МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____ Литвиненко О.Є.
«____»_____ 2023 р.

КВАЛІФІКАЦІНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: «Програмний модуль розпізнавання облич для систем ідентифікації осіб»

Виконавець: здобувач освіти СП-235М групи Давидова Аліна Станіславівна

Керівник: професор Тачиніна Олена Миколаївна

Нормоконтролер: Тупота Євгеній Вікторович

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій
Кафедра комп'ютеризованих систем управління
Спеціальність 123 «Комп'ютерна інженерія»
(шифр, найменування)

Освітньо-професійна програма «Системне програмування»
Форма навчання денна

ЗАТВЕРДЖУЮ
Завідувач кафедри

Литвиненко О.Є.
«____»_____202 3 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи
Давидової Аліни Станіславівни
(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема кваліфікаційної роботи: «Програмний модуль розпізнавання облич для систем ідентифікації осіб»
затверджена наказом ректора від « 28 » серпня 202 3 р. № 1494/ст

2. Термін виконання роботи: з 02.10.2023 р. по 12.12.2023 р.

3. Вихідні дані до роботи: Вимоги до розроблюваного програмного модуля, програмний засіб реалізації *VisualStudio Code*.

4. Зміст пояснювальної записки: Аналіз предметної області, застосування сіамської мережі для розпізнавання облич, розробка модуля розпізнавання облич.

5. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1. Діаграма компонентів модуля розпізнавання облич;
2. Діаграма роботи зв'язки компонентів.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1	Збір та аналіз теоретичних відомостей, визначення необхідності систем розпізнавання облич.	02.10.2023 – 13.10.2023	
2	Вивчення методів навчання та особливостей застосування сіамських мереж.	16.10.2023 – 20.10.2023	
3	Розробка архітектури мережі та навчання на базі різних архітектур (*VGG19, Inception ResNet*).	23.10.2023 – 27.10.2023	
4	Опис вимог та розробка компонентів модуля.	30.10.2023 – 03.11.2023	
5	Інтеграція компонентів, тестування продуктивності та функціональності модуля.	06.11.2023 – 10.11.2023	
6	Аналіз результатів тестування, підсумковий вибір алгоритму розпізнавання облич.	13.11.2023 – 17.11.2023	
7	Виправлення помилок, оптимізація та фінальне налагодження модуля.	20.11.2023 – 24.11.2023	
8	Підготовка опису керівництва програміста та завершення написання Розділу 3.	27.11.2023 – 01.12.2023	
9	Проходження нормоконтролю, підготовка презентації, виконання останніх корекцій та підготовка доповіді до захисту.	04.12.2023 – 08.12.2023	

7. Дата видачі завдання: «__2__»__жовтня__ 202__3__ р.

Керівник кваліфікаційної роботи _____ Тачиніна О.М.
(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ Давидова А.С.
(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Програмний модуль розпізнавання облич для систем ідентифікації осіб»: 78 сторінок, 48 рисунків, 42 літературних джерела, 1 додаток.

ПРОГРАМНИЙ МОДУЛЬ РОЗПІЗНАВАННЯ ОБЛИЧ, РОЗПІЗНАВАННЯ ОБРАЗІВ, РОЗПІЗНАВАННЯ ОБЛИЧ, ВИЯВЛЕННЯ ОБЛИЧ, МЕТОД СІАМСЬКИХ МЕРЕЖ, АРХІТЕКТУРА *VGG19*, АРХІТЕКТУРА *INCEPTION RESNET*.

Об’єкт дослідження: процес розпізнавання облич в системах ідентифікації осіб.

Предмет дослідження: програмний модуль розпізнавання облич для систем ідентифікації осіб.

Мета роботи: розробити програмний модуль розпізнавання облич з інтеграцією із сервером системи відеоаналітики.

Методи дослідження: технології розпізнавання облич, методи аналізу та проектування, написання програмного коду для модуля та його тестування.

Новизна кваліфікаційної роботи полягає в тому, що отримав подальший розвиток метод сіамської мережі для розпізнавання облич в системах ідентифікації осіб.

Практична значущість кваліфікаційної роботи. Запропонований підхід до вирішення проблеми розпізнавання облич та реалізований програмний модуль, рекомендується використовувати під час проведення наукових досліджень, а також в практичній діяльності фахівців-програмістів та відео аналітиків.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

НМ – нейронна мережа.

CNN – *Convolution Neural Network*; згорткові нейронні мережі.

ПММ – приховані Марківські моделі.

PCA – метод головних компонент.

LDA – метод лінійного дискримінанта Фішера.

AAM – *Active Appearance Models*; активна модель зовнішнього вигляду.

ASM – *Active Shape Models*; модель активної форми.

SNN – сіамська нейронна мережа.

ВСТУП

Завдання ідентифікації осіб за фотографіями завжди було однією з найважливіших цілей для дослідників, які працюють у галузі комп'ютерного зору та штучного інтелекту. З розвитком комп'ютерних технологій з'явилася низка алгоритмів для вирішення цього завдання, але більшість з них мають серйозні обмеження, пов'язані з продуктивністю і точністю розпізнавання.

Розробка «нейронних мереж» набула широкого поширення в останнє десятиліття завдяки стрімкому зростанню обчислювальних потужностей і накопиченню величезної кількості даних, необхідних для функціонування цих алгоритмів. Застосування цієї технології також привернуло увагу в галузі розпізнавання облич, де її точність була значно покращена.

Однак, незважаючи на десятиліття досліджень у багатьох провідних світових дослідницьких центрах і досягнення дуже високої точності розпізнавання, близької до людського сприйняття, не було створено жодної системи комп'ютерного зору, яка могла б працювати в реальних умовах і розпізнавати людей навіть у складних умовах.

Однією з головних проблем, з якою стикаються системи комп'ютерного зору, є велика мінливість візуального зображення через зміну освітлення, кольору, масштабу і кута спостереження. Крім того, такі фактори, як різні головні убори, аксесуари та макіяж, також ускладнюють розпізнавання облич. У деяких з цих ситуацій відрізнити одне обличчя від іншого може бути надзвичайно складно навіть для людини. Нейробіологи виявили, що існує спеціальна ділянка людського мозку, яка відповідає за розпізнавання облич. Пошук оптимального рішення для цієї задачі залишається актуальним дослідницьким викликом.

З огляду на вищесказане, зрозуміло, що це питання залишається надзвичайно актуальним і буде широко використовуватися в майбутньому в різних галузях, пов'язаних з аналізом фото- та відеоматеріалів. Зокрема, ідентифікація обличчя важлива для систем контролю доступу на підприємствах, для верифікації осіб в банківській та інших галузях. Вона також має широкий спектр застосувань, таких як управління громадською безпекою шляхом пошуку правопорушників, забезпечення

безпеки навчальних закладів шляхом моніторингу відвідуваності та ідентифікації злочинців, які можуть проникнути в приміщення.

Однак для вирішення цих завдань надзвичайно важливо зменшити кількість хибних спрацьовувань, і навіть невелике покращення точності розпізнавання було б значним досягненням і варте подальших досліджень у цій галузі.

Метою даного дослідження є створення програмного модуля, призначеного для виявлення людських облич на зображеннях, виконання різних операцій для нормалізації отриманих даних та перетворення отриманих облич у компактний формат, який може бути використаний для подальшого аналізу та обробки даних. Цей модуль має бути реалізований у вигляді *API*, що дозволяє інтегруватися з системами відеоаналізу.

Основними завданнями дослідження є:

1. Провести аналіз існуючих методів розпізнавання облич;
2. Розробити схему інтегрованої системи відеоаналітики, що інтегрується;
3. Створити набори даних для навчання нейронних мереж;
4. Провести навчання нейронних мереж на створених наборах даних;
5. Розробити інтерфейс для взаємодії з модулем розпізнавання облич;
6. Провести інтеграцію модулів з серверами системи для використання в режимі реального часу.

РОЗДІЛ 1
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Теоретичні відомості

Історія розпізнавання облич бере свій початок на початку 19 століття. Американський шериф тримав портрет жорстокого злочинця, такого як Джессі Вудсон Джеймс або Біллі Кід, з написом: «Розшукується живим або мертвим».

Перші комп'ютеризовані системи розпізнавання облич з'явилися в 60-х роках минулого століття. Вудро Вільсон (Вуді) Бледсо розробив метод класифікації облич за допомогою сітки ліній. Цей метод вимагав втручання людини і міг розпізнавати до 40 *облич на годину* (приблизно 90 *секунд на одну людину*) [18].

Як показує практика, люди не дуже добре розпізнають обличчя незнайомців. Тому технології зі штучним інтелектом повинні впоратися з цим завданням краще і швидше. Так, наприкінці 20-го століття в Рурському університеті в Бохумі було знайдено ефективне рішення. Відтоді технологія розпізнавання облич почала стрімко розвиватися. За оцінками уряду США, з 1993 по 2010 рік рівень помилок автоматичних систем розпізнавання облич зменшився більш ніж у 270 разів.

Принцип технології розпізнавання облич полягає у використанні комп'ютерів зі спеціальним програмним забезпеченням та відеокамер з вбудованим інтелектуальним розпізнаванням облич. Використання різних типів обладнання покликане вирішити одну задачу: отримання якісного зображення людського обличчя (профіль або анфас для паспортних цілей). Штучний інтелект визначає вузлові точки обличчя та вимірює відстань між ними (рис. 1.1). Залежно від технології, система може ідентифікувати до 80 вузлових точок.

Розвиток інтелектуальних систем розпізнавання облич довів, що відбитки пальців на обличчі є унікальним кодом для кожної людини і можуть зчитуватися на відстані.

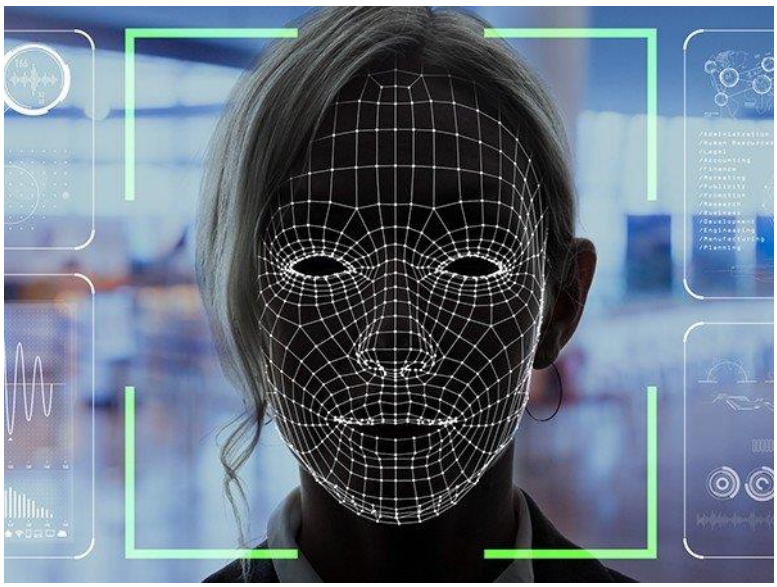


Рис. 1.1. Розпізнавання облич

Після отримання зображення програмне забезпечення порівнює фотографію обличчя з інформацією в базі даних. Швидкість зіставлення може досягати лише 1/100 *секунди*, а ймовірність помилки зведена до мінімуму (вісім помилкових спрацьовувань на 1000 сканувань обличчя).

Найпоширенішим прикладом системи розпізнавання облич є система, яка автоматично пропонує позначати друзів на фотографіях у *Facebook* та інших соціальних мережах (рис. 1.2).

Рис. 1.2. Приклад системи розпізнавання облич

Зарубіжні країни змагаються за кількість встановлених відеокамер із вбудованою функцією розпізнавання облич. Наприклад, у Китаї вже використовується понад 170 мільйонів пристроїв.

1.2. Необхідність систем розпізнавання облич

Перш за все, система розпізнавання облич використовується для ідентифікації та упіймання злочинців (на жаль, злочинців не бракує з початку 19 століття, і вони все ще переховуються).

У деяких країнах технологія розпізнавання облич використовується для притягнення до відповідальності за порушення правил дорожнього руху. Наприклад, щоб судити пішоходів, які переходять дорогу в недозволеному місці, або водіїв, які розмовляють по телефону під час руху (рис. 1.3).

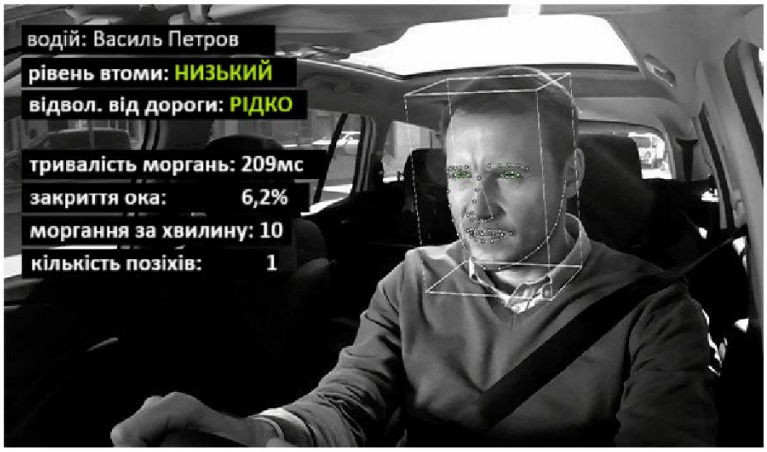


Рис. 1.3. Приклад технології розпізнавання облич

Варто зазначити, що такий підхід викликав багато суперечок щодо законності використання технології розпізнавання облич. Наприклад, Сан-Франциско стало першим містом в країні, яке заборонило використання цієї технології. Причиною позову стало втручання у приватне життя, особливо щодо темношкірих громадян США та мусульман.

«Помилки в системі можуть призвести до того, що невинні кольорові люди будуть втягнуті в поліцейські розслідування і опиняться в небезпечних для життя

ситуаціях», – говорить Метт Кегель, адвокат з питань громадянських свобод з відділення Американського союзу громадянських свобод (*ACLU*) в Північній Каліфорнії [18].

Компанія *Tencent* знайшла унікальне рішення для застосування технології розпізнавання облич у сфері охорони здоров'я. Робот аналізує симптоми захворювання з обличчя пацієнта і забезпечує відповідне лікування. Експерименти також показали, що розпізнавання обличчя ефективне для виявлення рідкісних генетичних захворювань.

В Японії технологію розпізнавання облич збираються впроваджувати в автомобілебудуванні. Мета – виявляти ознаки втоми на обличчях водіїв і запобігати аваріям.

В Україні технологія розпізнавання облич використовується переважно в корпоративному секторі. Промислові та комерційні компанії впроваджують її у свої системи безпеки. На думку експертів, великий потенціал має впровадження інтелектуального розпізнавання облич у системи контролю доступу та обліку робочого часу. Технологія дозволяє відмовитися від ідентифікаційних карток і підвищити рівень контролю за персональною відповідальністю співробітників.

Незабаром українські банки зможуть використовувати функціонал систем розпізнавання облич для підвищення безпеки платежів. Наприклад, ПриватБанк і *Visa* тестують систему *FacePay24*. Якщо цей експеримент буде успішним, то в найближчому майбутньому люди зможуть здійснювати платежі автоматично за допомогою технології розпізнавання облич, без використання гаманців, кредитних карток або навіть мобільних телефонів.

Світові експерти вважають, що кількість транзакцій, для яких можна буде використовувати технологію розпізнавання облич, буде стрімко зростати. Суперечливе питання може бути легко вирішене, якщо держави приймуть відповідне законодавство. Вони прогнозують, що ринок відеокамер з функцією розпізнавання облич досягне понад 8 мільярдів доларів США в найближчі три роки.

1.3. Сфери використання розпізнавання облич

Технологія розпізнавання облич використовується в різних сферах [20]:

- безпека в місцях масового скупчення людей;
- охоронні системи;
- запобігання проникненню в цільову зону;
- пошук зловмисників;
- управління обличчям у сфері громадського харчування та розваг;
- пошук підозрілих та небезпечних відвідувачів;
- аутентифікація банківських карт;
- онлайн-платежі;
- контекстна реклама;
- цифровий маркетинг;
- інтелектуальні та цифрові вивіски;
- фотографічне обладнання;
- криміналістика;
- телеконференції;
- мобільні додатки;
- пошук фотографій у великих фотобазах даних;
- позначення людей на фотографіях у соціальних мережах та ін.

Apple планує використовувати систему розпізнавання облич для розблокування мобільних телефонів. Вона порівнюватиме селфі, зроблені власником мобільного телефону за допомогою фронтальної камери, із заздалегідь зареєстрованими фотографіями.

Технологія розпізнавання облич може здатися футуристичною, але вона вже активно використовується в різних сферах. Ось деякі з дивовижних застосувань цієї технології [19]:

1. Безпека пристроїв.

Деякі програми використовують розпізнавання облич для захисту даних. Оскільки навіть надійні паролі не можуть захистити акаунти та інформацію від досвідчених хакерів, люди вирішили використовувати технологію розпізнавання облич. Ці програми вимагають показати обличчя, щоб розблокувати смартфон або отримати доступ до особистих даних.

2. Виявлення генетичних захворювань.

Існують спеціалізовані медичні програми, такі як *Face2Gene* та *DeepGestalt*, які використовують розпізнавання обличчя для виявлення генетичних захворювань. Обличчя аналізуються і порівнюються з базою даних осіб з різними захворюваннями.

3. Крадіжки в магазинах.

Багато магазинів обладнані системами розпізнавання облич, які ідентифікують крадіїв як небезпечних. Такі системи можуть ідентифікувати крадіїв та інформувати власників магазинів про їхні минулі злочини, навіть якщо людина ніколи раніше не відвідувала магазин. Такі системи приносять значну користь власникам магазинів, але їх ефективність часто ставиться під сумнів. Якщо невинну людину називають крадієм, це може вплинути на її життя.

4. Купівля алкоголю.

Деякі продуктові магазини та бари у Великій Британії використовують розпізнавання облич, щоб визначити, чи достатньо дорослим є покупець для придбання алкоголю. Продуктові магазини дозволяють покупцям використовувати систему самоперевірки без необхідності залучення додаткових працівників для перевірки паспортів. Якщо система визначає, що покупцеві менше 25 років, його просять пред’явити паспорт для перевірки.

5. Безпека в школах.

Розпізнавання облич також було запроваджено в школах. Одна школа у Швеції використовує *FRT* для перевірки відвідування занять. Школи в США, зокрема в Нью-Йорку, почали тестувати використання технології розпізнавання облич як «системи раннього попередження» від загроз з боку сексуальних насильників та інших осіб. Технологія також може розпізнавати 10 видів зброї для запобігання актам насильства

в школах. Деякі заклади освіти також досліджують можливості використання систем розпізнавання облич для підвищення рівня безпеки на території університетів та коледжів. Більше того, інші країни почали впроваджувати цю технологію у громадські місця з метою забезпечення загальної безпеки громадян.

6. Використання в авіакомпаніях.

Такі авіакомпанії, як *Delta* і *JetBlue Airways*, використовують розпізнавання облич для ідентифікації пасажирів. Біометричне сканування обличчя не є обов'язковим, але дозволяє використовувати обличчя пасажирів як квитки, заощаджуючи час на перевірку квитків і знижуючи витрати.

1.4. Огляд існуючих програм розпізнавання облич

За останні роки системи безпеки досягли значних успіхів. Якщо раніше, щоб знайти людину, потрібно було зателефонувати в поліцію і порівняти її фотографію із зображенням з камер спостереження, то тепер всю роботу виконує програмне забезпечення для розпізнавання облич. Крім того, багато додатків, пов'язаних з фотографіями, таких як *Picasa* і *iPhoto*, тепер використовують цю функцію.

Розпізнавання облич також можна використовувати для власних цілей. Наприклад, його можна використовувати для запобігання входу в будівлю людей з чорних списків або для побудови системи контролю доступу. Далі представлено чотири найпопулярніші програми, якими може скористатися кожен [21].

1. *SecurOS FaceX*.

Ця програма позиціонує себе як система розпізнавання облич нового покоління. У ньому використовується нейронна мережа, що дозволило значно підвищити точність – найважливішою особливістю *FaceX* (рис. 1.4) є здатність працювати в широкому діапазоні умов. Це означає, що якісний аналіз можливий навіть при різному освітленні та кутах огляду. У той же час, програмне забезпечення може працювати з необмеженим розміром бази даних, надаючи величезні можливості для аналізу інформації.

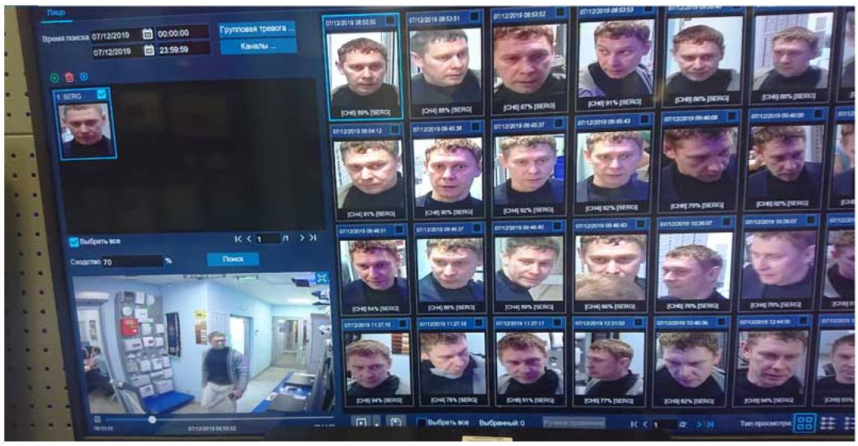


Рис. 1.4. *SecurOS FaceX*

SecurOS FaceX також легко інтегрується в системи контролю доступу. Це дозволяє зробити обличчя співробітників аналогом їхніх ідентифікаційних карток, що дозволяє їм входити в приміщення без використання рук. Важливо, що це гарантує безпеку компанії від несанкціонованого доступу. Крім того, можливість інтегрувати обличчя з іншими персональними даними створює платформу для впровадження багатофакторної автентифікації. Щоб уникнути можливих проблем, програмне забезпечення перевіряє реальність обличчя і гарантує, що будь-хто, хто намагається використати роздруковану фотографію або зображення з мобільного пристрою, отримає відмову в доступі.

Варто зазначити, що окрім створення системи розпізнавання облич та контролю доступу, *SecurOS FaceX* також допомагає в аналізі даних (рис. 1.5). Програмне забезпечення зчитує кожне обличчя і визначає стать, вікову групу, етнічну приналежність, факт використання обличчя та інші дані. Це дозволяє проводити опитування клієнтів, групуючи їх і визначаючи час доби, коли трафік є найвищим. На основі цієї інформації можна проводити маркетингові кампанії, змінювати графіки продажів і оптимізувати витрати часу співробітників.

Варто також згадати про додаткові функції, пов’язані з безпекою. Завдяки цій програмі можна створювати списки авторизованих і неавторизованих осіб і гарантувати, що небажані люди не зможуть відвідати певні приміщення. Крім того, програма може виконувати криміналістичний пошук, порівнюючи фотографії з

фотографіями людей, які з’являлися в кадрі в минулому. Всі обличчя зберігаються в базі даних нейронної мережі і можуть бути миттєво розпізнані.

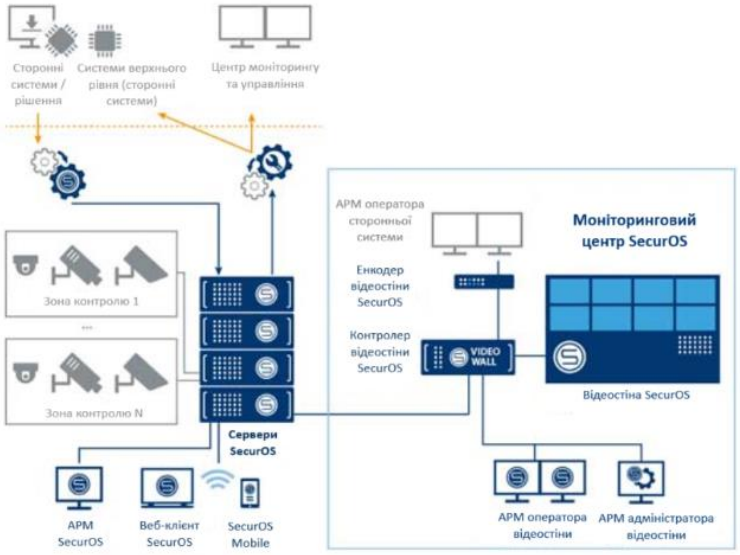


Рис. 1.5. Архітектура продукту *SecurOS FaceX*

2. *Face*-Інтелект

Ця програма широко використовується в системах громадської безпеки, на об’єктах, що потребують складного контролю доступу, і в бізнесі, де потрібна аналітична інформація. *Face*-Інтелект (рис. 1.6) виконує розпізнавання облич на відео в режимі реального часу і порівнює його з базою даних, що містить конкретних людей. Якщо особа є співробітником, їй дозволяється увійти на об’єкт; якщо виявлено порушника, сигнал надсилається до служби безпеки.

Рівні схожості встановлюються заздалегідь. Наприклад, якщо користувач є співробітником, виконується запрограмована дія, наприклад, відчинення або зачинення дверей. Таким чином, для кожної групи, зареєстрованої в базі даних, виконуються різні дії. Отримана облікова інформація може бути використана для автоматичного підрахунку робочого часу. Це дозволяє визначити ефективність працівників і відповідно до отриманої інформації винагороджувати або зменшувати їхню зарплату. Всі особи, зафіксовані камерами спостереження, вносяться до бази даних. Потім їх можна шукати за зображенням (рис. 1.7), і відображаються всі відеофайли, в яких ця людина була помічена. Це особливо корисно в місцях великого

скупчення людей, де живі співробітники не можуть впоратися з потоком інформації, створюючи загрозу безпеці.

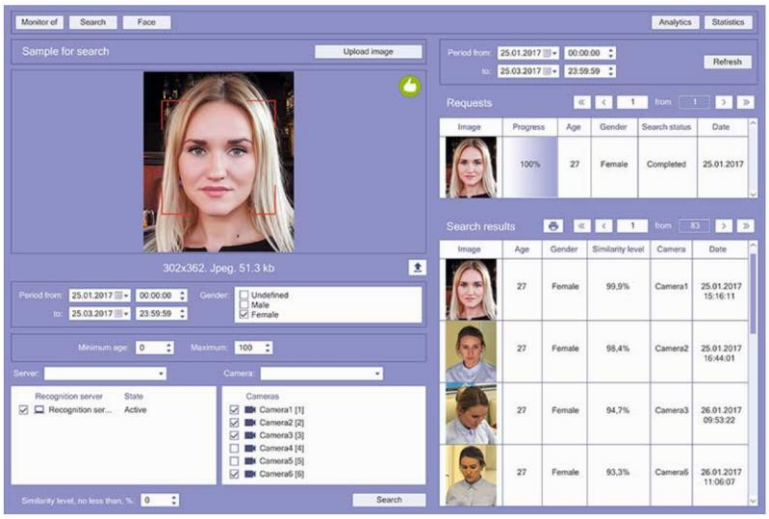


Рис. 1.6. *Face*-Інтелект

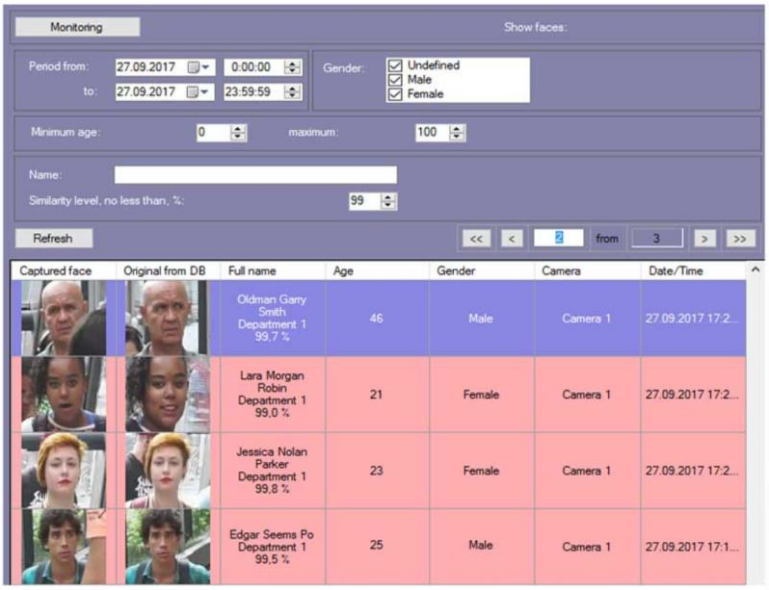


Рис. 1.7. Пошук зображення у програмі *Face*-Інтелект

Для бізнесу додаток може збирати статистику, зчитуючи кількість людей на камері, ідентифікуючи унікальних відвідувачів, визначаючи стать і вік людини та створюючи звіти. Крім того, він може попереджати про прибуття *VIP*-клієнтів і створювати статистичні звіти на основі проаналізованих даних, що дозволяє їм

будувати маркетингові стратегії. Програма також використовує три бібліотеки облич і систему аналізу.

3. *Face Director*.

Компанія *Cinezis* спеціалізується на розробці інтелектуальних систем відеоспостереження та додатків бізнес-аналітики на їх основі. Логічно, що така організація повинна мати власне рішення для розпізнавання облич, до того ж це рішення має унікальну особливість. Тобто, починаючи з першого зчитування обличчя, воно може супроводжувати людину на всьому маршруті, який вона проходить. Це гарантує, що людина завжди буде в полі зору і не пропустить жодної реакції з боку служби безпеки.

Face Director мають низку переваг. По-перше, вони мають широкий кут огляду. Тому розпізнавання обличчя продовжує працювати навіть при зміщенні початкового фокусу на 90° в горизонтальній площині і на 30° у вертикальній площині, забезпечуючи максимальну гнучкість при налаштуванні систем відеоспостереження. В інших випадках поворот камери може призвести до збою в роботі системи розпізнавання, що потенційно може вплинути на безпеку об’єкта.

Ще одна особливість *Face Detector* – здатність викликати тривогу, якщо людина навмисно намагається приховати обличчя перед камерою. У таких випадках зображення з камери негайно передається в кімнату охорони, де система продовжує стежити за ситуацією до її прояснення. Завдяки такому рішенню можна надійно запобігти вторгненню осіб з чорного списку компанії.

За словами *Cinezis*, ймовірність ідентифікації особи в базі даних становить 99%. Це означає, що вони надійно захищені від помилок системи. До цього слід додати наявність унікального рішення для бізнес-аналітики, яке, як і у випадку з попередніми програмами, може позитивно вплинути на операційну ефективність завдяки отриманню даних про клієнтів. Не слід також забувати, що можливість керувати фактичним робочим часом співробітників дозволяє оптимізувати штат.

4. *VOCORD FaceControl*.

VOCORD (рис. 1.8) використовує технологію біометричного розпізнавання облич у своїй роботі. Алгоритми та технологія обробки зображень, які використовує

компанія, були визнані найкращими у світі на конкурсі *Megaface*. За даними незалежних джерел, *FaceControl* є однією з найдосконаліших систем на світовому ринку. Точність розпізнавання облич варіюється від 90% до 100% в залежності від кута огляду камери, мінливих умов освітлення і наявності інших людей, що рухаються і перекривають огляд.

Рис. 1.8. *VOCORD FaceControl*

VOCORD FaceControl також покликаний покращити бізнес-послуги завдяки використанню аналітичної системи. Окрім розпізнавання облич, алгоритми мінімізують крадіжки в магазинах, аналізують аудиторію та оптимізують витрати на рекламу. Ключовою особливістю з боку маркетингу є можливість аналізувати поведінку покупців, що дозволяє тонко налаштовувати рекламні кампанії.

Варто зазначити, що *VOCORD* виробляє власні камери розпізнавання облич, які ідеально інтегровані в програму. Фахівці рекомендують встановлювати їх на входах до магазинів і торгових центрів. Якщо потрібно контролювати робочий час ваших співробітників, достатньо буде звичайних *IP*-камер або веб-камер. При виборі цього варіанту важливо розуміти, що використовується ряд різних апаратних і програмних засобів.

Відразу після першої активації система створює архів облич і автоматично додає туди всіх осіб, які потрапили в кадр камери спостереження. Якщо пристрій виявляє розшукувану особу, оператор кімнати безпеки отримує повідомлення протягом декількох секунд. Для забезпечення максимальної безпеки система може

здійснювати пошук за фотографією, після чого надає список файлів із зазначенням дати, місця, часу, а також імені та прізвища розшукуваної особи (якщо такі є).

5. Система «*FaceVACS*» компанії «*Cognitec Systems*»

«*FaceVACS-VideoScan*» – просте у використанні, програмне забезпечення розпізнавання облич за відео-потоком у реальному часі, що налаштовується, пропоноване компанією «*Cognitec Systems*».

Система «*FaceVACS-VideoScan*» складається з декількох системних компонентів: відео-сервера, що керує відео-потоками; сервера відео-сканування, що координує всі компоненти системи та виконує основні біометричні операції; обчислювального вузла, використовуваного для розподілу обчислювального навантаження; призначеного для користувача інтерфейсу; диспетчера сигналів, який одержує сповіщення про події та обслуговує мобільні пристрої; операційної бази даних; і комплекту інтеграторів.

На сьогоднішній день технологія *FaceVACS* використовує алгоритм розпізнавання облич *B10T9*. Цей алгоритм стійкий до змін міміки, поворотів обличчя (на ±15°), часткового його закриття, використання окулярів і зміни освітлення [34].

Крім того, система *FaceVACS* має такі особливості:

- можливість одночасного відстеження кількох облич;
- порівняння облич відбувається в реальному часі;
- можливість відображення та надсилання статистики про потоки;
- підтримка інтерактивної реєстрації з нерухомого;
- застосування *C++ API* і *Web Services API*.

6. Система «*NEC's Face Recognition*» компанії «*NEC*»

«*NEC's Face Recognition*» – одна з передових систем розпізнавання облич, розроблена японською компанією «*NEC*», що дає змогу ідентифікувати людей за кадрами багаторічної давності та навіть, якщо людина перебуває в окулярах або гримасує. Усі розпізнані обличчя зберігаються в базі даних, тому в разі потреби можна підняти всю історію відео-реєстрації та переглянути дату і час будь-якого збереженого зображення.

Технологія *NEC* перевершує безліч інших систем розпізнавання своєю точністю і швидкістю. Вона має хороші показники продуктивності в різних ситуаціях, зокрема під час роботи з відео низької якості та сильно стиснутими зображеннями. *NEC* аналізує індивідуальні особливості обличчя (розмір, форму зіниць, лінії носа і рота), їх взаємне розташування, і знаходить потім за цією інформацією відповідну людину в базі даних.

Система містить кілька модулів, що реалізують такі алгоритми:

1. Використовується метод узагальненої відповідності (*GMFD*), який забезпечує високу швидкість детектування і високу точність розпізнавання обличчя. Метод *GMFD* заснований на нейронних мережах і здійснює попередній пошук пар очей;

2. Алгоритм *PSM* (*Perturbation Space Method*), дає змогу ефективно впоратися зі складнощами, пов'язаними з розташуванням обличчя в кадрі (обличчя під нахилом або деяким кутом).

3. Метод *ARBM* (*Adaptive Regional Blend Matching*), який зменшує вплив невеликих змін на обличчі (наприклад, зміни виразу обличчя, наявність окулярів, головного убору) на точність розпізнавання [35].

Система розпізнавання облич *NeoFace* має такі особливості:

– можливість спостереження і контролю в реальному часі;

– ідентифікація на основі індивідуальних рис обличчя;

– множинне розпізнавання;

– можливість пошуку подій по базі даних;

– ведення журналу зображень облич;

– стійкість до повороту обличчя на ±15° і нахилу голови до 45° в будь-якому напрямку від фронтального положення;

– «*Drag and Drop*» управління;

– масштабований і необмежений розмір Бази Даних;

– незалежне розпізнавання напрямку погляду і характеристик обличчя (окуляри, борода і вираз обличчя).

7. Система «*VeriLook SDK*» компанії «*Neurotechnology*»

«*VeriLook SDK*» – технологія ідентифікації облич, розроблена компанією «*Neurotechnology*». Являє собою систему виявлення облич з можливістю одночасного множинного розпізнавання людей, присутніх у кадрі, і швидкої ідентифікації облич (знаходить до 100000 *облич за секунду*). *VeriLook SDK* доступна у вигляді комплекту для розроблення ПЗ і підтримує широкий вибір пристроїв на *Windows Linux*, *Mac OS X*, *iOS* і *Android* [36].

Алгоритм *VeriLook* реалізує локалізацію обличчя з використанням алгоритмів оброблення цифрових зображень, що базуються на глибоких нейронних мережах. Основними перевагами системи *VeriLook* є відсутність необхідності контакту із засобами сканування та швидке впровадження функцій біометричної ідентифікації в прикладні системи замовника, а також існує низка інших переваг:

– одночасне опрацювання кількох облич;

– гендерна класифікація. За бажанням, стать може бути визначена для кожної людини на зображенні;

– живе розпізнавання обличчя. *VeriLook* визначає, чи є обличчя у відео-потоці «живим» чи фотографією;

– розпізнавання емоцій. *VeriLook* аналізує шість основних емоцій: гнів, огиду, страх, щастя, печаль і здивування;

– атрибути обличчя. *VeriLook* може бути налаштовано для виявлення певних атрибутів під час вилучення обличчя – посмішки, відкритого рота, закритих очей, окулярів, бороди або вусів;

– визначення якості зображення обличчя. Поріг якості може використовуватися під час реєстрації обличчя, щоб гарантувати, що в базі даних зберігатимуться тільки шаблони найкращої якості;

– кілька зразків одного і того ж обличчя. Ці зразки можуть бути зараховані з різних джерел і в різний час, що дає змогу поліпшити якість зіставлення;

– ідентифікаційна здатність. Функції *VeriLook* можуть використовуватися в зіставленні 1-до-1 (перевірка), а також у режимі 1-ко-множині (ідентифікація);

– малий шаблон обличчя. Шаблон обличчя може бути розміром від 4 *кілобайт*;

– особливості режиму узагальнення. Цей режим генерує колекцію узагальнених функцій обличчя з декількох зображень одного й того самого об’єкта;

– алгоритм *VeriLook* здатний зіставляти грані, які були захоплені в інфрачервоному спектрі.

1.5. Область відповідальності учасника

Модуль розпізнавання облич є важливим компонентом вбудованої системи відеоаналізу. Він відповідає за ефективне розпізнавання та ідентифікацію облич у потоці відеоданих. Модуль розпізнавання включає в себе реалізацію спеціальних алгоритмів, інтеграцію з модулями розпізнавання облич, розробленими іншими учасниками, і забезпечує ефективну взаємодію з сервером системи.

На структурі системи, зображеній на рисунку 1.9, модуль розпізнавання облич виділений червоним квадратом і позначений як важливий елемент системи відеоаналізу. Завданням цього модуля є точне і швидке виявлення облич і подальша обробка ідентифікаційних даних, що є запорукою успішної роботи всієї системи.

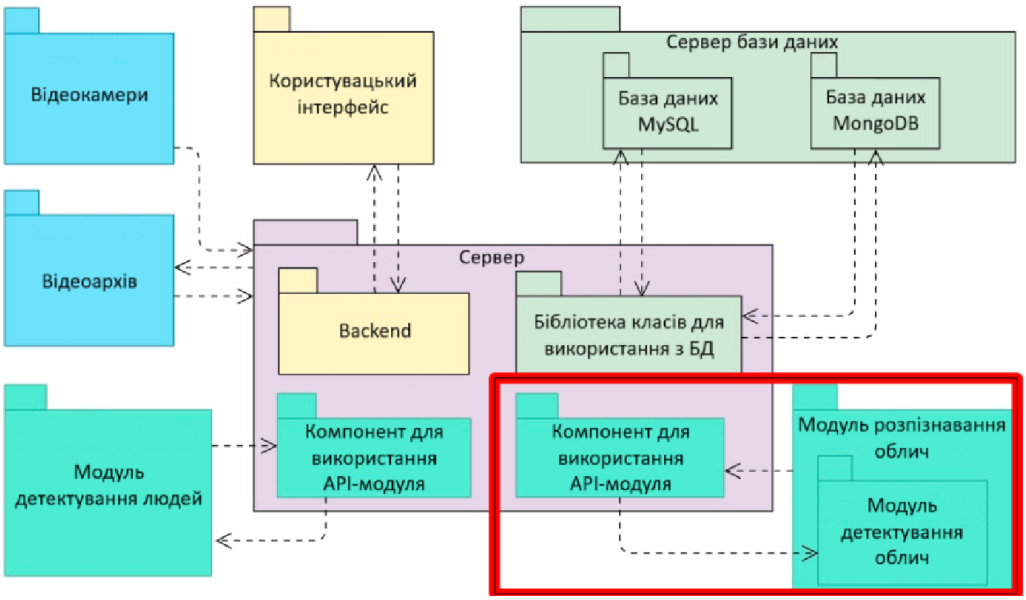


Рис. 1.9. Інтегрована система відеоаналітики, що інтегрується

1.6. Розпізнавання облич в задачах машинного зору

Задача розпізнавання людських облич є однією з найактуальніших проблем, що активно досліджуються в галузі машинного зору. Основною метою цієї задачі є встановлення відповідності між зображенням обличчя та конкретною людиною.

До основних цілей реалізації алгоритмів розпізнавання можна віднести:

1. Створення систем контролю доступу на основі біометричних даних.

2. Автоматичне виявлення злочинців і зловмисників на камерах спостереження для спрощення роботи правоохоронних органів.

3. Відіграють важливу роль у забезпеченні безпеки, сповіщаючи людей про несанкціонований доступ на об’єкти.

4. Вибудовуючи маршрути пересування людей і збираючи статистичні дані, вона може допомогти в різних сферах, таких як торгівля і маркетинг.

Вирішення цих завдань сприятиме автоматизації діяльності в багатьох галузях, зокрема в сфері безпеки та масштабної обробки даних. Однак слід зазначити, що це завдання є дуже складним в обчислювальному плані, про що свідчить розмір архітектури нейронної мережі, яка використовується для розпізнавання облич.

Одними з основних проблем, що ускладнюють розпізнавання облич, є:

1. Різноманітність умов освітлення та їх інтенсивності на зображенні.

2. Різноманітність кутів огляду облич на зображенні.

3. Вплив тіней, що створюються навколишніми об’єктами.

4. Наявність різних атрибутів одягу та аксесуарів.

5. Наявність або відсутність макіяжу.

6. Зміни з віком.

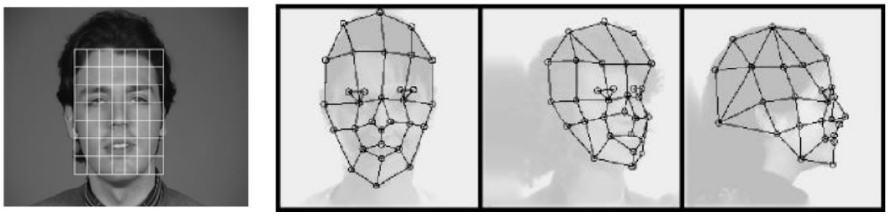
7. Наявність або відсутність вусів чи борідки.

Ці фактори значно ускладнюють процес розпізнавання обличчя і створюють значні труднощі для алгоритмів розпізнавання в цій області.

1.7. Вибір методу розпізнавання облич

1.7.1. Метод гнучкого порівняння на графах (*Elastic graph matching*)

Метод базується на еластичному зіставленні графів, що описують зображення облич [22]. Обличчя представляються у вигляді графів зі зваженими вершинами та ребрами. На етапі розпізнавання один граф (еталонний) залишається незмінним, а інший деформується так, щоб найкраще відповідати першому. У такій системі розпізнавання граф може бути або прямокутною сіткою, або структурою, утвореною характерними (антропометричними) точками обличчя (рис. 1.10).



а) б)

Рис. 1.10. Приклад структури графа для розпізнавання облич: а) регулярна решітка; б) граф на основі антропометричних точок обличчя

Ознаки обчислюються у вершинах графа, часто з використанням комплексних чисел фільтра Габора (рис. 1.11) або їх впорядкованої множини (рис. 1.12), тобто вейвлетів Габора (структур Габора).

Ребра графа зважуються відстанню між сусідніми вершинами; різниця між двома графами (відстань, дискримінативна властивість) обчислюється за допомогою певної функції вартості деформації, яка враховує як різницю значень ознак, обчислених у вершинах, так і ступінь деформації ребер графа.

Граф деформується шляхом зсуву кожної вершини графа на певну відстань у певному напрямку від її початкового положення і вибору положення, в якому різниця між значеннями ознак (відгуком фільтра Габора) вершини деформованого графа і відповідної вершини еталонного графа є мінімальною (рис. 1.13). Ця операція

виконується для всіх вершин графа по черзі до тих пір, поки сумарна різниця між значеннями ознак деформованого графа та еталонного графа не буде мінімальною. Значення функції вартості деформації в цій позиції деформованого графа є мірою різниці між вхідним зображенням обличчя та еталонним графом. Таку «розслаблену» процедуру деформації необхідно виконати для всіх еталонних графів у базі даних системи. Результатом розпізнавання системи буде еталон з найкращим значенням функції вартості деформації.

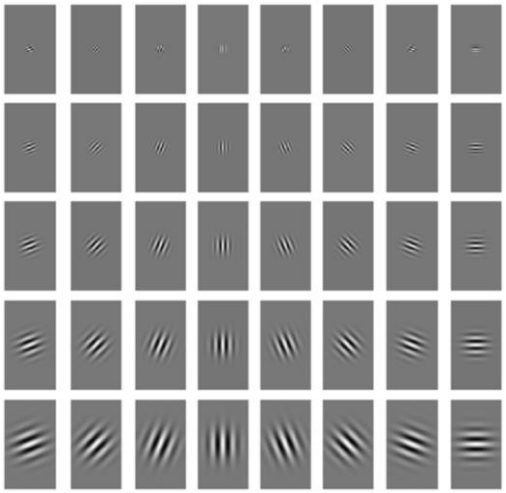


Рис. 1.11. Набір (банк, *jet*) фільтрів Габора

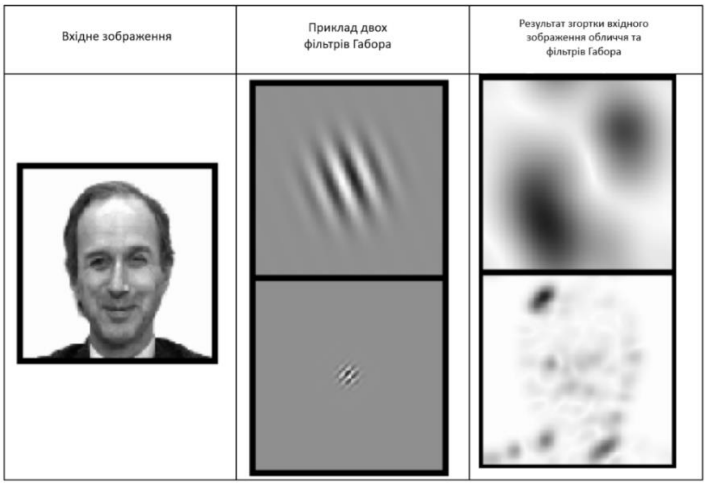


Рис. 1.12. Приклад згортки зображення обличчя з двома фільтрами Габора

Кілька публікацій показали показники розпізнавання на рівні 95-97%, навіть з різними емоційними виразами і змінами кутів обличчя до 15°. Однак розробники

системи графічного порівняння еластичності вказують на високу обчислювальну вартість такого підходу. Наприклад, для порівняння вхідного зображення обличчя з 87 еталонними зображеннями на паралельному комп'ютері з 23 трансп'ютерами було потрібно приблизно 25 *секунд* [23] (примітка: ця публікація датована 1993 роком. В інших публікаціях на цю тему час не вказується або називається великим).

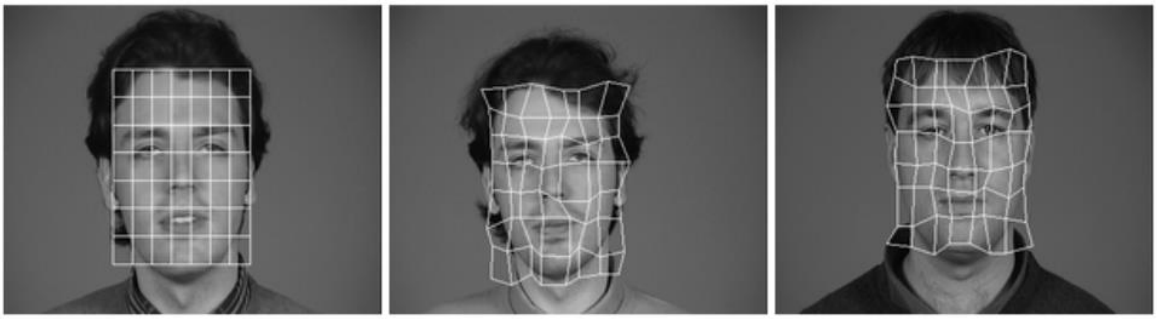


Рис. 1.13. Приклад деформації графа у вигляді регулярної решітки

Недоліки:

- процедура розпізнавання вимагає великих обчислювальних витрат;
- низька технологічність при зберіганні нових еталонів;
- лінійна залежність часу роботи від розміру бази даних облич.

1.7.2. Згорткові нейронні мережі

У сучасному суспільстві існує безліч архітектур нейронних мереж (НМ), серед яких особливо поширеними є кілька типів. Одним з них є багатошаровий персептрон, який може класифікувати певне зображення або сигнал відповідно до попереднього навчання мережі.

Нейронні мережі навчаються на навчальних прикладах, і суть навчання полягає в регулюванні ваг між нейронами під час вирішення оптимізаційної задачі за допомогою методу градієнтного спуску. Під час навчання нейронна мережа автоматично ідентифікує ключові ознаки, визначає їх важливість та будує зв'язки між ними. Очікується, що навчені нейронні мережі зможуть використовувати набутий

досвід для розпізнавання невідомих об’єктів завдяки своїм узагальненим властивостям [1].

Найкращі результати в галузі розпізнавання облич (на основі аналізу досліджень) [4] продемонстрували згорткові нейронні мережі (*Convolutional Neural Network*), які є логічним розвитком таких концепцій, як когнітрони та неокогнітрони Однією з важливих особливостей *CNN* є те, що вони можуть враховувати двовимірну топологію зображення, на відміну від багатошарового персептрона.

Важливими особливостями *CNN* є локальні рецептивні поля (які забезпечують локальні двовимірні зв’язки нейронів), загальні ваги (які можуть виявляти специфічні особливості в будь-якій частині зображення) та ієрархічна організація за допомогою просторової субдискретизації. Ці нововведення дозволяють *CNN* демонструвати часткову толерантність до змін масштабу, зсувів, поворотів, кутових змін та інших спотворень.

CNN (*Convolutional Neural Networks*) привертають увагу завдяки їхній здатності до автоматичного виявлення та ідентифікації важливих ознак без необхідності ручного вибору або наперед заданої фільтрації. Ця здатність дозволяє мережам під час навчання самостійно виявляти набори властивостей, що відрізняють класи об’єктів, або навіть визначати нові властивості, які можуть бути корисними для розпізнавання. Крім того, ці мережі виявляють високу стійкість до варіантності у вихідних даних, такі як зміни масштабу чи розташування об’єктів.

CNN виявилися дуже ефективними в обробці великих обсягів даних, таких як зображення та відео, забезпечуючи високу точність розпізнавання при адекватній швидкості обчислень. Вони знаходять широке застосування у сферах комп’ютерного зору, відеоаналітиці, медицинській діагностиці, робототехніці та інших галузях, де потрібне високоточне розпізнавання об’єктів на зображеннях чи відео.

Однією з переваг *CNN* є їхня здатність до паралельної обробки вхідних даних за рахунок використання згорткових шарів та пулінгу, що дозволяє зменшити кількість параметрів мережі. Це робить їх ефективними у роботі з великими обсягами даних, а також дозволяє прискорити процес тренування.

Завдяки інтеграції методів підвищення швидкості навчання, таких як аугментація даних та оптимізація архітектури мережі, *CNN* можуть забезпечувати високу точність і відмінні результати навіть у випадку обмеженості вхідних наборів даних.

Принцип роботи згорткової нейронної мережі показано на рисунку 1.14.

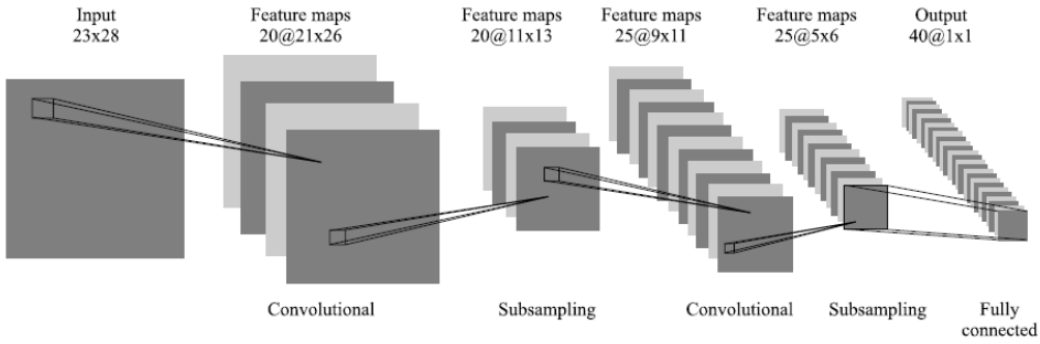


Рис. 1.14. Схематичне зображення архітектури згорткової нейронної мережі

Прикладом такої архітектури є *DeepFace*, розроблена компанією *Facebook* для розпізнавання облич користувачів. Проте всі деталі цієї архітектури є запатентованими. Схема роботи *DeepFace* зображена на рисунку 1.15.

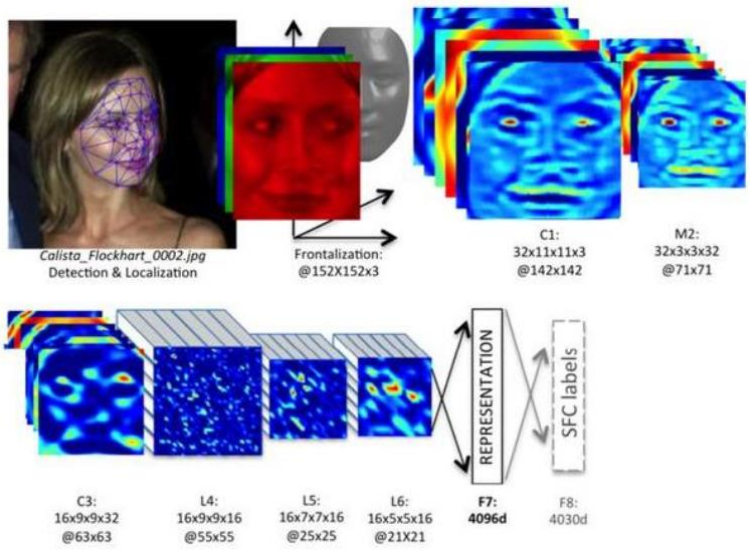


Рис. 1.15. Принцип роботи DeepFace [1]

Перевагою *CNN* є висока точність розпізнавання обличчя, навіть за наявності змін в освітленні, масштабі та просторовому повороті.

До недоліків *CNN* відносяться:

1. При додаванні нового обличчя в базу даних мережа потребує повного перенавчання на всій наявній вибірці (ця процедура може зайняти від однієї години до декількох днів).

2. Складно сформулювати на етапі вибору архітектури мережі (кількість нейронів, шарів, характер зв’язків), яку, можливо, доведеться змінювати при зміні розміру бази даних.

3. Неможливість видалення раніше навчених зображень, оскільки може знадобитися повне перенавчання мережі.

Незважаючи на ці обмеження, згорткові нейронні мережі призвели до значних проривів у розпізнаванні облич та обробці зображень.

1.7.3. Приховані Марківські моделі (ПММ)

Одним із статистичних методів розпізнавання облич є дискретна прихована Марківська модель (ПММ) [24]. ПММ використовують статистичні властивості сигналу і безпосередньо враховують його просторові властивості. Елементами моделі є набір прихованих станів, набір спостережуваних станів, матриця ймовірностей переходу та ймовірності початкових станів [25]. Кожному з них відповідає своя Марківська модель. Під час розпізнавання об’єктів перевіряються Марківські моделі, згенеровані для заданої бази даних об’єктів, і шукається максимальна ймовірність того, що послідовність спостережень даного об’єкта згенерована відповідною моделлю [26].

На сьогоднішній день не знайдено жодного комерційного застосування ПММ для розпізнавання облич.

Недоліки:

– параметри моделі необхідно підбирати для кожної бази даних;

– ПММ не мають дискримінативної здатності. Це означає, що алгоритм навчання лише максимізує реакцію кожного зображення на свою модель і не мінімізує реакцію на інші моделі.

1.7.4. Метод головних компонент (*PCA*)

Цей метод визнаний одним з найкращих алгоритмів для зменшення розмірності даних з мінімальною втратою інформації. Метод головних компонент спочатку був запропонований в статистиці, але широко застосовується в задачах розпізнавання облич.

У контексті розпізнавання облич алгоритм в основному використовується для перетворення зображення обличчя в компактний вектор, який потім порівнюється з еталонним вектором, що зберігається в базі даних [8].

Основна мета методу головних компонент – значно зменшити розмірність простору ознак, щоб цей простір найкраще відображав «типові» зображення, що належать багатьом обличчям.

Алгоритм працює наступним чином. Спочатку весь навчальний набір облич перетворюється на загальну матрицю даних, де кожен рядок відповідає одному зображенню обличчя, розгорнутому в один рядок. Всі обличчя в навчальному наборі повинні бути стандартизовані до однакового розміру і мати нормалізовані гістограми [8].

На рисунку 1.16 показано блок-схему алгоритму, що ілюструє його етапи та логіку роботи.

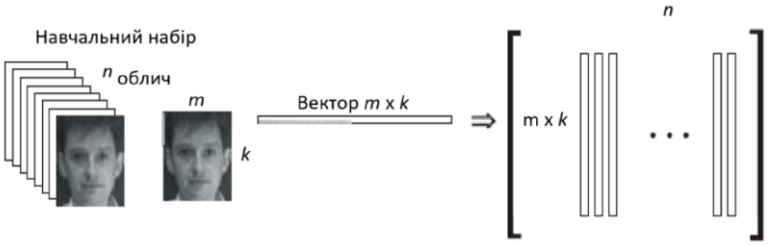


Рис. 1.16. Перетворення навчального набору осіб в одну загальну матрицю [8]

Наступним кроком є нормалізація даних і стандартизація рядків до рівня, коли середнє значення дорівнює нулю, а дисперсія – одиниці. Потім обчислюється коваріаційна матриця. Маючи отриману коваріаційну матрицю, розв’язується задача знаходження пов’язаних з нею власних значень та власних векторів (власних зображень). Наступним кроком є сортування власних векторів у порядку спадання власних значень, залишаючи лише перші *k* векторів (рис. 17).

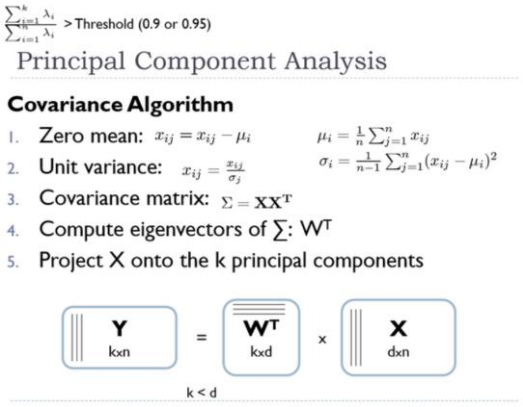


Рис. 1.17. Алгоритм *PCA*

Щоб краще зрозуміти цей процес, необхідно розглянути деякі приклади власних векторів та їх використання при побудові граней, як показано на рисунках 1.18-1.21.

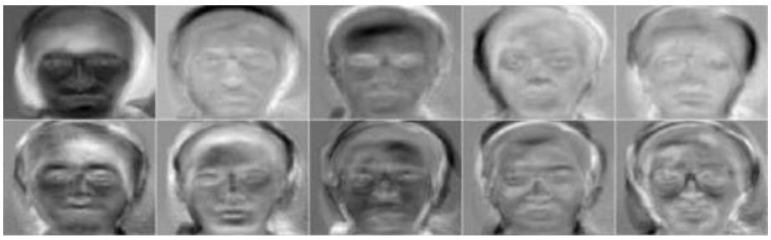


Рис. 1.18. Приклад перших десяти власних векторів [8]

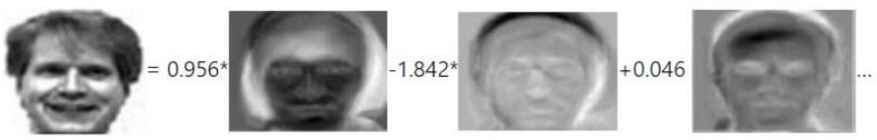


Рис. 1.19. Побудова людського обличчя за допомогою головних компонент[8]

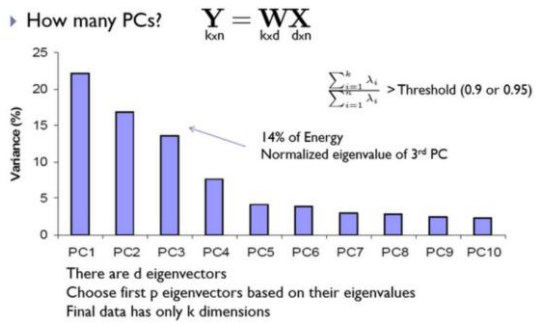


Рис. 1.20. Принцип вибору базису з перших найкращих власних векторів

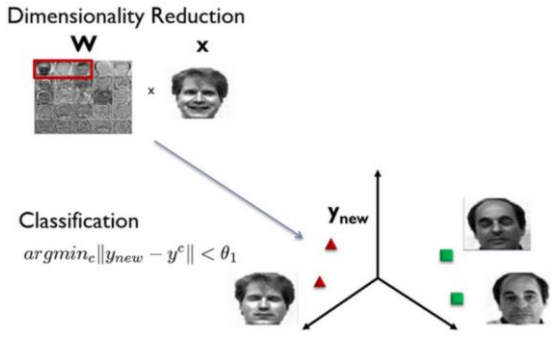


Рис. 1.21. Приклад відображення обличчя у тривимірний метричний простір, отриманому за трьома власними обличчями та подальше розпізнавання

Метод головних компонент довів свою корисність. Однак ефективність цього методу значно знижується, коли відбуваються значні зміни в освітленні або виразі обличчя на зображеннях облич. Причиною цього є те, що *PCA* вибирає підпростір не для того, щоб розрізняти класи облич, а для того, щоб максимально апроксимувати вхідний набір даних.

У статті було запропоновано вирішення цієї проблеми за допомогою методу лінійного дискримінанта Фішера (також відомого як «*Eigen-Fisher*», «*Fisherface*» або *LDA*) [27]. *LDA* обирає лінійний підпростір, який максимізує відношення:

$$\frac{|\Phi^{\mathrm{T}} S_{\text{ь}} \Phi|}{|\Phi^{\mathrm{T}} S_w \Phi|},$$

де $S_{\text{ь}} = \sum_{i=1}^{m} N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$ – матриця міжкласового розкиду, $S_w = \sum_{i=1}^{m} \sum_{x \in X_i} (x - \bar{x}_i)(x - \bar{x}_i)^T$ – матриця внутрішньокласового розкиду, m – число класів в БД.

LDA шукає такі проекції даних, де класи є максимально лінійно відокремлюваними (див. рис 1.22). Для порівняння, *PCA* шукає проекцію даних, яка максимізує розкид по всій базі даних облич (без урахування класів). Експериментальні результати показують, що *Fisherface* ефективний на 95% в умовах сильного затінення дна та резервуарів на зображеннях облич, порівняно з 53% для *Eigenface* [27].

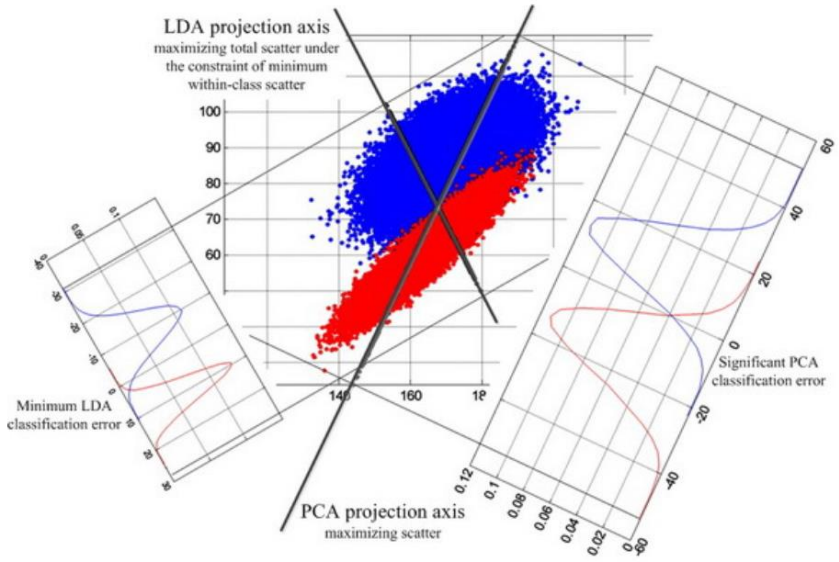


Рис. 1.22. Принципова відмінність формування проекцій *PCA* та *LDA*

Переваги методу головних компонент у розпізнаванні облич дуже очевидні:

1. Виняткова простота використання – метод дозволяє легко перетворювати зображення обличчя в компактні вектори головних компонент, що спрощує подальший аналіз і порівняння.

2. Можливість зберігати обличчя у вигляді набору коефіцієнтів головних компонент – отримані власні вектори та їх коефіцієнти можна зберігати як репрезентативний опис обличчя, що полегшує зберігання та порівняння облич у майбутньому.

3. Висока продуктивність – метод головних компонент швидкий і ефективний, що робить його ідеальним для задач розпізнавання облич у реальному часі.

Однак, на жаль, у методу є і недоліки:

1. Значна втрата ефективності через зміни в освітленні – зміни в освітленні можуть серйозно вплинути на результати розпізнавання і знизити точність цього методу.

2. Чутливість до малих поворотів і зміщень обличчя в межах досліджуваної області – цей метод може втрачати точність при малих зміщеннях і поворотах обличчя.

3. Чутливість до зашумлених даних – шуми на зображенні можуть спричинити помилки при розпізнаванні облич цим методом.

1.7.5. Активні моделі зовнішнього вигляду

Активна модель зовнішності (*Active Appearance Models*, *AAM*) – це статистична модель зображення, яка може бути адаптована до реальних зображень шляхом додавання різних перетворень. Метод був запропонований Тімом Кутсом і Крісом Тейлором у 1998 році і спочатку використовувався для аналізу облич [4].

Активна модель зовнішності містить два основних типи параметрів: параметри форми, які визначають геометричну структуру обличчя, і параметри зовнішнього вигляду, які описують текстурні властивості зображення. Для того, щоб використовувати модель, її необхідно попередньо навчити на великій кількості маркувальних зображень. Ручне маркування обличчя визначає ключові моменти, які модель повинна розпізнавати при адаптації до нових зображень.

Приклад маркування зображення обличчя наведено на рисунку 1.23.

Процес навчання *AAM* починається з нормалізації форми маркованого зображення та корекції масштабу, нахилу і зміщення. Цей процес включає так званий узагальнений проксі-аналіз. Після нормалізації з набору нормалізованих точок виділяються головні компоненти за допомогою аналізу головних компонент, приклад набору головних компонент тріангуляційної решітки наведено на рисунку 1.24.

Потім з пікселів, розташованих в межах кожного трикутника, утвореного точками форми, формується матриця текстур, що містить значення пікселів кожної

текстури. Після отримання головних компонент текстурної матриці відбувається навчання *AAM*-моделі [4].

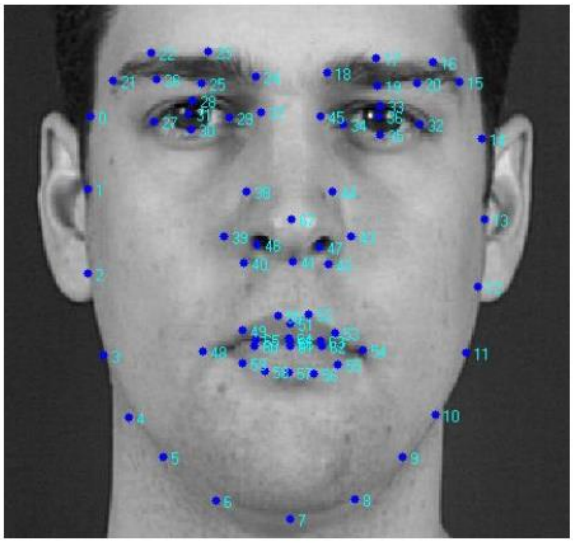


Рис. 1.23. Маркування обличчя з 68 точок

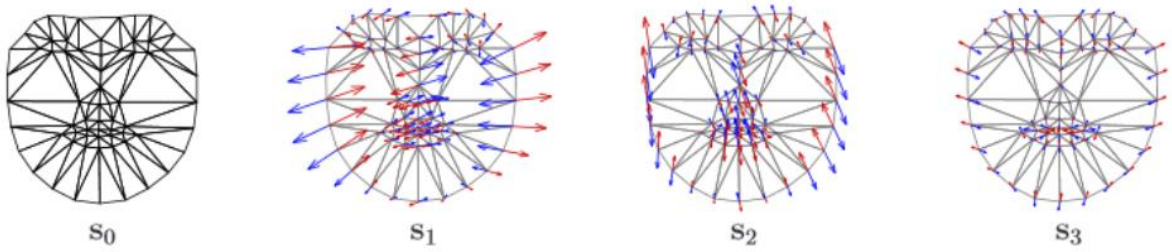


Рис. 1.24. Тріангуляційна решітка S_0, що складається з лінійних комбінацій S_i

Загальна модель складається з комбінації двох компонентів: форми та зовнішнього вигляду. Підбір цієї моделі до конкретного зображення виконується шляхом розв'язання оптимізаційної задачі мінімізації функції за допомогою методу градієнтного спуску. Параметри моделі, знайдені в цьому процесі, визначають положення моделі на конкретному зображенні.

Процес підбору модулі до конкретного обличчя представлено на рисунку 1.25.

Переваги методу *AAM* полягають у наступному:

1. Висока точність розпізнавання обличчя, навіть при наявності різних деформацій.

2. Відносно стійкий до змін освітлення завдяки фотометричній нормалізації, яка компенсує різні умови освітлення.

Рис. 1.25. Підбір моделі для конкретного зображення за 20 ітерацій градієнтного спуску

До недоліків цього методу можна віднести:

1. Необхідність попереднього навчання моделі на великій кількості мічених зображень.

2. Необхідність багатьох ітерацій алгоритму градієнтного спуску для збіжності моделі зовнішнього вигляду на проаналізованих зображеннях.

1.7.6. Моделі активної форми

Суть методу *Active Shape Models* (*ASM*) полягає у врахуванні статистичних зв’язків між положеннями точок вимірювання тіла людини (рис. 1.26). Використовується зразок зображення анфас людини [28]. Експерт відмічає на зображенні положення точок вимірювання тіла людини [29]. На кожному зображенні точки нумеруються в однаковому порядку [30].

Для приведення координат усіх зображень до єдиної системи координат зазвичай виконується так званий узагальнений прокрустів аналіз, де всі точки

відцентровуються в одному масштабі. Далі обчислюється середня форма і коваріаційна матриця всього зображення. На основі коваріаційної матриці обчислюються власні вектори, які сортуються в порядку спадання відповідних власних значень; модель *ASM* визначається матрицею Φ і вектором середньої форми $\bar{s}$.

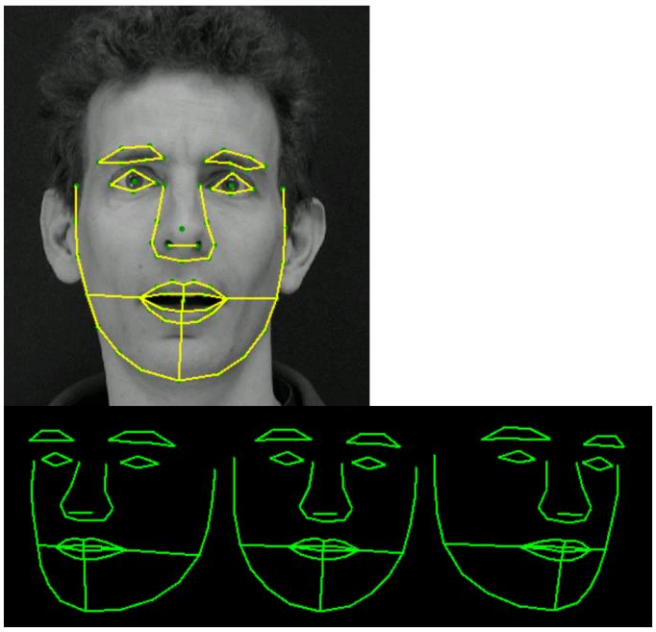


Рис. 1.26. Приклад представлення форми обличчя з використання 68 точок

Модель і параметри можуть бути використані для опису довільних фігур:

$$b_i = \Phi^T \bar{s}_i = \Phi^T (s_i - \bar{s}).$$

Локалізація моделі *ASM* для нових зображень, які не входять до навчальної вибірки, виконується в процесі розв'язання задачі оптимізації (рис. 1.27).

Однак основною метою *AAM* і *ASM* є не розпізнавання облич, а точна ідентифікація точок вимірювання обличчя і тіла людини на зображенні для подальшої обробки.

Це означає або вирівнювання зображення обличчя у фронтальному положенні відносно камери, або вирівнювання набору облич в єдиній системі координат. На цьому етапі необхідно визначити антропометричні точки на зображенні, які є

спільними для всіх облич (найчастіше це центри зіниць або кути очей). Різні дослідники виділяють різні групи таких точок. Для зменшення обчислювальних витрат систем реального часу розробники виділяють не більше 10 таких точок [31].

а) б) в) г)

Рис. 1.27. Ілюстрація процесу локалізації моделі *ASM* на конкретному зображенні: а) початкове положення; б) після 5 ітерацій; в) після 10 ітерацій; г) модель зійшлася

Моделі *AAM* та *ASM* призначені для точної ідентифікації цих антропометричних точок на зображенні обличчя.

1.7.7. Сіамські мережі

Сіамські мережі – це цікавий тип архітектури нейронних мереж, який навчається розрізняти та ранжувати вхідні дані за схожістю. Основна ідея полягає в тому, щоб навчити мережу розрізняти схожі та відмінні об’єкти [2].

Сіамська мережа складається з двох ідентичних нейронних мереж, кожна з яких має однаковий набір ваг. Кожна з цих мереж обробляє вхідні об’єкти і перетворює їх у стиснутий вектор ознак. Ці вектори ознак містять числові представлення об’єктів і можуть визначати подібності та відмінності між об’єктами [3].

Під час навчання сіамська мережа отримує пари вхідних даних разом з мітками, що вказують на їхню схожість або відмінність. Мережа оптимізує ваги, щоб збільшити схожість між схожими об’єктами та зменшити схожість між різними об’єктами [2].

Сіамські мережі поєднують в собі переваги згорткових нейронних мереж, які довели свою ефективність у розпізнаванні образів, і в той же час усувають деякі недоліки традиційних мереж класифікації.

Загальна структура і структура використання згорткової сіамської мережі представлені на рисунках 1.28-1.29.

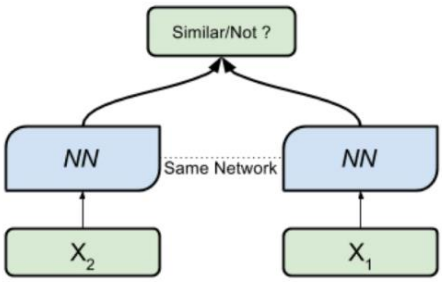


Рис. 1.28. Структура сіамських мереж [9]

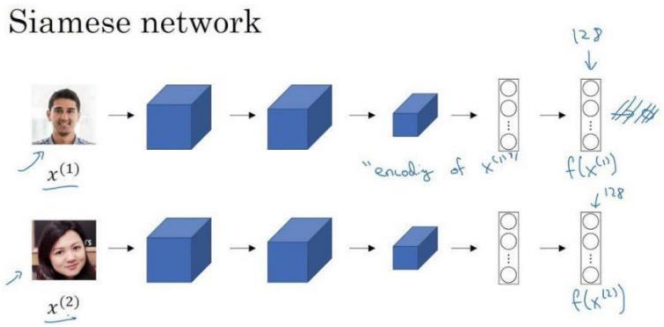


Рис. 1.29. Структура використання згорткової сіамської мережі [2]

Переваги сіамських мереж полягають у наступному:

1. Висока точність розпізнавання завдяки використанню згорткових нейронних мереж.

2. Низька чутливість до змін умов освітлення та геометричних переміщень об’єктів.

3. Навчити мережу можна лише один раз, тому немає необхідності в перенавчанні для подальшого використання.

4. Можливість зберігати інформацію про обличчя у вигляді стислих векторів ознак. Функції помилок дозволяють порівнювати об’єкти шляхом обчислення евклідової відстані між векторами ознак.

Однак вона має наступні недоліки:

1. Поріг евклідової відстані потрібно визначати експериментально для перевірки ідентичності об’єктів.

2. Потреба у великих вибірках даних, що містять зображення різних об’єктів та їх відмінностей.

Сіамські мережі є потужним інструментом для задач ранжування та порівняння об’єктів і широко використовуються в області розпізнавання облич, пошуку схожості та багатьох інших задач.

1.8. Підсумковий вибір методу

На основі аналізу найпоширеніших методів розпізнавання облич можна зробити висновок, що найкращим вибором для реалізації модулю відеоаналізу є використання сіамських мереж. Таке рішення обґрунтоване кількома факторами. В контексті побудови ефективної системи відеоспостереження існує актуальна потреба в компактному зберіганні інформації про обличчя. Крім того, нейронні мережі, особливо сіамські, показали найвищу точність розпізнавання облич порівняно з іншими алгоритмами.

Сіамські мережі демонструють високу стійкість до змін у середовищі через їхню здатність ефективно відокремлювати обличчя від фону навіть при різнобарвних умовах освітлення та камерах різної якості. Вони також проявляють високу ефективність у реальному часі, що дозволяє використовувати їх у вимогливих до швидкості системах безпеки та відеоспостереження. Одним з ключових переваг є їхня здатність впевнено розпізнавати особи навіть за наявності різних емоцій, змін у зовнішності та облич, зокрема при зміні зачіски або волосся.

Ці мережі дозволяють побудовувати системи, які можуть адаптуватися до нових зразків та відтінків облич, роблячи їх більш універсальними та гнучкими у реальних умовах експлуатації. Також важливо відзначити, що сіамські мережі дозволяють збільшити стійкість до перешкод, таких як зміни у взутті, очках чи аксесуарах, що забезпечує більш відмінну точність розпізнавання. Їхня архітектура

дозволяє навчання на малих обсягах даних, що може бути критичним в умовах обмеженої кількості наявних вибірок облич.

Потужність і адаптивність сіамських мереж у поєднанні з швидкістю та точністю створюють оптимальні умови для впровадження в різноманітних сферах, від систем безпеки до побудови інтелектуальних систем управління доступом. Розвиток цієї технології в майбутньому може сприяти покращенню безпеки та ефективності в багатьох галузях життєдіяльності.

Таким чином, виходячи з перерахованих вище переваг і успішних результатів, сіамські мережі є найкращим вибором для розпізнавання облич.

1.9. Висновки по розділу

В даному розділі було проведено детальний аналіз предметної області розпізнавання облич для систем ідентифікації осіб. Аналіз був спрямований на встановлення необхідності та актуальності розробки програмного модуля.

У розділі підкреслено важливість систем розпізнавання облич у сучасному суспільстві та їх роль у різних сферах, таких як безпека, контроль доступу та відеоспостереження. Було описано існуючі програмні рішення для розпізнавання облич, а також їхні переваги та недоліки.

Аналіз також включав дослідження різних методів і алгоритмів розпізнавання облич, включаючи гнучке порівняння графів, згорткові нейронні мережі, приховані Марківські моделі, методи головних компонент, активні моделі зовнішності, активні моделі форми і сіамські мережі. Цей аналіз сприяв вибору найкращих алгоритмів розпізнавання облич для подальшої розробки.

Таким чином, даний розділ забезпечив важливу теоретичну і практичну основу для подальших досліджень і розробки програмних модулів розпізнавання облич для систем ідентифікації осіб.

РОЗДІЛ 2
ЗАСТОСУВАННЯ СІАМСЬКОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ

У галузі машинного навчання глибокі нейронні мережі зарекомендували себе як дуже потужний інструмент для вирішення широкого спектру завдань. Однак досягнення високої точності при використанні глибоких мереж вимагає великої кількості даних для кожного класу, що може бути проблематичним у деяких сценаріях.

Для певних завдань, таких як розпізнавання обличчя, перевірка підпису, не завжди легко отримати великі обсяги даних для ефективного навчання глибоких мереж. У цих випадках використовується спеціальна архітектура нейронної мережі, відома як сіамська нейронна мережа. Цей тип мережі складається з 2 однакових підмереж з однаковою вагою.

Сіамська нейронна мережа (SNN) – це ефективний алгоритм, який використовується, коли кожен клас має обмежений обсяг даних. Це використовується при одноразовому навчанні, коли існує обмежена кількість навчальних даних, але цього достатньо для точного прогнозування.

Сіамська нейронна мережа (рис. 2.1) оптимально працює з невеликою кількістю зразків кожного класу та незбалансованим розподілом класів і демонструє високу точність. Їх основний принцип полягає у вивченні функції подібності. Вони можуть вирішити проблему визначення подібності зображень і класифікувати нові класи даних без перенавчання мережі.

Ці мережі спеціалізуються на визначенні подібності між двома вхідними об’єктами та вимірюванні ймовірності того, що вони належать до одного класу. Сіамська нейронна мережа базується на двох однакових підмережах, кожна з яких обчислює характеристики вхідних даних та оцінює подібність ознак за допомогою функції втрат.

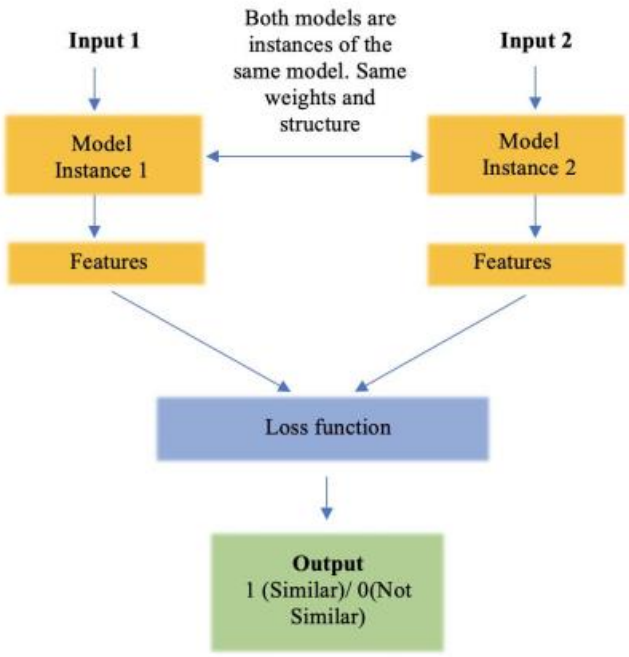


Рис. 2.1. Структура сіамської нейронної мережі

Навчання такої мережі спрямовується на мінімізацію відстані між об’єктами одного класу та збільшення відстані між об’єктами різних класів (рис. 2.2). Для цього використовуються різні подібні функції, такі як втрата контрасту та триплетна втрата.

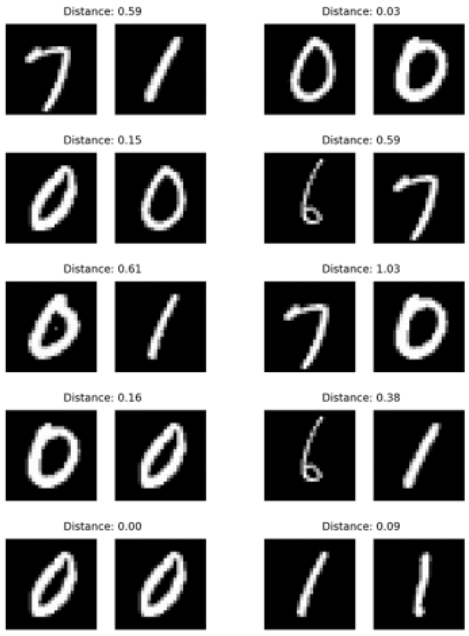


Рис. 2.2. Зображення з найменшими відстанями передбачається, що належать до одного класу, з більшими – до різних

2.1. Контрастна функція втрат

Contrastive Loss (рис. 2.3) – це функція втрат на основі відстані. У цьому випадку розглядаються пари зображень. Вона дає змогу розрізняти чи є два вхідні зображення схожими чи ні. Контрастна втрата вимагає пари позитивних і негативних навчальних даних. Метою цієї функції втрат є те, щоб евклідова відстань між представниками одного класу була меншою, ніж між парою з різних класів. Функція втрат має такий вигляд:

$$L = Y \cdot D^2 + (1 - Y) \cdot max\{margin - D, 0\}^2,$$

де D – евклідова відстань, що визначається як $D = \sqrt{\left(\left(G(x_1) - G(x_2)\right)^2\right)}$, $margin > 0$ – параметр, що дозволяє відокремлювати різні класи, $Y = \begin{cases} 1, \text{якщо зображення } x_1, x_2 \text{ з одного класу,} \\ 0, \text{якщо } x_1, x_2 \text{ з різних класів.} \end{cases}$.

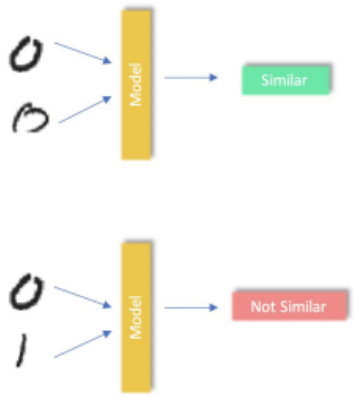


Рис. 2.3. Результат застосування *Contrastive loss*

Нейромережа штрафується за віддаленість одне від одного зображень x_1, x_2, якщо ці зображення насправді схожі. Аналогічно, виникає штраф за близькість зображень, які не належать до одного класу.

2.2 Функція втрат триплетів

Triplet Loss (рис. 2.4) – функція втрат, яка бере до уваги три об'єкти – якір, позитивний (схожий на якір) і негативний об'єкт (відмінний від якоря). Це найбільш часто використовувана функція втрат у завданнях визначення схожості. Тут робиться наголос на мінімізацію відстані від якоря до позитиву і максимізацію відстані від якоря до негативу (рис. 2.5). Математично цю функцію втрат можна записати таким чином:

$$L(A, P, N) = \max\{0, D(A, P) - D(A, N) + margin\},$$

де A – якір, P – позитив (зображення з того самого класу, що й A, N – негатив (зображення з будь-якого класу, відмінного від класу якоря), *margin* – параметр, що дає змогу уникнути збіжності до тривіального рішення. Оскільки модель може навчитися створювати однакове кодування для різних зображень (тобто відстані дорівнюватимуть нулю), з цієї причини додається гіперпараметр *margin*, щоб завжди був розрив між A і P порівняно з A і N. Так триплетна функція втрат відповідає за те, щоб різнорідні пари були віддалені від подібних пар, принаймні, на значення *margin*.

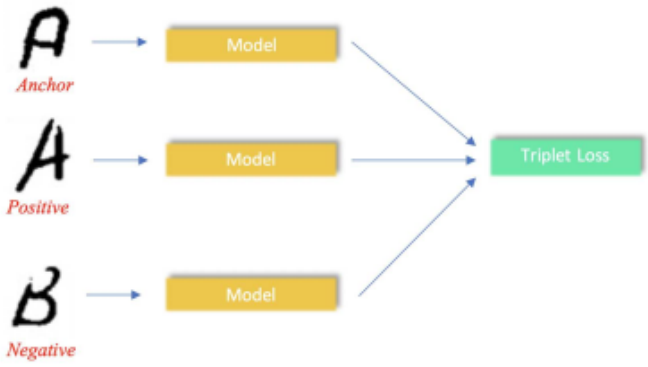


Рис. 2.4. Приклад *Triplet loss*

Рис. 2.5. Навчання функції втрат *Triplet loss* для отримання правильного кодування

Основною перевагою цієї функції втрат над контрастною втратою є те, що *Triplet loss* не має побічного ефекту, що полягає в необхідності кодувати якір і позитивні зразки в одній і тій самій точці векторного простору, як при застосуванні контрастної втрати. Це дозволяє втраті триплетів допускати деяку внутрішньокласову дисперсію, на відміну від контрастної втрати, оскільки остання зводить відстань між якорем і будь-яким позитивним значенням практично до 0.

Іншими словами, втрата триплетів дає змогу розтягувати кластери таким чином, щоб включати викиди, водночас, як і раніше, забезпечуючи достатній запас між зразками з різних кластерів.

2.3. Принцип роботи сіамської мережі

Сіамські нейронні мережі – важливий тип штучних нейронних мереж, що складається з двох ідентичних підмереж, з’єднаних спільним набором ваг. Така архітектура дозволяє представляти вхідні об’єкти у вигляді компактних векторів, які потім можна порівнювати для визначення схожості та відмінностей між об’єктами. Основна ідея сіамських мереж полягає в тому, щоб зменшити відстань між схожими об’єктами та максимізувати відстань між різними об’єктами [2].

Завдяки специфіці функції помилки, вектори, згенеровані базовою сіамською архітектурною блоковою мережею, можна легко порівнювати за допомогою евклідових відстаней. Це дозволяє встановити ступінь схожості між об’єктами та певний поріг, який визначає, коли можна вважати, що розглянуті об’єкти ідентичні [9].

Базові мережі, які є частиною сіамської архітектури, можуть використовувати різні типи нейронних мереж залежно від поставленого завдання. Наприклад, згорткові нейронні мережі часто використовуються для задач розпізнавання зображень.

Таким чином, сіамські мережі є потужним інструментом для різноманітних задач порівняння об’єктів і широко використовуються в сучасних системах розпізнавання облич, системах біометричної ідентифікації та багатьох інших сферах.

2.4. Метод навчання сіамських мереж

Для навчання сіамської мережі використовується спеціальна функція втрат, яка називається *TripletLoss*. Основна мета цієї функції – мінімізувати відстань між «якірним» зображенням та зображеннями зі схожими ознаками (позитивними) і максимізувати відстань між «якірним» зображенням та зображеннями без схожих ознак (негативними).

Фунція помилки *TripletLoss* [2]:

$$Loss = \sum_{i=1}^{N} \left[\left\| f_i^a - f_i^p \right\|_2^2 - \left\| f_i^a - f_i^n \right\|_2^2 + \alpha \right]_+,$$

де f_i^a якір – вихідне порівнюване зображення, f_i^p – схоже обличчя (позитивне), f_i^n – несхоже обличчя (*negative*), α – константа, яка дає змогу бути впевненим, що мережа не намагатиметься оптимізувати ваги безпосередньо $f_i^a - f_i^p = f_i^a - f_i^n = 0$, [...]+ еквівалентено *max(0, sum)*.

Навчання фіктивних мереж з функцією *TripletLoss* відбувається за допомогою градієнтного спуску, який схожий на процес навчання звичайної нейронної мережі. Однак важливо зазначити, що сіамська мережа має одну підмережу, яка використовується для порівняння двох зображень. Таким чином, базова мережа навчається в два етапи, розглядаючи перше і друге зображення [2].

Під час навчання з функцією *TripletLoss*, сутність використання трьох зображень полягає у тому, щоб нейронна мережа вчилася робити висновки не лише про подібність двох зображень, а й про їх відмінність від третього. Цей процес сприяє створенню унікального векторного представлення для кожного зображення, яке враховує взаємозв'язок між ними та їхні характеристики. Такий підхід робить сіамські мережі досить ефективними у вирішенні завдань розпізнавання облич та інших візуальних задач.

2.5. Особливості застосування сіамських мереж у сфері розпізнавання облич

Сіамські нейронні мережі, завдяки своїй унікальній архітектурі, розпізнають об'єкти з високою точністю та стійкістю до змін в умовах зйомки. Ця універсальність дозволяє їм працювати з великою надійністю в умовах змінного освітлення, різних кутів огляду та варіацій положень об'єктів на знімках. Особливість полягає в наявності відповідного і репрезентативного навчального набору даних, а також методів покращення якості зображень [9].

Після завершення навчання, сіамська нейронна мережа перетворюється на готову до використання модель. Це відрізняється від традиційних мереж класифікації, де потрібно багато фотографій об'єктів з різних ракурсів для додавання нового обличчя в систему. Такий метод класифікації зазвичай потребує постійного навчання мережі в робочому режимі, що зумовлено нестачею даних та архітектурними особливостями [5].

Сіамські мережі пропонують більш ефективний варіант, не вимагаючи великої кількості даних для ефективного навчання. Це пов'язано з методами збільшення обсягу навчальних даних за рахунок створення різноманітних комбінацій зображень, що відбувається на основі основних принципів роботи цих мереж. Такий підхід дає можливість суттєво зменшити вимоги до великих обсягів навчальних даних та водночас підтримувати високий рівень точності [8].

Додатково, сіамські мережі забезпечують унікальну можливість працювати з лише однією фотографією об’єкта для включення його в систему, що різко відрізняє їх від класичних алгоритмів розпізнавання. Це дозволяє суттєво спростити процес додавання нових елементів до системи ідентифікації [3].

Значною перевагою сіамських мереж є їхні здатності до роботи з обмеженими об’ємами даних без втрати точності. Це досягається завдяки їх внутрішнім механізмам роботи, спрямованим на максимально ефективне використання наявної інформації та здатності генерувати нові дані з наявних [7].

Такий підхід суттєво робить сіамські мережі більш гнучкими та адаптивними до різних умов роботи та потреб користувачів. Завдяки цьому, вони знаходять широке застосування в сферах, де обмеженість даних є основним фактором [4].

Окрім цього, сіамські мережі дозволяють уникнути складнощів, пов’язаних з перенавчанням та додатковою настройкою архітектури під час навчання мережі на нових даних. Це дає можливість максимально ефективно використовувати наявний набір даних для роботи мережі [2].

Останнім важливим аспектом є висока швидкість роботи сіамських мереж. Це досягається за рахунок їхньої здатності працювати з невеликим обсягом навчальних даних та високої точності розпізнавання. Це робить їх особливо привабливими для задач реального часу та вимогливих умов роботи [6].

2.6. Створення навчаючої вибірки

Щоб створити навчальну вибірку для цього дослідження, необхідно було об’єднати різноманітні набори даних, що знаходяться у вільному доступі. Оскільки проект призначався для комерційного використання, використання існуючих великих датасетів, призначених для змагань, не було дозволено через обмеження комерційного використання.

В результаті успішного злиття було зібрано та об’єднано понад 400 000 зображень облич. В середньому для кожного обличчя в навчальному наборі було зібрано близько 50 зображень.

Далі важливо було обробити зображення. Це було пов’язано з тим, що більшість зображень були половиною або повним тілом людини. Тому було розроблено та застосовано наступні технологічні стеки: *Python* та *MTCNN*.

Бібліотека *MTCNN* використовує точний нейромережевий алгоритм для високоточного розпізнавання облич. За допомогою цієї бібліотеки було написано скрипт для обрізання вхідного зображення та створення нового зображення, обрамленого контурами обличчя; *MTCNN* був досить точним, але мав обмеження у швидкості. Тому навчальний набір був розділений на чотири частини і оброблявся паралельно на різних комп’ютерних пристроях.

Було витрачено приблизно 18 годин на обробку даних. В результаті цього було зібрано 400 000 зображень облич різних людей і організовано в структуру зберігання.

2.7. Розробка архітектури мережі

Для першого прототипу архітектури базової мережі було вирішено використати згорткову нейронну мережу *VGG19* – 19-шарову модель, яка є покращеною версією *VGG16*, що успішно здобула перше місце на конкурсі з розпізнавання зображень *ILSVRC* [6].

Схема цієї мережі показана на рисунку 2.6.

Однак варто зазначити, що ця архітектура розрахована на розпізнавання 1000 класів різних об’єктів. Оскільки задача має справу з обличчями, було вирішено замінити лінійний шар цієї мережі на власний.

Деталі цієї лінійної частини показано на рисунку 2.7.

У цьому випадку на вхід мережі подавалася матриця розміром 224×224×3, з трьома колірними каналами для зображень відповідної розмірності. На виході мережі був набір із 128 значень. Ця кількість була визначена на основі аналізу існуючих мереж, таких як *FaceNet*.

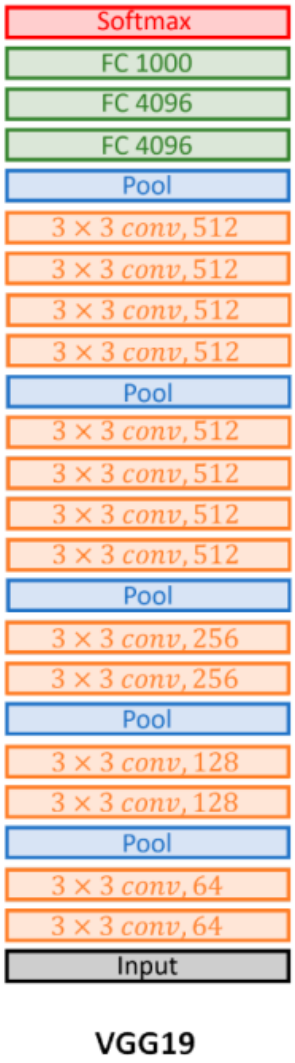


Рис. 2.6. Архітектура *VGG19* [6]

Рис. 2.7. Лінійна частина архітектури мережі

Останній шар *VGG19* був замінений лінійним шаром, оскільки дані у вихідному шарі вже були згенеровані з урахуванням мережі. Потім дані були нормалізовані таким чином, щоб вектори мали однакову довжину. Таким чином, можна зробити висновок, що нелінійне перетворення у вихідному шарі не є необхідним у архітектурі.

2.8. Навчання мережі на базі архітектури *VGG19*

Використаний стек технологій:

1. *Python*.

2. *Keras*.

3. *Tensorflow*.

Бібліотека *TensorFlow* є однією з найпопулярніших бібліотек машинного навчання з відкритим вихідним кодом, що демонструє високу продуктивність завдяки широкому інструментарію та можливостям *GPU* [15].

Keras є додатковою бібліотекою до *TensorFlow*, яка була обрана через простоту використання, а також її потенціал для перетворення моделей у загальнодоступний формат представлення нейронних мереж *ONNX*.

Оскільки *Keras* не має вбудованої реалізації функції помилки *TripletLoss*, цю функцію було створено за допомогою *TensorFlow* та інструментів *Python* [15].

Навчання мережі проводилося на основі набору даних, описаного вище, при цьому кожна пара зображень компілювалася, і мережа навчалася протягом 130 епох, поки значення функції помилки не стабілізувалося.

Як метод оптимізації навчання було обрано метод *Adam*, а гіперпараметри швидкості навчання змінювалися від 0.001 до 0.00001 протягом усіх епох [5].

Для навчання використовувалося наступне обладнання:

1. Ноутбук *ASUS ROG GL503VS*.

2. Процесор *Inter Core I7-7700HQ* 3.8 *ГГц*.

3. ОЗУ 16 *ГБ*.

4. Відеокарта *NVIDIA GeForce GTX* 1070 8*GB*.

Загальний час навчання склав приблизно 60 *годин* для трьох різних модельних екземплярів з дещо зміненими гіперпараметрами. У даній роботі описано параметри найточнішого екземпляра навченої мережі.

2.9. Тестування мережі на базі архітектури *VGG19*

Для ефективного тестування мережі необхідно було встановити оптимальний поріг відстані, при якому точність розпізнавання обличчя є найвищою. Поріг – це відстань між векторами, за межами якої система вважає, що обличчя належить іншій людині.

Для оцінки точності розпізнавання облич використовувався індекс, який визначається як відношення кількості правильних прогнозів до загальної кількості прогнозів, зроблених мережею.

Для визначення оптимального порогу було розроблено генетичний алгоритм на основі мови програмування *Python*. Алгоритм автоматично підбирав оптимальний поріг, який максимізував точність розпізнавання. Результати експерименту показано на графіку на рисунку 2.8.

У цьому випадку найкраща точність розпізнавання склала 72% при пороговому значенні 0,6236. Точність розпізнавання на тестовому наборі з 10 000 зображень, які не використовувалися під час навчання, склала 63%. Для повноцінної роботи модуля потрібна точність понад 90%. Ця цифра була обрана як оптимальне значення, враховуючи можливість використання обробки зображень для підвищення точності та можливість покращення розпізнавання облич за рахунок багаторазового аналізу відеопотоку.

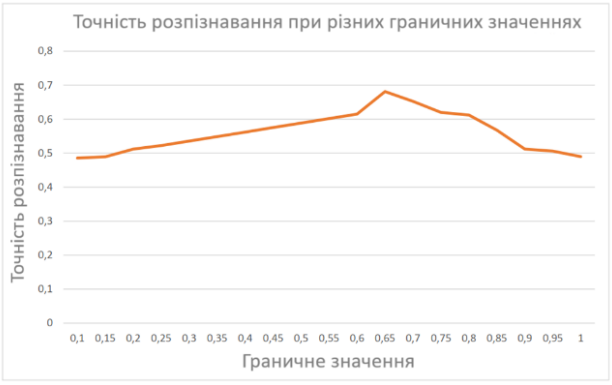


Рис. 2.8. Точність розпізнавання за різних порогових значень мережі на базі архітектури *VGG19*

За результатами навчання можна зробити висновок, що мережа потребує перенавчання та покращення точності розпізнавання: мінімальна точність розпізнавання, яка можлива за допомогою цієї мережі, становить 50%, якщо враховувати ймовірність вгадування одного з двох варіантів.

2.10. Створення моделі мережі з використанням навченої моделі *Inception ResNet*

Ретельний аналіз архітектури моделей для розпізнавання облич став ключовим кроком у вдосконаленні точності цього процесу. Попередні дослідження свідчать, що використання архітектури VGG19 має певні обмеження у досягненні високої точності розпізнавання. Більшість вдосконалених моделей використовують глибокі архітектури з елементами глибокого залишкового навчання, що значно покращує результати розпізнавання.

Обрана для подальшого дослідження архітектура Inception ResNet з 152 шарами відображена на рисунку 2.9 і відзначена своєю складною структурою. Однак через обмежені обчислювальні ресурси навчати цілу мережу виявилося неможливим. Таким чином, вирішено було використати згорткову частину попередньо навченої моделі, замінивши лінійну частину, та подальше її уточнення за рахунок наявних навчальних даних.

Підхід полягав у використанні мережі для обробки зображень розміром 160×160×3, що на виході генерувала набір із 128 значень, аналогічний моделі, побудованій на основі VGG19. Важливо відзначити, що цей підхід дозволив отримати покращення точності розпізнавання облич у порівнянні зі стандартними підходами.

Переконливо відобразити архітектуру та зберегти значення параметрів попередньо навченої моделі для подальшої її уточнення було ключовим етапом в цьому процесі. Це дозволило значно економити час та зберігати потенціал моделі для високоякісного розпізнавання облич в умовах обмежених обчислювальних ресурсів.

Результати цього підходу показали значні покращення у точності розпізнавання облич та ефективності моделі, забезпечуючи більш точне та швидке визначення

облич у порівнянні з попередніми варіантами. Такий підхід виявився більш оптимальним та продуктивним у вигляді точності та ресурсоємності.

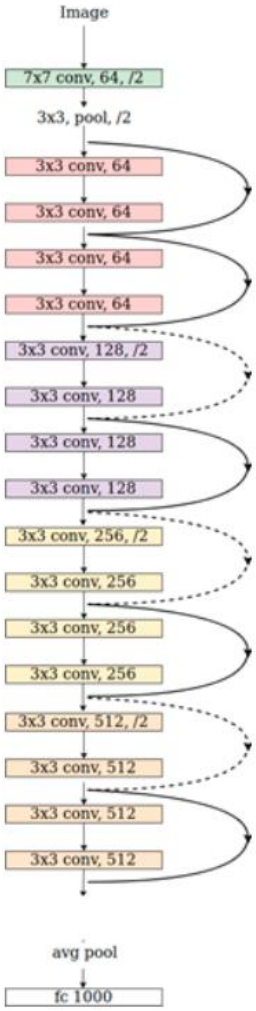


Рис. 2.9. Архітектура *Inception ResNet* [13]

Враховуючи специфіку ресурсів та складності обчислень, вибір даної моделі став оптимальним компромісом між точністю та обчислювальною ефективністю. Цей підхід дозволяє забезпечити задовільну точність при використанні обмежених обчислювальних ресурсів.

Ретельна робота над вдосконаленням моделі та відбір оптимальних параметрів відіграв важливу роль у підвищенні точності та швидкості розпізнавання облич. Такий підхід дозволив отримати значне покращення результатів на виході моделі при обмежених обчислювальних ресурсах.

2.11. Навчання мережі на базі архітектури *Inception ResNet*

Для навчання цієї мережі використовувався той самий стек технологій та апаратне забезпечення, що й для попередньої моделі на основі архітектури *VGG19*. Навчання проводилося на заздалегідь підготовленому наборі даних.

Для оптимізації процесу навчання використовувався метод *Adam* з початковою швидкістю навчання 0,002, яка поступово зменшувалася до 0,0001 під час навчання [10].

Критеріями завершення навчання були або досягнення дуже низького значення функції помилки, близького до нуля, або досягнення 95% точності на невеликій тестовій вибірці з 10 000 зображень, які не використовувалися для навчання.

Для визначення оптимального порогу було використано раніше розроблений алгоритм. Результати цього експерименту показано на рисунку 2.10.

Згідно з цим дослідженням, максимальна точність розпізнавання досягла 92% при порозі 0,7334. Така точність вважається задовільною для використання мережі в рамках розробленого програмного комплексу. Однак на практиці ця точність може бути покращена за рахунок різних перетворень вхідних зображень облич.

Рис. 2.10. Точність розпізнавання за різних порогових значень мережі на базі архітектури *Inception ResNet*

2.12. Висновки по розділу

В даному розділі розглянуто важливі аспекти використання сіамських мереж для розпізнавання облич з точки зору розробки програмних модулів для систем ідентифікації облич. Основні висновки цього розділу:

1. Описано принципи роботи сіамських мереж. Сіамська мережа обробляє два обличчя за допомогою двох ідентичних гілок мережі і порівнює їх вектори.

2. Описано метод навчання сіамської мережі, включаючи використання функції *TripletLoss*.

3. Представлено ключові особливості та переваги використання сіамських мереж для розпізнавання облич. До них відносяться висока точність, можливість роботи з невеликою кількістю навчальних вибірок та ефективне порівняння облич.

4. Описано процес створення навчальної вибірки, включаючи збір та обробку зображень облич.

5. Описано розробку архітектури мережі на основі сіамської моделі та вибір архітектури *VGG19* для навчання.

6. Представлено процес навчання мережі на основі архітектури *VGG19*, включаючи вибір методу оптимізації, налаштування гіперпараметрів та визначення критеріїв завершення навчання.

7. Оцінено результати тестування мережі на основі архітектури *VGG19*. Це включає визначення точності та оптимального порогу розпізнавання.

8. Описано процес створення мережевої моделі з використанням навченої моделі *Inception ResNet* для підвищення точності розпізнавання облич.

9. Представлено процес навчання мережі на основі архітектури *Inception ResNet*, включаючи вибір методу оптимізації та досягнення цільової точності.

Загалом, в даному розділі надано вичерпну інформацію про використання сіамських мереж для розпізнавання облич, а також про процес розробки та навчання моделей на основі різних архітектур. Знання та результати досліджень, викладені в цьому розділі, є важливим внеском у розробку програмних модулів для систем розпізнавання облич.

РОЗДІЛ 3
РОЗРОБКА МОДУЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ

3.1. Опис вимог до роботи модуля

Модуль повинен функціонувати як окрема програма, яка приймає вхідні дані у вигляді піксельної матриці. За допомогою вбудованого модуля розпізнавання облич він аналізує вхідну матрицю, знаходить на ній зображення облич і надає їх на виході у вигляді стисненого представлення. Згідно з відомими правилами, користувач може порівняти ці вектори і визначити з певним ступенем впевненості, чи належать вони одній і тій самій людині.

Цей модуль, як окрема програма, буде відповідати за обробку вхідних даних, включаючи їх аналіз на предмет наявності облич у певній піксельній матриці. Його функціональність включатиме виявлення і подальше стиснення зображень облич для забезпечення виходу у вигляді векторного представлення, що дозволяє подальше їх порівняння.

Розробка *API* для взаємодії з модулем в межах однієї локальної мережі має на меті забезпечити зручний та ефективний спосіб комунікації з цим модулем. Це дозволить іншим системам чи програмам, що працюють у цій мережі, легко взаємодіяти з функціоналом розпізнавання облич.

Крім того, для цього модуля необхідно розробити *API*, що забезпечує можливість взаємодії з цим модулем в межах однієї локальної мережі. Для кращого розуміння принципів роботи модуля була розроблена компонентна діаграма.

Компонентна діаграма, розроблена для кращого розуміння принципів роботи модуля, створена з метою візуалізації взаємозв’язків та взаємодії окремих компонентів цієї програми. Це допомагає не лише розробникам, а й користувачам краще усвідомити основні складові та логіку роботи цього модуля.

Важливо відзначити, що забезпечення стабільності роботи цього модуля, особливо в контексті точності розпізнавання, є однією з основних мет цієї розробки.

Запровадження та оптимізація методів роботи модуля сприятимуть його ефективності та надійності у практичному використанні.

На рисунку 3.1 зображено компонентну діаграму модуля розпізнавання облич. На ній показані основні компоненти та їх взаємозв'язки, що допомагає краще зрозуміти структуру модуля та його функції.

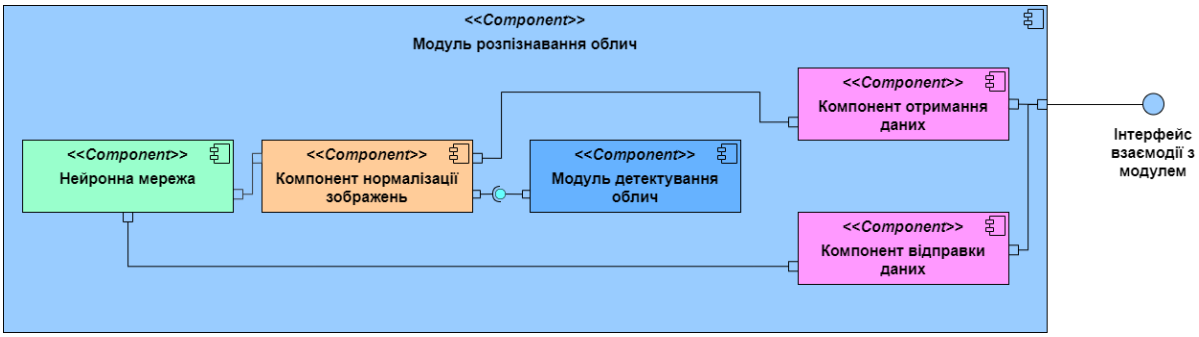


Рис. 3.1. Діаграма компонентів модуля розпізнавання облич

Завдяки взаємодії різних компонентів модуль здатний виконувати завдання пошуку та розпізнавання людських облич на зображенні, дозволяючи користувачеві впевнено визначати ідентичність облич на фотографії та забезпечуючи гнучкий спосіб взаємодії з модулем.

3.2. Реалізація компонента нормалізації зображень

Щоб підготувати зображення перед введенням його в нейронну мережу, потрібно виконати кілька важливих завдань:

1. Виявлення облич на зображенні: на цьому етапі відбувається виявлення облич на оригінальному зображенні. Можуть бути використані спеціальні модулі виявлення облич або інші алгоритми комп'ютерного зору.

2. Обрізання виявлених облич: виявлені обличчя потрібно обрізати, щоб отримати окремі фрагменти зображення, що містять людські обличчя.

3. Масштабування зображення до розмірів, необхідних мережі: вхідний розмір нейронної мережі є фіксованим. Тому важливо масштабувати обрізані обличчя до необхідного розміру.

4. Нормалізація значень пікселів: нейронні мережі зазвичай працюють зі значеннями пікселів у діапазоні від 0 до 1. Тому значення пікселів потрібно нормалізувати.

Для розв’язання першої проблеми детектування облич було включено модуль детектування облич в проект. Цей модуль надає методи для пошуку облич на зображенні та повертає координати знайдених облич.

Щоб вирішити решту завдань, був використаний такий стек технологій:

1. *Python*: Мова програмування, яка забезпечує зручний інтерфейс для обробки та обробки зображень.

2. *Pillow*: Бібліотека для роботи з зображеннями, яка надає можливості зчитування, збереження та обробки зображень у різних форматах.

3. *Numpy*: Бібліотека для обчислення математичних операцій та маніпуляції багатовимірними масивами даних.

4. *Scikit-Image*: Бібліотека для обробки зображень, яка містить інструменти для зміни розміру, обрізки, масштабування та нормалізації зображень.

З використанням цих технологій була розроблена власна бібліотека, яка містить методи для масштабування зображень зі збереженням пропорцій, зміни роздільної здатності та нормалізації значень пікселів. Ця бібліотека надає різноманітні варіанти для налаштування обробки зображень, що дозволяє досягти оптимальних результатів залежно від потреб проекту.

Такий підхід допомагає підготувати зображення для подальшого розпізнавання та забезпечити високу якість результатів.

3.3. Створення інтерфейсу взаємодії з модулем

Для забезпечення максимальної продуктивності взаємодії з модулем необхідно вирішити питання максимальної швидкості передачі даних: архітектури, що використовують *HTTP* і вимагають періодичного встановлення з’єднання, мають обмеження.

Наприклад, при використанні архітектури *REST* встановлення *TCP*-з'єднання займає від 13 до 50 *мілісекунд*, а передача двійкового файлу через *HTTP* потребує часу на конвертацію у формат *base64*. Це займає ще більше часу і збільшує обсяг переданих даних.

Наприклад, передача файлу розміром 6 220 800 *байт* на локальному комп'ютері займає 11 *мілісекунд* в локальній мережі. Однак час роботи таких компонентів, як модулі виявлення та нейронні мережі, становить приблизно 110 *мс*. Тому час, необхідний для передачі файлу через *REST*, становить 30-50% від часу, необхідного для роботи всього компонента, що є суттєвим обмеженням продуктивності.

Враховуючи ці обмеження, було вирішено розробити власний протокол з використанням *TCP* [7]. Це мінімізує інформацію, що передається, і прибирає надлишкові метадані, що надсилаються через *HTTP*. Завжди відкритий канал *TCP* готовий до зв'язку, що прискорює передачу даних.

Тримати канал *TCP* відкритим під час роботи системи є розумним, оскільки він завжди використовується для обробки потоків зображень і забезпечує оптимальну продуктивність системи.

Бібліотека *ZeroMQ* є зручною обгорткою для *TCP*-сокетів і була використана для забезпечення інтерфейсу взаємодії з модулем. Вона є зручною обгорткою для *TCP*-сокетів і може автоматично відновлювати з'єднання у разі короткочасного розриву, забезпечуючи надійний зв'язок. За допомогою мови програмування *Python* та бібліотеки *ZeroMQ* було розроблено програму для взаємодії з модулем за протоколом *TCP* .

Для передачі даних від сервера до модуля використовується специфічний формат, який містить два блоки по чотири *байти*, закодовані як цілі числа (*Int*). Формат вхідних даних представлено на рисунку 3.2.

Ці дані являють собою розміри вхідного зображення, а також блоки даних впорядкованої матриці зображення, довжина якої «ширина» помножена на «висоту», а кожен кольоровий піксель помножений на 3 *байти*.

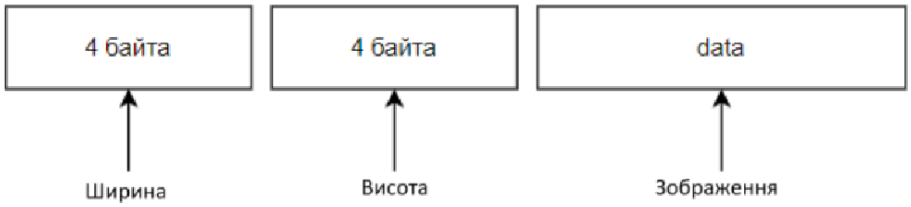


Рис. 3.2. Формат вхідних даних

Формат відповіді від модуля також має свою структуру. Він складається з серії блоків, кількість яких залежить від кількості виявлених на зображенні облич. Формат відповіді представлено на рисунку 3.3.

Рис. 3.3. Формат відповіді

Перший блок містить 4 *байти* і відображає кількість виявлених облич. Значення 0 означає, що на зображенні немає облич, тоді як -1 вказує на помилку за модулем. Якщо кількість блоків додатна, модуль повертає відповідну кількість блоків. Кожен блок має розмір 1024 *байти* і являє собою 128 значень по 8 *байт* у форматі *Double*.

3.4. Реалізація компонента отримання статистики

Цей механізм призначений для відповіді на широкомовні запити та надання поточного часу виконання одного такту модуля. Ця інформація необхідна для інформування сервера про працездатність модуля та визначення його адреси в локальній мережі.

Для вирішення задачі обробки широкомовних запитів модуль використовує бібліотеку *Socket*, яка надає засоби для роботи з протоколом *UDP*, та бібліотеку *ZeroMQ*, яка забезпечує передачу даних за протоколом *TCP*. Параметри

широкомовної адреси та порту прослуховування задаються у конфігураційному *XML*-файлі перед запуском [7, 12].

Цей компонент виконується в окремому потоці для уникнення конфліктів з основною частиною модуля; важливо зазначити, що мова програмування *Python* має певні обмеження для багатопотокових реалізацій. Зокрема, глобальне блокування інтерпретатора (*Global Interpreter Lock, GIL*) може запускати лише один потік в будь-який момент часу. Це обмеження може призвести до низької продуктивності багатопотокових програм [16].

Для подолання цього обмеження використовують механізми багатопроцесорності та багатопроцесорні бібліотеки. Багатопроцесорність дозволяє запускати окремі підпроцеси, кожен з яких має власний екземпляр інтерпретатора (зазвичай окремий процес), і надає засоби обміну даними між підпроцесами, наприклад, спільну пам’ять.

Діаграму взаємодії модуля розпізнавання зі статистичним компонентом показано на рисунку 3.4. Ця діаграма показує, як взаємодіють компоненти системи.

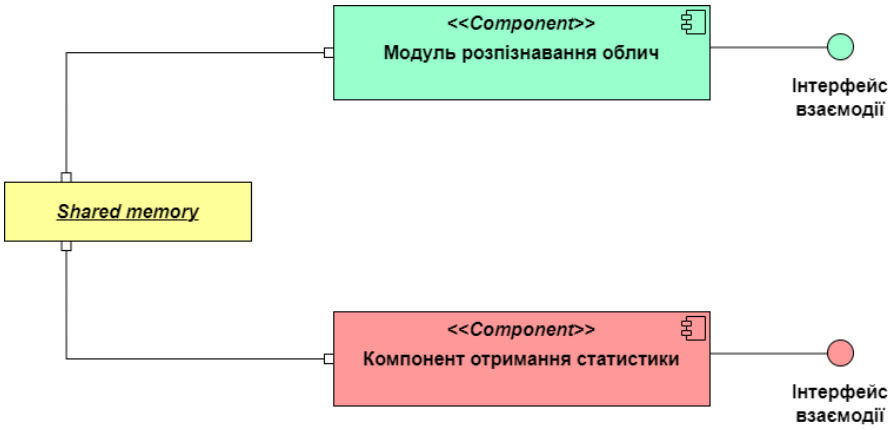


Рис. 3.4. Діаграма роботи зв’язки компонентів

У цьому сценарії компонент отримання статистики працює як підпроцес модуля розпізнавання облич. Ці компоненти мають загальну пам’ять, яку можна розділити. Компонент можна запитувати за допомогою широкомовних запитів, які відповідають формату даних, показаному на рисунку 3.5.

Формат широкомовного запиту включає перші чотири байти. Це число має бути однаковим як для конфігурації сервера, так і для конфігурації модуля. Порт, який використовується для надсилання запиту, також має бути однаковим для всіх конфігурацій. Наступні чотири *байти* використовуються для підтвердження запиту, і коли компонент надсилає відповідь, цей номер також включається. Це дозволяє ідентифікувати *IP*-адресу модуля в мережі; порти для запитів *UDP* і *TCP* однакові. Спільна пам’ять містить системний час, коли модуль розпізнавання починає обробляти інформацію (за замовчуванням це час запуску модуля). Також вона містить змінну, яка зберігає три прапори стану: 0 – очікування вхідного запиту, 1 – обробка даних, -1 – виникла помилка.

Рис. 3.5. Формат широкомовного запиту

Під час роботи модуля розпізнавання прапори стану змінюються у відповідь на різні події. Змінна, що зберігає системний час, оновлюється при кожному зверненні до модуля. Тому при опитуванні модуля відомий фактичний час, що минув з моменту початку обробки даних, і можна оцінити стан модуля. Якщо на стороні модуля виникає помилка, повертаються дані зі значенням -1, а сам модуль змінює своє значення статусу на -1. Однак у випадку некритичних помилок модуль може повернутися до початкового стану, змінивши значення стану на 0 (очікування).

3.5. Інтеграція модуля в систему

Для успішної інтеграції розроблених модулів розпізнавання облич у системний сервер, розроблений на мові програмування *C#*, було розроблено декілька компонентів для спрощення взаємодії з протоколами *TCP* та *UDP*. Основними

завданнями цих компонентів були забезпечення надійної передачі даних, синхронізація та балансування робочих процесів, а також механізм відновлення з'єднання та коректного завершення роботи в разі виходу з ладу окремого модуля системи.

На схемі інтеграції, зображеній на рисунку 3.6, показані основні компоненти, розроблені для цієї мети.

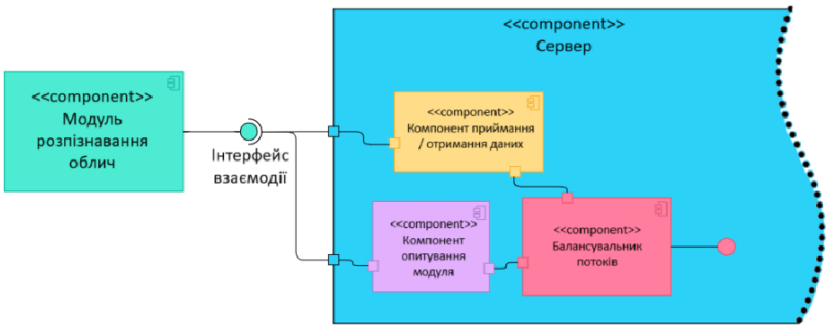


Рис. 3.6. Інтеграція

Компонент «Прийом/передача даних» містить класи, які надають методи для пакування, надсилання та отримання даних за допомогою *ZeroMQ*, забезпечуючи ефективний обмін інформацією між модулем розпізнавання облич та сервером системи.

Під час запуску система почала надсилати запити до модуля за допомогою компонента опитування, і на основі отриманих даних було ініціалізовано декілька потоків, що відповідали кількості запущених екземплярів модуля розпізнавання облич. Кожен з цих потоків використовував патерн «Виробник-Споживач» для ефективної обробки даних, що надходили з відеокамери.

Реалізація патерну «Виробник-Споживач» передбачала використання структури *BlockingCollection*, яка дозволила розподілити обробку даних між різними потоками. Це дозволило оптимально використовувати ресурси та ефективно обробляти вхідні дані [11].

При обробці даних з модулів були враховані механізми обробки помилок. При отриманні повідомлення про помилку від певного модуля відповідний потік починав

опитування модуля на наявність помилок або зупинок. У разі повторних помилок або відсутності зв’язку потік припиняв роботу, про що повідомляла подія *ModuleIsSuspend*.

За допомогою цих компонентів і механізмів модуль розпізнавання облич був успішно інтегрований в систему ідентифікації осіб.

3.6. Тестування продуктивності модуля

Для оцінки роботи модуля була створена спеціальна програма-тестер, яка може захоплювати зображення з веб-камери і перетворювати виявлені обличчя у вектори за допомогою модуля розпізнавання облич. Потім вектори порівнюються з векторами облич людей, що зберігаються в базі даних, і результати відображаються у відеопотоці. Структура компонентів цього додатку показана на рисунку 3.7.

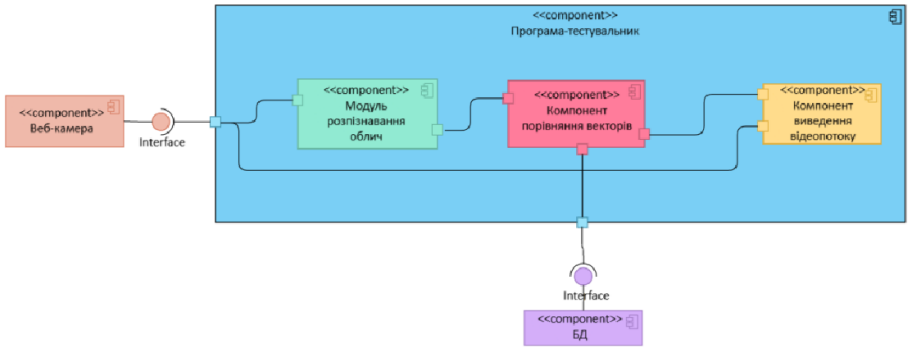


Рис. 3.7. Програма-тестер

Для створення цього додатку були використані наступні технології:

1. *OpenCV*.
2. *Python*.
3. *C#*.
4. *Numpy*.

OpenCV – бібліотека, яка забезпечує можливість захоплення та відображення зображень.

Модуль виявлення облич використовується як частина модуля розпізнавання і надає координати облич на зображенні для подальшого відображення. База даних використовується для зберігання векторів, які порівнюються для ідентифікації розпізнаних облич.

Компонент порівняння векторів реалізовано за допомогою бібліотеки *Numpy*, яка виконує всі обчислення за матрицею та вектором, дозволяючи нормалізувати вектори та швидко обчислювати евклідові відстані.

Результати тестування виводилися у відеопотоці у вигляді кадрів і тексту, а також зображень, як показано на рисунку 3.8.

Також, було проведено запис розпізнавання відеопотоку, як показано на рисунку 3.9.

Рис. 3.8. Виведення відеопотоку

```
NAME,TIME

Alina,16:33-57

Alina,16:37-11

Alina,16:37-11

Alina,17:00-33

Alina,17:04-19

Alina,17:12-15

Alina,17:12-56

Alina,17:13-20
```

Рис. 3.9. Запис розпізнавання

Для оцінки продуктивності було проведено серію тестів для вимірювання швидкості роботи програми з моменту захоплення кадру до моменту відправки вектора на компонент порівняння. Згідно з результатами тестів, середня швидкість роботи модулів на обладнанні, яке використовувалося для навчання мережі, склала 96 *мс*. Така продуктивність дозволяє виводити розпізнане зображення обличчя однієї людини зі швидкістю 11-12 *кадрів на секунду*.

3.7. Керівництво програміста

Призначення і умови застосування модуля

Модуль призначений для наочності отримання відеопотоку даних через модуль розпізнавання облич. Основними функціями модуля є: отримання обличчя з відеопотоку, запис до БД та розпізнавання у майбутньому.

Звернення до модуля

Завантажити модуль можна з флешки, жорсткого диску, відкривши папку *FR*. Файли, що знаходяться у папці необхідно відкривати за допомогою *Visual Studio Code* чи *PyCharm*.

Файл з ім’ям *add_faces.py* дає змогу запам’ятати та записати до БД нове обличчя. При відкритті файлу необхідно мати доступ до камери. Якщо файл був успішно завантажений – необхідно зачекати 100 *мс* для кращого запам’ятовування обличчя.

Файл з ім’ям *test.py* дає можливість переглядати записані в БД дані та розпізнавати обличчя. Приклад роботи можна побачити на рисунку 3.8.

Файл з ім’ям *Attendance_(date).csv* записує усі розпізнавання, що були здійснені модулем. Приклад запису можна побачити на рисунку 3.9.

Повідомлення

Програмісту може видаватися повідомлення про неможливість запуску модуля. Це може бути пов’язано з наступним:

1. Не підключено відеокамеру. Якщо доступу до камери модуль не отримав – буде виведено повідомлення до терміналу.

2. Не встановлено розширення *Python*. Без даного розширення модуль не зможе продовжити роботу. Якщо після встановлення *Python* модуль все ще не запускається – рекомендовано завантажити версію 3.6.5.54.

3. Не встановлено розширення *OpenCV Python*. Без даного розширення модуль не зможе продовжити роботу. Якщо після встановлення *OpenCV* модуль все ще не запускається – рекомендовано завантажити версію 4.5.2.

Керівництво технічного обслуговування

Для забезпечення нормальної роботи модуля повинна бути використана наступна мінімальна конфігурація комп'ютера:

– процесор *Inter Core I7-7700HQ* 3.8 *ГГц*.

– ОЗУ 16 *ГБ*.

– відеокарта *NVIDIA GeForce GTX* 1070 8*GB*.

3.8. Висновки по розділу

В даному розділі було детально розроблено програмний модуль розпізнавання облич для системи ідентифікації осіб. Експлуатаційні вимоги до описаного модуля визначили основні функціональні та технічні характеристики, які необхідно було врахувати під час розробки.

У цьому розділі описано реалізацію таких компонентів модуля, як нормалізація зображення, генерація статистики та створення інтерфейсів взаємодії. Ці кроки визначили технічні деталі реалізації та забезпечили ефективну взаємодію модуля з іншими компонентами системи.

Інтеграція розробленого модуля в систему ідентифікації осіб є важливим етапом, оскільки він визначає здатність модуля працювати в реальних умовах і взаємодіяти з існуючими елементами системи.

Тестування працездатності модуля показало його ефективність та стабільність роботи в різних умовах. Результати тестування свідчать про те, що розроблений модуль має потенціал для успішного використання в практичних задачах системи ідентифікації осіб.

Керівництво програміста містить документацію та інструкції щодо можливих проблем та мінімальну характеристику комп’ютера для кращої роботи модуля.

Загалом, даний розділ слугує для конкретизації та реалізації теоретичних концепцій, представлених у попередніх розділах цієї кваліфікаційної роботи, і надає повний огляд процесу створення та реалізації програмного модуля розпізнавання облич для системи ідентифікації осіб.

ВИСНОВКИ

У першому розділі кваліфікаційної роботи проведено аналіз літературних джерел та теоретичних даних за темою кваліфікаційної роботи, обґрунтовано необхідність створення програмного модуля розпізнавання облич для систем ідентифікації осіб та визначено сфери його використання. Проведено аналіз існуючих методів та додатків для розпізнавання облич, та обрано метод сіамських мереж для подальшого дослідження.

У другому розділі досліджено методи застосування сіамських мереж, принципи роботи та методи навчання сіамських мереж, а також проведено навчальні та тестові експерименти з використанням архітектур *VGG19* та *Inception ResNet*.

У третьому розділі розроблено програмний модуль розпізнавання облич. Розроблено вимоги до модуля, реалізовано компоненти нормалізації зображення та генерації статистики, а також створено інтерфейс взаємодії. Проведено інтеграцію модуля в систему та тестування продуктивності, що підтвердило його ефективність і готовність до практичного застосування. Розроблено керівництво програміста, що містить документацію та інструкції щодо вирішення можливих проблем під час роботи з модулем.

Тестування розробленого програмного модуля показало, що застосування сіамської мережі дозволяє отримати високі показники розпізнавання облич, а саме: точність розпізнавання – 82%, швидкість розпізнавання – 87%, що значно вищі за показники програм-аналогів.

Таким чином, мета кваліфікаційної роботи – розробка ефективного програмного модуля розпізнавання облич для систем ідентифікації осіб – досягнута.

Запропонований підхід до вирішення проблеми розпізнавання облич та реалізований програмний модуль, рекомендується використовувати під час проведення наукових досліджень, а також в практичній діяльності фахівців-програмістів та відео аналітиків.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Програмування глибоких нейронних мереж на *Python* [Електронний ресурс] / О.І. Созикін. – *URL*: *https://www.asozykin.com/courses/nnpython* (дата звернення: 16.10.2023).

2. *FaceNet* – приклад простої системи розпізнавання облич [Електронний ресурс]: *Neurohive*, 2018. – *URL*: *https://neurohive.io/ua/tutorial/raspoznavanie-lica-facenet/* (дата звернення: 18.10.2023).

3. *One Shot Learning*: розпізнавання обличчя з використанням сіамської нейронної мережі [Електронний ресурс]: *machinelearningmastery*, 2017. – *URL*: *https://www.machinelearningmastery.com/one-shot-learning-facerecognition-using-siamese-neural-network-a13dcf739e/* (дата звернення: 19.10.2023).

4. *Бішоп К.М.* Розпізнавання образів і машинне навчання/ *К.М. Бішоп*; пер. з англ. *Д.А. Клюшин*: *Вільямс*, 2020. – 960 с.

5. *Рашка С. Python* і машинне навчання/ *С. Рашка*, *В. Мірджалілі*: *Вільямс*, 2019. – 665 с.

6. *VGG16* – згорткова мережа для виділення ознак зображень [Електронний ресурс]: *Neurohive*, 2018. – *URL*: *https://neurohive.io/ua/vidy-nejrosetej/vgg16-model/* (дата звернення: 26.10.2023).

7. Протокол *TCP* [Електронний ресурс]: *Metanit*. – 2015. – *URL*: *https://metanit.com/sharp/net/4.1.php* (дата звернення: 10.11.2023).

8. *Рассел Д.* Метод головних компонент/ *Д. Рассел*, *Р. Кон*: Книга на вимогу, 2012. – 56 с.

9. Розпізнавання облич за допомогою сіамських мереж [Електронний ресурс]: *Habr*, 2019. – *URL*: *https://habr.com/ua/company/jetinfosystems/blog/465279/* (дата звернення: 25.10.2023).

10. *Чару А.* Нейронні мережі та глибоке навчання. Навчальний курс/ *А. Чару*: *Вільямс*, 2020. – 752 с.

11. *The Producer Consumer Pattern in .NET (C#)* [Електронний ресурс]: *dotnetcurry*. – 2017 р. – *URL*: *https://www.dotnetcurry.com/patternspractices/1407/producer-consumer-pattern-dotnet-csharp* (дата звернення: 13.11.2023).

12. Протокол *UDP* [Електронний ресурс]: *Metanit*. – 2015. – *URL*: *https://metanit.com/sharp/net/5.1.php* (дата звернення: 11.11.2023).

13. Еволюція нейромереж для розпізнавання зображень у *Google*: *Inception-ResNet* [Електронний ресурс]: *Habr*, 2016. – *URL*: *https://habr.com/ua/post/303196/* (дата звернення: 28.10.2023).

14. *Face Detection using MTCNN* [Електронний ресурс]: *Towards data science*, 2020. – *URL*: *https://towardsdatascience.com/face-detection-using-mtcnna-guide-for-face-extraction-with-a-focus-on-speed-c6d59f82d49* (дата звернення: 29.10.2023).

15. *Шакла Н.* Машинне навчання & *TensorFlow*/ *Н. Шакла*, 2019. – 336 с.

16. Вчимося писати багатопотокові та багатопроцесні додатки на *Python* [Електронний ресурс]: *Habr*, 2012. – *URL*: *https://habr.com/ua/post/149420/* (дата звернення: 12.11.2023).

17. *Video analytics market size* [Електронний ресурс] *URL*: *https://www.fortunebusinessinsights.com/industry-reports/video-analytics-market-101114* (дата звернення: 30.10.2023).

18. Технологія розпізнавання облич [Електронний ресурс] *URL*: *https://greenvision.ua/blog/overview/Tekhnologiya_raspoznavaniya_lits* (дата звернення: 03.10.2023).

19. Системи розпізнавання облич *Facial recognition technology* (*FRT*) [Електронний ресурс] *URL*: *https://www.tadviser.com/index.php/Стаття:Системи_розпізнавання_облич_(Facial_recognition)#cite_note-2* (дата звернення: 05.10.2023).

20. Технології розпізнавання облич або фейсконтроль по-розумному [Електронний ресурс] *URL*: *https://iot.com/gorodskaya-sreda/tekhnologii-raspoznavaniya-lits-ili-feyskontrol-po-umnomu* (дата звернення: 03.11.2023).

21. Огляд популярних програм розпізнавання облич [Електронний ресурс] *URL*: *https://videmir.pro/stat/obzor-populyarnyh-program-raspoznavania-lic/* (дата звернення: 07.10.2023).

22. *Face Recognition by Elastic Bunch Graph Matching / In Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp.* 355-396, (1999).

23. *Distortion Invariant Object Recognition in the Dynamic Link Architecture / IEEE transactions on computers, vol. 42, no. 3, march* 1993.

24. Методи розпізнавання облич на основі прихованих Марківських процесів. Автореферат / кандидат наук *Двойний Ілля Ростиславович*, 2013.

25. Застосування прихованих Марківських моделей для розпізнавання облич / Латвійська науково-технічна конференція «Інформатика і проблема телекомунікацій». Матеріали латвійської науково-технічної конференції. *Hbuf: Latvijas Universitāte*, 2006. Том I, с. 150-154.

26. *Face Detection and Recognition Using Hidden Markovs Models / Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.*98CB36269).

27. *Eigenfaces vs. Fisherfaces Recognition Using Class Specific Linear Projection / IEEE Transactions on Pattern Analysis and Machine Intelligence* (*July* 1997).

28. *Facial Recognition Using Active Shape Models, Local Patches and Support Vector Machines / NZCSRSC ’08 Christchurch New Zealand*.

29. *Face Alignment Using Active Shape Model And Support Vector Machine / International Journal of Biometrics and Bioinformatics, 2011, pp.* 224-234.

30. *Active Shape Models — Their Training and Application / Univ Manchester, Dept Med Biophys, Oxford Rd, Manchester M13 9PT, Lancs, England, Computer Vision and Image Understanding, January 1995, Pages* 38-59.

31. *Image-based Face Recognition – Issues and Methods / Wenyi Zhao, R. Chellappa*. 2002.

32. *A friendly introduction to Siamese Networks* [Електронний ресурс] / *Sean Benhur J* – Режим доступу: *https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942* (дата звернення: 05.11.2023).

33. *One-Shot Learning With Siamese Network* [Електронний ресурс] / *Renu Khandelwal* – Режим доступу: *https://medium.com/swlh/one-shot-learning-with-siamese-network-1c7404c35fda* (дата звернення: 06.11.2023).

34. Хабрахабр – найбільший ресурс для *IT*-фахівців. [Електронний ресурс]. // Режим доступу – *https://habr.com/post/133826/* (дата звернення: 09.10.2023)

35. Системи технічної безпеки та охорони «*MTI*». *FaceVACS – FaceVACS. VideoScan* [Електронний ресурс]. // Режим доступу: *http://www.security.mti.ua/products/sistemy-videonabludeniya/Soft-raspoznavanieiz/Cognitec/152-facevacsvideoscan_/* (дата звернення: 10.10.2023).

36. *VisionLabs. LUNA SDK* [Електронний ресурс]. // Режим доступу: *https://visionlabs.ai/ua/luna-platform-info.html* (дата звернення: 11.10.2023).

37. *Самсонов В.В., Сільвестров А.М., Тачиніна О.М.* Методологія наукових досліджень та приклади її використання: Навч. посібник. К.:НУХТ, 2022. – 385 с.

38. *Давидова А.С.* «Програмний модуль розпізнавання облич для систем ідентифікації осіб»: Міжнародна наукова-технічна конференція «Інтелектуальні технології лінгвістичного аналізу»: Тези доповідей. – К.: НАУ, 2023. – 64 с.

39. *Давидова А.С.* «Програмний модуль розпізнавання облич для систем ідентифікації осіб»: Наукова-технічна конференція «Сучасні тенденції розвитку системного програмування»: Тези доповідей. – К.: НАУ, 2023. – 56 с.

40. *Бойченко С.В., Іванченко О.В.* Положення про дипломні роботи (проекти) випускників Національного авіаційного університету. – К.: НАУ, 2017, 63 с.

41. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення / Держстандарт України. – Вид. офіц. – [Чинний від 1995-02-23]. – Київ, 2007. – 86с.

42. ГОСТ 2.301-68. Єдина система конструкторської документації. Формати. – Введ. 2002–01–01. – М. : Вид.-во стандартів, 2006. – 27 с.

ДОДАТОК А

Частковий лістинг вихідного коду програми

add_faces.py

```
import cv2
import pickle
import numpy as np
import os
video = cv2.VideoCapture(0)
facedetect = cv2.CascadeClassifier(‘data/haarcascade_frontalface_default.xml’)
faces_data = []
i = 0
name = input("Enter You Name: ")
while True:
    ret, frame = video.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = facedetect.detectMultiScale(gray, 1.3, 5)
…
        i = i + 1
        cv2.putText(frame, str(len(faces_data)), (50, 50), cv2.FONT_HERSHEY_COMPLEX, 1, (50, 50, 255), 1)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (50, 50, 255), 1)
…
    if k == ord(‘q’) or len(faces_data) == 100:
        break
video.release()
cv2.destroyAllWindows()
…
```

test.py

```
from sklearn.neighbors import KNeighborsClassifier
import cv2
import pickle
import numpy as np
from datetime import datetime
from win32com.client import Dispatch
def speak(str1):
   speak = Dispatch(("SAPI.SpVoice"))
   speak.Speak(str1)
facedetect = cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml')
with open('data/names.pkl', 'rb') as w:
        …
while True:
   ret, frame = video.read()
        …
   for (x, y, w, h) in faces:
      crop_img = frame[y: y + h, x: x + w, :]
      resized_img = cv2.resize(crop_img, (50, 50)).flatten().reshape(1, -1)
      output = knn.predict(resized_img)
      ts = time.time()
        …
      cv2.rectangle(frame, (x,y-40), (x+w, y), (221, 160, 221), -1)
      cv2.putText(frame, str(output[0]), (x,y-15),
cv2.FONT_HERSHEY_COMPLEX, 1, (0, 0, 0), 1)
      cv2.rectangle(frame, (x, y), (x + w, y + h), (221, 160, 221), 1)
      attendance = [str(output[0]), str(timestamp)]
      k = cv2.waitKey(1)
…
```