300, 10, 541, 31))
self.label.setStyleSheet("font: 75 14pt \"Times New Roman\";\n"
"background-color: rgb(255, 255, 255);")
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(self.tab)
self.label_2.setGeometry(QtCore.QRect(300, 40, 541, 271))
self.label_2.setStyleSheet("background-color: rgb(255, 255, 255);")
self.label_2.setText("")
self.label_2.setAlignment(QtCore.Qt.AlignCenter)
self.label_2.setObjectName("label_2")
self.tabWidget.addTab(self.tab, "")
self.tab_2 = QtWidgets.QWidget()
self.tab_2.setObjectName("tab_2")
self.label1_2 = QtWidgets.QLabel(self.tab_2)
self.label1_2.setGeometry(QtCore.QRect(10, 10, 271, 241))
```

```
self.label1_2.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1_2.setText("")
self.label1_2.setScaledContents(False)
self.label1_2.setAlignment(QtCore.Qt.AlignCenter)
self.label1_2.setWordWrap(False)
self.label1_2.setObjectName("label1_2")
self.pushButton1_2 = QtWidgets.QPushButton(self.tab_2)
self.pushButton1_2.setGeometry(QtCore.QRect(10, 260, 271, 28))
self.pushButton1_2.setObjectName("pushButton1_2")
self.label1_3 = QtWidgets.QLabel(self.tab_2)
self.label1_3.setGeometry(QtCore.QRect(300, 10, 271, 241))
self.label1_3.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1_3.setText("")
self.label1_3.setScaledContents(False)
self.label1_3.setAlignment(QtCore.Qt.AlignCenter)
self.label1_3.setWordWrap(False)
self.label1_3.setObjectName("label1_3")
self.pushButton1_3 = QtWidgets.QPushButton(self.tab_2)
self.pushButton1_3.setGeometry(QtCore.QRect(300, 260, 271, 28))
self.pushButton1_3.setObjectName("pushButton1_3")
self.pushButton1_4 = QtWidgets.QPushButton(self.tab_2)
self.pushButton1_4.setGeometry(QtCore.QRect(10, 290, 271, 28))
self.pushButton1_4.setObjectName("pushButton1_4")
self.label_3 = QtWidgets.QLabel(self.tab_2)
self.label_3.setGeometry(QtCore.QRect(580, 10, 261, 31))
self.label_3.setStyleSheet("font: 75 14pt \"Times New Roman\";\n"
"background-color: rgb(255, 255, 255);")
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(self.tab_2)
self.label_4.setGeometry(QtCore.QRect(580, 40, 261, 281))
```



```
self.label_4.setStyleSheet("background-color: rgb(255, 255, 255);")
self.label_4.setText("")
self.label_4.setAlignment(QtCore.Qt.AlignCenter)
self.label_4.setObjectName("label_4")
self.tabWidget.addTab(self.tab_2, "")
self.tab_3 = QtWidgets.QWidget()
self.tab_3.setObjectName("tab_3")
self.lineEdit = QtWidgets.QLineEdit(self.tab_3)
self.lineEdit.setGeometry(QtCore.QRect(10, 10, 831, 22))
self.lineEdit.setObjectName("lineEdit")
self.pushButton = QtWidgets.QPushButton(self.tab_3)
self.pushButton.setGeometry(QtCore.QRect(10, 40, 831, 28))
self.pushButton.setObjectName("pushButton")
self.label1_4 = QtWidgets.QLabel(self.tab_3)
self.label1_4.setGeometry(QtCore.QRect(10, 80, 271, 241))
self.label1_4.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1_4.setText("")
self.label1_4.setScaledContents(False)
self.label1_4.setAlignment(QtCore.Qt.AlignCenter)
self.label1_4.setWordWrap(False)
self.label1_4.setObjectName("label1_4")
self.label1_5 = QtWidgets.QLabel(self.tab_3)
self.label1_5.setGeometry(QtCore.QRect(290, 80, 271, 241))
self.label1_5.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1_5.setText("")
self.label1_5.setScaledContents(False)
self.label1_5.setAlignment(QtCore.Qt.AlignCenter)
self.label1_5.setWordWrap(False)
self.label1_5.setObjectName("label1_5")
self.label1_6 = QtWidgets.QLabel(self.tab_3)
```

```

self.label1_6.setGeometry(QQtCore.QRect(570, 80, 271, 241))
self.label1_6.setStyleSheet("background-color: rgb(154, 154, 154);")
self.label1_6.setText("")
self.label1_6.setScaledContents(False)
self.label1_6.setAlignment(QQtCore.Qt.AlignCenter)
self.label1_6.setWordWrap(False)
self.label1_6.setObjectName("label1_6")
self.tabWidget.addTab(self.tab_3, "")
Diplom.setCentralWidget(self.centralwidget)

self.retranslateUi(Diplom)
self.tabWidget.setCurrentIndex(0)
QQtCore.QMetaObject.connectSlotsByName(Diplom)

self.add_functions()

def retranslateUi(self, Diplom):
    _translate = QQtCore.QCoreApplication.translate
    Diplom.setWindowTitle(_translate("Diplom", "Лободюк А.К. - ДИПЛОМ"))
    self.pushButton1.setText(_translate("Diplom", "Добавить изображение"))
    self.pushButton2.setText(_translate("Diplom", "Распознать"))
    self.label.setText(_translate("Diplom",
                                   "<html><head/><body><p
align='center'><span
                                   style='
                                   font-size:14pt;
                                   font-
weight:600;'>Информация</span></p></body></html>"))
    self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), _translate("Diplom",
"Распознавание"))
    self.pushButton1_2.setText(_translate("Diplom", "Добавить изображение"))
    self.pushButton1_3.setText(_translate("Diplom", "Добавить изображение"))
    self.pushButton1_4.setText(_translate("Diplom", "Верификация"))

```

```

self.label_3.setText(_translate("Diplom",
                                "<html><head/><body><p
align=\"center\"><span                style=\"                font-size:14pt;                font-
weight:600;\">Информация</span></p></body></html>"))
self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
_translate("Diplom", "Верификация"))
self.pushButton.setText(_translate("Diplom", "Поиск"))
self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_3),
_translate("Diplom", "Поиск"))

```

```
def add_functions(self):
```

```

self.pushButton1.clicked.connect(self.open_image1)
self.pushButton1_2.clicked.connect(self.open_image2)
self.pushButton1_3.clicked.connect(self.open_image3)
self.pushButton1_4.clicked.connect(self.vert)
self.pushButton2.clicked.connect(self.rp)
self.pushButton.clicked.connect(self.search)

```

```
face = None
```

```
face0 = None
```

```
fafa = None
```

```
def open_image1(self):
```

```

file_name = QtWidgets.QFileDialog.getOpenFileName()[0]
if not file_name:
    return

```

```
pixmap = QtGui.QPixmap(file_name)
```

```
self.label1.setPixmap(pixmap)
```

```
img = io.imread(file_name)
```

```
global fafa
fafa = file_name

dets = detector(img, 1)

for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img, d)

global face
face = facerec.compute_face_descriptor(img, shape)

def rp(self):

    img = io.imread('Johnny Depp.jpg')

    dets = detector(img, 1)

    for k, d in enumerate(dets):
        print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
            k, d.left(), d.top(), d.right(), d.bottom()))
        shape = sp(img, d)

    global face0
    face0 = facerec.compute_face_descriptor(img, shape)

    a = distance.euclidean(face, face0)
    if a <= 0.55:
```

```

self.label_2.setText("<html><head/><body><p align=\"center\"><span style=\"
font-size:10pt; font-weight:500;\">Это Джонни Депп</span></p></body></html>")
path5 = 'D:\Diplom2021\Famous people\Johnny Depp'
listdir5 = os.listdir(path5)
my_tuple5 = os.path.split(fafa)

if my_tuple5[1] in listdir5:
    return
else:
    shutil.copy2(fafa, path5)

else:
    img = io.imread('Ryan Reynolds.jfif')

dets = detector(img, 1)

for k, d in enumerate(dets):
    print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
    shape = sp(img, d)

face0 = facerec.compute_face_descriptor(img, shape)
a = distance.euclidean(face, face0)
if a <= 0.55:
    self.label_2.setText("<html><head/><body><p align=\"center\"><span
style=\" font-size:10pt; font-weight:500;\">Это Райан
Рейнольдс</span></p></body></html>")
path4 = 'D:\Diplom2021\Famous people\Ryan Reynolds'
listdir4 = os.listdir(path4)
my_tuple4 = os.path.split(fafa)

```

```

if my_tuple4[1] in listdir4:
    return
else:
    shutil.copy2(fafa, path4)

else:
    img = io.imread('Lauren Cohan.jpg')

    dets = detector(img, 1)

    for k, d in enumerate(dets):
        print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
            k, d.left(), d.top(), d.right(), d.bottom()))
        shape = sp(img, d)

        face0 = facerec.compute_face_descriptor(img, shape)
        a = distance.euclidean(face, face0)
        if a <= 0.55:
            self.label_2.setText("<html><head/><body><p align=\"center\"><span
style=\" font-size:10pt; font-weight:500;\">Это Лорен
Коэн</span></p></body></html>")
            path3 = 'D:\Diplom2021\Famous people\Lauren Cohan'
            listdir3 = os.listdir(path3)
            my_tuple3 = os.path.split(fafa)

            if my_tuple3[1] in listdir3:
                return
            else:
                shutil.copy2(fafa, path3)

```

else:

```
img = io.imread('Katheryn Winnick.jfif')
```

```
dets = detector(img, 1)
```

```
for k, d in enumerate(dets):
```

```
    print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
        k, d.left(), d.top(), d.right(), d.bottom()))
```

```
    shape = sp(img, d)
```

```
face0 = facerec.compute_face_descriptor(img, shape)
```

```
a = distance.euclidean(face, face0)
```

```
if a <= 0.55:
```

```
    self.label_2.setText("<html><head/><body><p align=\"center\"><span
style=\"          font-size:10pt;          font-weight:500;\">Это          Кэтрин
Винник</span></p></body></html>")
```

```
    path2 = 'D:\Diplom2021\Famous people\Katheryn Winnick'
```

```
    listdir2 = os.listdir(path2)
```

```
    my_tuple2 = os.path.split(fafa)
```

```
    if my_tuple2[1] in listdir2:
```

```
        return
```

```
    else:
```

```
        shutil.copy2(fafa, path2)
```

```
else:
```

```
    img = io.imread('Matthew McConaughey.jpg')
```

```
    dets = detector(img, 1)
```

```
    for k, d in enumerate(dets):
```

```

print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
    k, d.left(), d.top(), d.right(), d.bottom()))
shape = sp(img, d)

face0 = facerec.compute_face_descriptor(img, shape)
a = distance.euclidean(face, face0)
if a <= 0.55:
    self.label_2.setText("<html><head/><body><p
align='\"center\"'><span style='\" font-size:10pt; font-weight:500;\">Это МЭТТЮ
Макконехи</span></p></body></html>")
    path1 = 'D:\Diplom2021\Famous people\Matthew McConaughey'
    listdir1 = os.listdir(path1)
    my_tuple1 = os.path.split(fafa)

    if my_tuple1[1] in listdir1:
        return
    else:
        shutil.copy2(fafa, path1)
else:
    self.label_2.setText("<html><head/><body><p
align='\"center\"'><span style='\" font-size:10pt; font-weight:500;\">Я не знаю кто
это</span></p></body></html>")
    face1 = None
    face2 = None

def open_image2(self):
    file_name = QtWidgets.QFileDialog.getOpenFileName()[0]
    if not file_name:
        return

```



```
 pixmap = QtGui.QPixmap(file_name)
 self.label1_2.setPixmap(pixmap)

 img = io.imread(file_name)

 dets = detector(img, 1)

 for k, d in enumerate(dets):
     print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
         k, d.left(), d.top(), d.right(), d.bottom()))
     shape = sp(img, d)

 global face1
 face1 = facerec.compute_face_descriptor(img, shape)

 def open_image3(self):
     file_name = QtWidgets.QFileDialog.getOpenFileName()[0]
     if not file_name:
         return

     pixmap = QtGui.QPixmap(file_name)
     self.label1_3.setPixmap(pixmap)

     img = io.imread(file_name)

     dets2 = detector(img, 1)

     for k, d in enumerate(dets2):
```

```

print("Detection {}: Left: {} Top: {} Right: {} Bottom: {}".format(
    k, d.left(), d.top(), d.right(), d.bottom()))
shape = sp(img, d)

```

```

global face2

```

```

face2 = facerec.compute_face_descriptor(img, shape)

```

```

def vert(self):

```

```

    a = distance.euclidean(face1, face2)

```

```

    if a <= 0.55:

```

```

        self.label_4.setText("<html><head/><body><p align='center'><span style='font-size:10pt; font-weight:500;'>Это один человек</span></p></body></html>")

```

```

    else:

```

```

        self.label_4.setText("<html><head/><body><p align='center'><span style='font-size:10pt; font-weight:500;'>Это разные люди</span></p></body></html>")

```

```

def search(self):

```

```

    file_name = self.lineEdit.text()

```

```

    cur = 'D:\Diplom2021\Famous people'

```

```

    listdir = os.listdir(cur)

```

```

    if not file_name in listdir:

```

```

        self.label1_4.clear()

```

```

        self.label1_5.clear()

```

```

        self.label1_6.clear()

```

```

        self.label1_4.setText('Не найдено')

```

```

        self.label1_5.setText('Не найдено')

```

```

        self.label1_6.setText('Не найдено')

```

```
        return None

    cur_dir = 'D:\Diplom2021\Famous people\\'+ file_name +'\'

    file_list = os.listdir(cur_dir)
    print(file_list)

    index1 = random.choice(file_list)
    index2 = random.choice(file_list)
    index3 = random.choice(file_list)

    fil1 = cur_dir + index1
    fil2 = cur_dir + index2
    fil3 = cur_dir + index3

    pixmap1 = QtGui.QPixmap(fil1)
    pixmap2 = QtGui.QPixmap(fil2)
    pixmap3 = QtGui.QPixmap(fil3)
    self.label1_4.setPixmap(pixmap1)
    self.label1_5.setPixmap(pixmap2)
    self.label1_6.setPixmap(pixmap3)

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Diplom = QtWidgets.QMainWindow()
    ui = Ui_Diplom()
    ui.setupUi(Diplom)
    Diplom.show()
    sys.exit(app.exec_())
```