

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики
Кафедра інформатики

Обробка відео для розпізнавання та ідентифікації об'єктів

**Текстова частина до магістерської роботи (тези) за спеціальністю
«Інженерія програмного забезпечення»**

Виконав: студент 2-го року навчання,
спеціальності 8.05010101

Інформаційні управляючі системи та
технології

Бондар Ілля Борисович

Науковий керівник
Ковалюк Т.В.,
Доцент, к.н.

Магістерська робота захищена
з оцінкою «_____»

Секретар ДЕК _____
«_____» _____ 2021 р.

Київ 2021

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ

доцент, к.н.

_____ Ковалюк Т.В

(підпис)

8 листопада 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на дипломну роботу

Студенту Бондару І.Б факультету інформатики 2 курсу МП
Тема «Обробка відео для розпізнавання та ідентифікації об'єктів»

Зміст ТЧ до дипломної роботи:

- Індивідуальне завдання
- Вступ
- Огляд існуючих рішень
- Огляд та аналіз методів розпізнавання обличч
- Проектування і розробка власного ПЗ

Дата видачі 8 листопада 2021 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Календарний план виконання робіт

№	Назва етапу	Термін виконання	Примітка
1	Отримання завдання	08.11.20	
2	Огляд теоретичного матеріалу	15.01.21	
3	Продумування та обговорення з науковим керівником вимог до практичної частини	05.02.21	
4	Розробка застосунку	15.04.21	
5	Написання пояснювальної роботи.	20.05.21	
6	Захист дипломної роботи	15.06.21	

Зміст

Зміст

Анотація.....	5
Вступ	6
Розділ 1.....	8
1.1 Огляд методів для розпізнавання облич8	
1.2 Попередня обробка зображень17	
1.3 Методи розпізнавання облич18	
1.4 Оцінка ефективності систем розпізнавання23	
1.5 Сучасні системи розпізнавання облич25	
Розділ 2.....	30
2.1 Етапи обробки кадрів відеопотоку.....30	
2.2 Виявлення облич методом Віюлі-Джонса31	
2.3 Фільтр Гаусса35	
2.4 Emgu.CV36	
2.5 EigenFaces37	
2.6 FisherFace.....39	
Розділ 3.....	42
3.1 Інструментарій42	
3.2 Реалізація44	
Висновки.....	47
Література.....	49

Анотація

Дипломна робота описує можливі варіанти вирішення проблеми розпізнавання та ідентифікації об'єктів на відео за допомогою різних підходів. У роботі у якості об'єкту розпізнавання та ідентифікації буде розглянуто обличчя людини. Результатом цієї роботи має бути програмне забезпечення, яке може навчитись на сеті для тренування, потім знайти обличчя на відео в реальному часі та, звісно, унікально його ідентифікувати.

У першому розділі було проведено детальний огляд інструментів для вирішення задачі, їх порівняння та опис. Огляд існуючих рішень для вирішення проблеми, порівняння, аналіз та висновки.

У другому розділі ми детальніше описали обраний варіант для реалізації програмного застосунку, глибше розглянули обрані інструменти та підходи до розпізнавання обличчя у відеопотоці.

У третьому розділі описана концепція реалізації власного програмного застосунку, розглянуті та обґрунтовані обрані технічні інструменти для досягнення поставленої цілі.

Вступ

Останнім часом задача пошуку, ідентифікації та розпізнавання об'єктів на відео стає більш і більш актуальною. Звісно, зазвичай потреба розпізнавати об'єкти на відео виникає для зображення у реальному часі. Наприклад ідентифікація користувача мобільного пристрою, розпізнавання трафіку, ДТП та правопорушень, ідентифікація та розпізнавання номерних знаків автівок, обличь людей на вулиці, тощо.

З кожним днем попит на технології, пов'язанні з розпізнаванням людини за її обличчям зростає, як зростають і вимоги до точності розпізнавання. А з точністю зростає технічна складність як у реалізації, так і у підтримці апаратної частини рішень для розпізнавання обличь у реальному часі. Мета цієї роботи проаналізувати і знайти найбільш оптимальний підхід, що дозволить реалізувати програму, яка матиме баланс між технічною складністю та ефективністю у вирішенні цієї задачі.

Наразі існують певні інструменти та комплекси, що дають змогу вирішити цю задачу. Ми розглянемо у нашій роботі проблеми, які виникають у зв'язку з цим твердженням. А саме: чи достатньо вони ефективні; чи можливе покращення; що є основною мірою успіху та результату якщо говорити про розпізнавання об'єктів на відео.

Всім відомо, що розпізнавання обличь вже не є новинкою. Існує багато рішень та підходів до вирішення цієї задачі як на статичних кадрах з фотографій, так і на динамічному відеопотоці в реальному часі. Але що стоїть за цим? Чи можливо розробити просте застосування для розпізнавання обличь в реальному часі не маючи якісного технічного забезпечення та програмного забезпечення рівня енттерпрайз? Звісно, це можливо, але необхідно детально дослідити, які підходи та інструменти використовувати. Мета – розробити додаток, що буде оптимальним зі сторони балансу його можливостей (коректності розпізнавання обличь) та технічних характеристик пристрою.

В роботі буде проведено аналіз існуючих методів обробки відеопотоку, пошуку та розпізнавання на ньому облич. Розглянемо алгоритми та існуючі методи реалізації системи для розпізнавання облич в реальному часі.

В рамках роботи має бути реалізовано програмне забезпечення, концепція та архітектура якого буде напряму залежати від якісного аналізу існуючих рішень методів та підходів до розв'язання задачі розпізнавання облич в реальному часі (відеопотоці), аналітиці найкращої їх взаємодії та поєднанні. Застосунок має бути ефективним у вирішенні задачі розпізнавання облич, мати простий інтерфейс користувача.

Інструментом для реалізацію має бути мова програмування C# та платформа .NET. Такий вибір зроблено тому, що цей інструментарій має широкі можливості в галузі розпізнавання облич, є потужні бібліотеки та фреймворки, але реалізацій на цій мові програмування небагато. З іншого боку використання платформи .NET дає широкі можливості як зі сторони інтеграцій з іншими системами, так і зі сторони кросплатформерності.

Розділ 1

В першому розділі розглянемо основні проблеми та виклики задачі розпізнавання обличь в реальному часі, інструменти та підходи до їх розв'язання. Детально проаналізуємо найпопулярніші методи що використовуються для розпізнавання обличь, розглянемо їх переваги і вади. Для аналітичної інформації та побудови висновків що допоможуть у реалізації власного застосунку оглянемо існуючі рішення ентрепрайз рівня.

1.1 Огляд методів для розпізнавання облич

Процесом розпізнавання обличь прийнято називати набір різних завдань, для ідентифікації людини по цифровому зображенню або відео фрагменту. У загальному вигляді цей процес виглядає наступним чином: після того, як система отримала зображення з камери, за допомогою алгоритмів визначаються межі обличчя (етап виявлення). Тобто комп'ютер має знайти де саме на зображенні знаходиться об'єкт ідентифікації, визначити його точні або приблизні координати. Далі йде етап розпізнавання, на якому обличчя трансформується (змінюється його яскравість, воно вирівнюється, масштабується, і т.д.) і приводиться до деякого заданого виду. Після чого, відбувається обчислення ознак і безпосередньо порівняння їх із закладеними в базу даних еталонами. Цей заключний етап порівняння називається ідентифікація або верифікація, в залежності від системи.

Верифікація: порівняння зразків за схемою «1: 1». Для визначення особистості система порівнює біометричний зразок з одним із біометричних шаблонів, що зберігаються в базі даних, і дає відповідь на питання «Чи є ця людина (на фото) тією людиною, з якою ми порівнюємо (на шаблоні)?».

Ідентифікація: порівняння зразків за схемою «1: N». Для визначення особистості система порівнює біометричний зразок з усіма шаблонами обличь, що зберігаються в базі даних, і дає відповідь на питання «хто це?».



Рисунок 1. Загальний алгоритм розпізнавання обличчя

Після того, як зображення у вигляді цифрових даних з камер передається на комп'ютер - воно обробляється за допомогою спеціального алгоритму, який визначає розташування області обличчя за її основними рисами (очей, рота, брів, носа і т. д.). Таких методів виявлення осіб існує багато і більшість з них представляють собою комбінацію інших методів. Але всі ці методи можна розділити на дві основні категорії: методи на основі знань, які ґрунтуються на досвіді людини і методи виявлення особи за зовнішніми ознаками (методи при яких необхідно провести етап навчання системи, шляхом обробки тестових зображень). Класифікація цих методів виявлення приведена на рисунку 2.

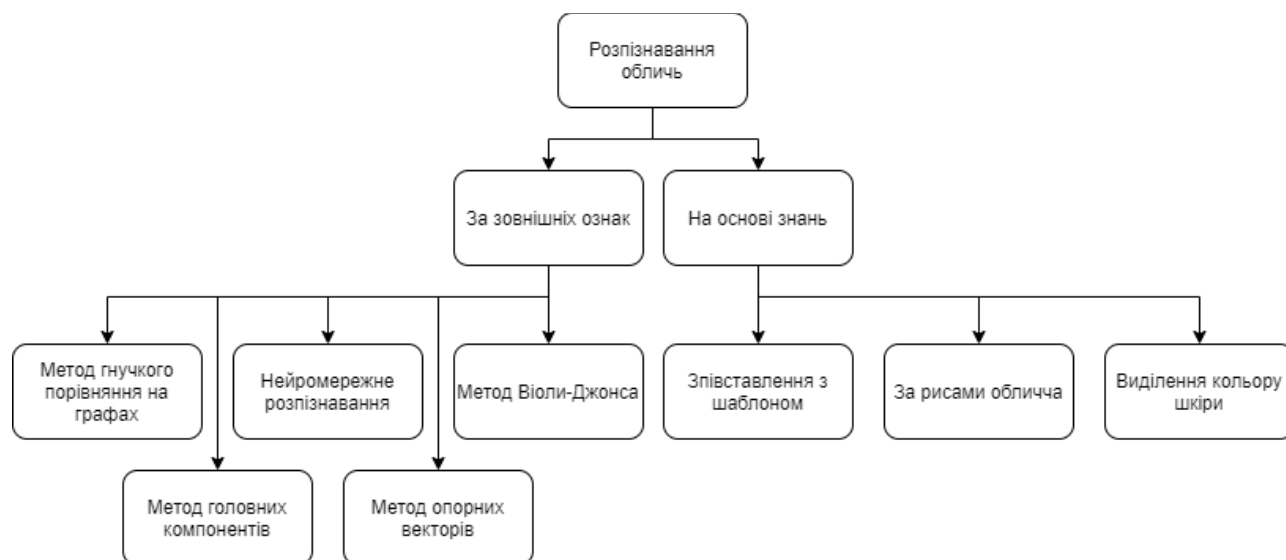


Рисунок 2 Класифікація методів виявлення обличчя

Методи, засновані на знаннях, використовують інформацію про обличчя, його риси, форму, тексту або колір шкіри. У цих методах виділяється певний набір правил (властивостей і особливостей обличчя), яким повинен відповідати фрагмент кадру, для того щоб вважатися людським обличчям. Визначити такий набір правил досить легко. Всі правила - це формалізовані знання, якими керується людина, коли визначає, обличчя перед ним або не обличчя.

Наприклад, основні правила: області очей, носа і рота відрізняються по яскравості від іншої частини обличчя; очі на обличчі завжди розташовуються симетрично відносно один одного. Спираючись на ці та інші схожі властивості, будуються алгоритми, які в ході виконання перевіряють присутність виконання цих правил на зображенні. До цієї ж групи методів відносять більш загальний метод - метод порівняння за шаблоном. У цьому методі, за допомогою опису властивостей окремих областей обличчя і їх заданому взаємному розташуванню визначається стандарт обличчя (шаблон), з яким в подальшому порівнюють вхідне зображення.

Методи на основі знань отримали досить широке поширення і мають непогані показники, однак вони демонструють гарні результати тільки на зображеннях з якісним розширенням, без шумів і з нескладним фоном. На кадрах з відео потоку або камер встановлених в громадських місцях, де можливі різні ракурси, кути, розташування обличч, а також мінливе освітлення і велика кількість заважають об'єктів на задньому плані, існує велика ймовірність виникнення помилок. У вирішенні задачі існують певні підходи для видалення шумів, відсікання непотрібних елементів на фоні. Детальніше методи так званої попередньої обробки будуть описані далі.

Методи виявлення облич за зовнішніми ознаками підходять до проблеми з іншого боку, вони не намагаються в явному вигляді формалізувати процеси, що відбуваються в нашому мозку, а навпаки намагаються виявити закономірності і властивості зображення обличчя неявно, застосовуючи методи математичної статистики і машинного навчання. Методи цієї групи позбавлені вище зазначених недоліків і тому стали частіше застосовуватися в системах відеоспостереження. Пошук облич за допомогою таких методів здійснюється перебором всіх прямокутних фрагментів зображення з метою визначення до якого класу належить зображення: до класу зображень, що містять обличчя або до класу зображень без обличчя.

Через такий великий обсяг роботи ці методи можна характеризувати надмірністю і великою обчислювальною складністю. Для того, щоб зменшити

кількість обчислень і прискорити процес пошуку обличчя, автори застосовують різні додаткові методи для скорочення кількості розглянутих фрагментів.

Кілька найбільш актуальних і популярних методів виявлення осіб розглянуті нижче:

Метод Віоли-Джонса (Viola-Jones object detection). Метод був запропонований Паулом Віолою і Майклом Джонсом в 2001 році і став першим методом, що демонструє високі результати при обробці зображень в реальному часі. У методу є безліч реалізацій, в тому числі в складі бібліотеки комп'ютерного зору OpenCV (функція `cvHaarDetectObjects()`). Детально даний метод розглянуто в розділі 2.

Переваги даного методу:

- висока швидкість роботи (за рахунок використання каскадного класифікатора);
- висока точність виявлення повернених обличчя на кут до 30 градусів (якщо кут більше, ефективність даного методу сильно зменшується);

Недоліки:

- тривалий час навчання. Алгоритму необхідно проаналізувати велику кількість тестових зображень;
- при роботі, на положення обличчя можуть бути встановлені деякі обмеження.

Метод гнучкого порівняння на графах (Elastic graph matching). Метод відноситься до 2D моделювання. Його суть полягає в зіставленні графів, які описують обличчя (обличчя представляється у вигляді сітки з індивідуальним розташуванням вершин і ребер).

Суть методу зводиться до еластичного порівнянню графів, що описують зображення обличчя. Обличчя представлені у вигляді графів зі зваженими вершинами і ребрами. На етапі розпізнавання один з графів – еталонний, залишається незмінним, в той час як інший деформується з метою набуття

якнайкращої відповідності до першого графу. У подібних системах розпізнавання графи можуть являти собою як прямокутну решітку, так і структуру, утворену характерними (антропометричними) точками облич.

У вершинах графа обчислюються значення ознак, найчастіше використовують комплексні значення фільтрів Габора або їх упорядкованих наборів - габоровських вейвлет (строї Габора), які обчислюються в деякій локальній області вершини графа локально, шляхом згортки значень яскравості пікселів за фільтрами Габора.

Ребра графа зважуються відстанями між суміжними вершинами. Різниця (відстань, дискримінаційна характеристика) між двома графами обчислюється за допомогою деякої цінової функції деформації, що враховує як відмінність між значеннями ознак, обчисленими в вершинах, так і ступінь деформації ребер графа.

Деформація графа відбувається шляхом зсуву кожної з його вершин на деяку відстань в певних напрямках щодо її вихідного розташування і вибору такої її позиції, при якій різниця між значеннями ознак (відгуків фільтрів Габора) в вершині деформованого графа і відповідної їй вершині еталонного графа буде мінімальною. Ця операція виконується по черзі для всіх вершин графа до тих пір, поки не буде досягнуто найменша сумарна відмінність між ознаками деформованого і еталонного графів. Значення цінової функції деформації при такому положенні графа і буде мірою відмінності між вхідним зображенням обличчя і еталонним графом. Дана «релаксаційна» процедура деформації повинна виконуватися для всіх еталонних облич, закладених в базу даних системи. Результат розпізнавання системи - еталон з найкращим значенням цінової функції деформації.

Недоліки методу:

- складність алгоритму розпізнавання призводить до необхідності використання значних обчислювальних потужностей;
- складна процедура введення нових шаблонів в базу даних;

- швидкодія аналітичної системи обернено пропорційна розмірами баз даних.

Сховані марківські моделі (СММ). Метод заснований на статистичному порівнянні об'єкта з базою шаблонів. Сховані марківські моделі використовують статистичні властивості сигналів і враховують їх просторові характеристики. Елементи моделі: початкова ймовірність станів, безліч спостережуваних станів, безліч прихованих станів, матриця перехідних ймовірностей. Кожному елементу відповідає своя марківська модель. Під час розпізнавання людини, перевіряються всі згенеровані марківські моделі і шукається найбільша ймовірність того, що послідовність спостережень для об'єкта згенерована відповідною моделлю.

Недоліки:

- низька швидкість обробки;
- низька розрізнявальна здатність і не оптимальний алгоритм навчання;
- система може оптимізувати тільки час обробки даних і час відгуку на власну модель, але не може мінімізувати час перебору інших моделей.

Метод головних компонент (Principal component analysis, PCA).

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

Спочатку метод головних компонент почав застосовуватися в статистиці для зниження простору ознак без істотної втрати інформації. У задачі розпізнавання облич його застосовують головним чином для представлення зображення обличчя вектором малої розмірності, який порівнюється потім з еталонними векторами, закладеними в базу даних.

Головною метою методу головних компонент є значне зменшення розмірності простору ознак таким чином, щоб воно якомога краще описувало

«типові» образи, що належать безлічі облич. Використовуючи цей метод можна виявити різні мінливості в навчальній вибірці зображень облич і описати цю мінливість в базисі декількох ортогональних векторів, які називаються власними (eigenface).

Отриманий один раз на навчальній вибірці зображень облич набір власних векторів використовується для кодування всіх інших зображень облич, які представляються зваженої комбінацією цих власних векторів.

Використовуючи обмежену кількість власних векторів можна отримати стислу апроксимацію вхідному зображенню обличчя, яку потім можна зберігати в базі даних у вигляді вектору коефіцієнтів, що є одночасно ключем пошуку в базі даних облич.

Також, ціллю методу PCA є зменшення простору ознак без значущої втрати інформації і так, щоб воно як найкраще описувало «типові» образи, що належать безлічі облич. В задачі розпізнавання облич його використовують головним чином для того, щоб уявити обличчя як вектор малої розмірності, який потім порівнюється з еталонними векторами з бази.

PCA добре зарекомендував себе в додатках. Однак, тоді, коли на зображенні особи є значні зміни у виразі обличчя або освітлення, ефективність методу значно падає. Це відбувається через те, що метод головних компонент вибирає підпростір з метою максимальної апроксимації вхідного набору даних, а не з метою виконання дискримінації між класами облич.

Метод опорних векторів (Support Vector Machines, SVM) - це набір схожих алгоритмів навчання, що використовуються для задач класифікації та регресійного аналізу. Суть методу опорних векторів полягає в знаходженні гіперплощини в просторі ознак, що відокремлює клас зображень облич від інших зображень (клас зображень «без облич»). При цьому з усіх можливих гіперплощин, які поділяють на два класи, необхідно вибрати таку гіперплощину, відстань до якої від кожного класу є максимальною.

Переваги даного методу:

- висока стійкість до перенавчання;
- висока швидкість роботи в порівнянні з нейронними мережами;
- можливість зменшення чутливості до шуму за рахунок зниження точності.

Недоліки:

- точність роботи методу поступається багатьом іншим методам.

Нейромережні методи. Досить поширені методи, які включають в себе близько десятка різних алгоритмів. Основною особливістю таких мереж є їх здатність до навчання на наборі готових прикладів заздалегідь занесених в базу даних. Під час навчання нейронних мереж, мережа автоматично витягує ключові ознаки і будує взаємозв'язок між ними. Після цього, для того щоб розпізнати до цього невідомий об'єкт, навчена нейронна мережа застосовує отриманий досвід. Нейромережні методи показують одні з найкращих результатів в області розпізнавання, але вважаються найбільш складними для реалізації.

Наразі існує близько десятка різних нейронних мереж (НМ). Одним з найбільш широко використовуваних та найпопулярніших екземплярів є мережа, що побудована на багат шаровому перцептроні, яка дозволяє класифікувати подане на вхід зображення або сигнал відповідно до попередніх налаштувань і досвіду навчання нейронної мережі.

Навчаються нейронні мережі на наборі навчальних прикладів. Суть навчання зводиться до налаштування ваг міжнейронних зв'язків в процесі рішення оптимізаційної задачі методом градієнтного спуску. В процесі навчання нейронної мережі відбувається автоматичне вилучення ключових ознак, визначення їх важливості та побудова взаємозв'язків між ними. Передбачається, що навчена мережа зможе застосувати досвід, отриманий в процесі навчання, на невідомі образи використовуючи при цьому узагальнюючі здібності.

Найкращі результати в області розпізнавання облич (за результатами аналізу публікацій) показала Convolutional Neural Network або сгорткова нейронна мережа (далі - СНМ), яка є логічним розвитком ідей таких архітектур нейронних мереж як когнітрон і неокогнітрон. Успіх обумовлений можливістю розуміння двовимірної топології зображення, на відміну від багат шарового перцептрона.

Відмінними рисами сгорткової нейронної мережі є локальні рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис в будь-якому місці зображення) і ієрархічна організація з просторовими семплінгом (spatial subsampling). Завдяки цим нововведенням СНМ забезпечує часткову стійкість до змін масштабу, зсувів, поворотам, зміні ракурсу і іншим спотворень.

Свій розвиток сгорткові нейронні мережі отримали в розробці DeepFace, яку придбав Facebook для розпізнавання облич користувачів своєї соцмережі. Нажаль, всі особливості архітектури носять закритий характер.

Переваги даного методу:

- висока точність виявлення при правильному налаштуванні параметрів мережі.
- Поліпшений процес навчання мережі.

Недоліки:

- складна процедура внесення змін (будь-яка зміна вимагає перенавчання мережі);
- важко формалізувати архітектуру мережі (кількість нейронів, шарів, характер зв'язків)
- висока обчислювальна складність.

1.2 Попередня обробка зображень

Для підвищення ефективності розпізнавання і якості виділення обличчя, в системах розпізнавання проводять етап попередньої обробки вхідних зображень. По-перше це робиться для того, щоб на наступних етапах інші елементи системи мали менше навантаження та більш якісні вхідні данні.

Після того, як система знаходить обличчя на кадрі і визначає положення голови, розмір, позу і його ключові риси, воно, як правило, піддається нормалізації: кодується, масштабується, перетворюється до горизонтального положення лінії, що з'єднує центри очей. Також, на етапі попередньої обробки обличчя, застосовуються різноманітні фільтри для зниження рівня шуму (медіанний, гаусовський та ін.).

Методи попередньої обробки досить різноманітні і залежать від задач досліджень. Найбільш часто використовуються такі фільтри:

- Нелінійні фільтри - фільтри, які використовують для видалення імпульсного шуму (окремих точок з максимальною (білою) або мінімальною (чорною) яскравістю) на зображенні. Алгоритм дії таких фільтрів полягає у визначенні позиції кожного імпульсу і заміні їх значеннями фіксованої або випадкової величини.
- Фільтр Гауса - фільтр розмиття зображення, який використовують, якщо на зображенні існують дрібні деталі, які не вимагають відділення від фону і їх можна розмити.
- Медіанні фільтри - фільтри, які використовуються для збереження перепадів яскравості контурів і зменшення впливу імпульсних шумів. Медіанні фільтри досить добре працюють в тих випадках, коли щільність шуму невелика.

1.3 Методи розпізнавання обличч

Інша важлива частина систем автоматичного розпізнавання обличч це безпосередньо самі алгоритми розпізнавання людини. На сьогоднішній день таких алгоритмів існує досить багато, і кожен з них має свою специфіку, свою швидкість роботи, свою складність реалізації, свою точність і свою надійність розпізнавання.

Алгоритми розпізнавання діляться на дві категорії, в залежності від застосовуваної технології розпізнавання - двовимірні, в яких розпізнавання відбувається по геометрії обличчя (2D-технології) і тривимірні, в яких розпізнавання відбувається за будовою черепа (3D-технології).

Системи 2D-розпізнавання працюють з «плоскими», двовірними зображеннями і розпізнають обличчя, аналізуючи його текстуру і ділянки обличчя з високою контрастністю, тому при порушенні освітлення або положення обличчя його розпізнавання сильно ускладнюється.

Системи 3D-розпізнавання є більш стійкими до таких змін. Ця чіткість набувається завдяки підходу з побудовою 3-D моделі черепа голови людини, чие обличчя необхідно розпізнати. При створенні моделі обличчя, ці моделі враховують усі можливі особливості побудови черепа, що дає чітке представлення та бачення всіх особливостей обличчя. Але проте, хоч системи 3-D розпізнавання обличчя і мають такі суттєві переваги, через відсутність можливості обслуговувати велику кількість користувачів в режимі ідентифікації, складність реалізації, високу потребу в ресурсах машини та через невисоку швидкості, 3D-технології поки не отримали широкого застосування.

Через те, що 3D-технології розпізнавання вимагають великих обчислювальних ресурсів, вартість обладнання для таких систем набагато вище, ніж у 2D. Тим не менш, такий підхід до вирішення проблеми розпізнавання обличч є очевидним майбутнім, адже з часом обчислювальні можливості технічного обладнання будуть тільки зростати та ставати доступнішими.

На сьогоднішній день в системах, де використовується розпізнавання обличчя, наприклад системах відеоспостереження часто використовується кілька ефективних алгоритмів розпізнавання, всі вони засновані або на значеннях пікселів, або на характерних точках. Нижче подано короткий опис кожної з цих підгруп.

Методи, засновані на значеннях пікселів, використовує для розпізнавання виявлених обличчя колір або яскравість пікселів. Найпростішими подібними методами є **Eigenfaces**, **Fisherfaces**, локальні бінарні шаблони, активні моделі зовнішнього вигляду, активні моделі форми.

Eigenfaces (дослівно «своє власне обличчя») - алгоритм, який в 1991 році запропонували Метью Терком і Алексом Пентландом і який здобув широку популярність в якості першого успішного методу розпізнавання обличчя. Основною ідеєю алгоритму **Eigenfaces** є уявлення відмінних характеристик зображення обличчя, за допомогою двовимірних зображень в градаціях сірого.

Для ідентифікації особистості алгоритм порівнює отримане обличчя з кадром, з уже зареєстрованим шаблоном в базі, і визначає коефіцієнт відмінності. Цей коефіцієнт відмінності між шаблонами і показує ступінь схожості. Власні вектори отримані з матриці коваріації розподілу ймовірностей у високовимірному векторному просторі зображень обличчя. Самі власні грані утворюють базовий набір усіх зображень, що використовуються для побудови матриці коваріації. Це забезпечує зменшення розмірів, що дозволяє меншому набору базових зображень представляти вихідні навчальні зображення.

Класифікація може бути досягнута шляхом порівняння того, як обличчя представлені набором базисів. Алгоритм **Eigenfaces** показує високі результати при використанні в добре освітлених приміщеннях і коли є можливість сканування обличчя в анфас, що стосується змін умов використання даного алгоритму, показники його ефективності різко знижуються.

Fisherfaces - нащадок **Eigenface**, що забезпечує більш високу точність розпізнавання при змінах освітлення або виразу обличчя. На відміну від

методу Eigenface, в основі Fisherfaces лежить лінійний дискримінантний аналіз LDA, а саме лінійний дискримінант Фішера.

Дія алгоритму Fisherfaces заснована на пошуку проекції даних, при якій класи зображень обличчя максимально роздільні (тобто основні специфічні особливості). Ця відмінність з Eigenface дозволяє вирішити проблему високої чутливості до змін освітлення.

Локальні бінарні шаблони (Local Binary Pattern, LBP) - простий в реалізації і ефективний метод розпізнавання обличчя. При розпізнаванні методом LBP, кожного пікселя зображення за допомогою функції присвоюється значення яскравості, яке описує його рамки. Отримане зображення розділяється на області, для кожної з яких розраховується гістограма. Далі гістограми конкатенуються і порівнюються за допомогою методів машинного навчання. У класичному варіанті використовується метод найближчого сусіда.

На відміну від Eigenfaces, алгоритм стійкий до монотонних змін освітлення, що робить його придатним для розпізнавання обличчя в системах обробки в реальному часі. Методи, засновані на характерних точках, на відміну від попередніх, які не оцінюють яскравості пікселів, використовують координати характерних точок на зображенні. Такими характерними точками можуть бути, наприклад, центр очей, положення носа, лінія брів, рота і т.д. До даного класу методів відносяться активні моделі зовнішнього вигляду і активні моделі форми.

Активні моделі зовнішнього вигляду (Active Appearance Models, AAM) - це статистичні моделі зображень, які шляхом різного роду деформацій можуть бути адаптовані під реальне зображення. Даний тип моделей в двовимірному варіанті було запропоновано Тімом Кутса і Крісом Тейлором в 1998 році . Спочатку активні моделі зовнішнього вигляду застосовувалися для оцінки параметрів зображень обличчя.

Активна модель зовнішнього вигляду містить два типи параметрів: параметри, пов'язані з формою (параметри форми), і параметри, пов'язані зі статистичною моделлю пікселів зображення або текстурою (параметри зовнішнього вигляду). Перед використанням модель необхідно навчити на доволі великій кількості заздалегідь розмічених зображень (звісно, чим більше – тим краще). Розмітка зображень виробляється вручну. Кожна мітка має свій номер і визначає характерну точку, яку повинна буде знаходити модель під час адаптації до нового зображення.

Процедура навчання ААМ починається з нормалізації форм на розмічених зображеннях з метою компенсації різниці в масштабі, нахилі і зміщенні обличчя на зображенні. Для цього використовується так званий узагальнений прокрустовський аналіз. З усієї безлічі нормованих точок потім виділяються головні компоненти з використанням методу РСА.

Далі з пікселів всередині трикутників, утворених точками форми, формується матриця, така що, кожен її стовпець містить значення пікселів відповідної текстури. Варто відзначити, що текстури, які використовуються для навчання можуть бути як одноканальними (градації сірого), так і багатоканальними (наприклад, простір кольорів RGB або інше). У разі багатоканальних текстур вектори пікселів формуються окремо по кожному з каналів, а потім виконується їх конкатенація. Після знаходження головних компонент матриці текстур модель ААМ вважається навченою.

За допомогою активної моделі зовнішнього вигляду можна моделювати зображення об'єктів, схильних до як жорсткої, так і нежорсткої деформації. ААМ складається з набору параметрів, частина яких представляють форму особи, інші задають його текстуру. Під деформацією зазвичай розуміють геометричне перетворення у вигляді композиції перенесення, повороту і масштабування. При вирішенні задачі локалізації особи на зображенні виконується пошук параметрів (розташування, форма, текстура) ААМ, які представляють синтез зображення, найбільш близьке до спостережуваного. За

ступенем близькості активної моделі зовнішнього вигляду обраному зображенню приймається рішення – присутнє обличчя чи ні.

Активні моделі форми (Active Shape Models, ASM) враховують статистичні зв'язку у взаємному розташуванні антропометричних точок. Суть методу ASM полягає в обрахуванні статистичних зв'язків між розташуванням антропометричних точок на наявній вибірці зображень облич, знятих в анфас. На зображенні експерт розмічає розташування антропометричних точок. На кожному зображенні точки пронумеровані в однаковому порядку.

Для того щоб привести координати на всіх зображеннях до єдиної системи, зазвичай виконується узагальнений прокрустовський аналіз, в результаті якого всі точки приводяться до одного масштабу і центруються. Далі для всього набору образів обчислюється середня форма і матриця коваріації. На основі матриці коваріації обчислюються власні вектору, які потім сортуються в порядку убутання відповідних їм власних значень.

Локалізації активної моделі форми на новому, що не входить в навчальну вибірку зображенні здійснюється в процесі рішення оптимізаційної задачі. Однак все ж головною метою AAM і ASM є не розпізнавання осіб, а точна локалізація особи і антропометричних точок на зображенні для подальшої обробки.

Практично у всіх алгоритмах обов'язковим етапом, що передуює класифікацію, є вирівнювання, під яким розуміється вирівнювання зображення обличчя у фронтальне положення щодо камери або приведення сукупності облич (наприклад, в навчальній вибірці для навчання класифікатора) до єдиної системи координат. Для реалізації цього етапу необхідна локалізація на зображенні характерних для всіх облич антропометричних точок - найчастіше це центри зіниць або куточки очей. Різні дослідники виділяють різні групи таких точок. З метою скорочення обчислювальних витрат для систем реального

часу розробники виділяють не більше 10 таких точок. Саме моделі AAM і ASM призначені для того щоб точно локалізувати ці антропометричні точки на зображенні обличчя.

1.4 Оцінка ефективності систем розпізнавання

Між спеціалізованими і аматорськими системами розпізнавання існує велика різниця. Аматорські - виглядають красивіше, дорожче, і до них пред'являються більш низькі вимоги. Також, за рахунок того, що в разі допущених помилок даними системами, наслідки зазвичай незначні, їх вартість набагато нижча за вартість професійних систем, вимоги до яких пред'являються серйозніше.

Для прикладу можна взяти технологію розпізнавання осіб на фотографіях в соціальних мережах, і професійну систему, призначену для пошуку розшукуваних людей. Перша система працює з фотографіями, на яких люди зазвичай дивляться в об'єктив, а друга - з людьми, знятими в громадському місці, які не дивляться в камери відеоспостереження, а іноді навіть спеціально від них ховаються. Також варто зауважити, що якщо перша система допустить помилку, то нічого страшного не станеться, а от якщо система пошуку розшукуваних людей буде допускати багато помилок, то це серйозний недолік, так як будь-який з пропущених людей може виявитися злодієм, або навпаки, звичайний громадянин може бути виявлений у якості злодія. Такі похибки недопустимі у застосунка рівня ентєрпрайз та в цьому прикладі. Тому при виборі системи важливо правильно визначити не тільки мету, але і критерії для оцінки ефективності цієї системи. А це значить визначити пристосованість системи до роботи в певних умовах і різні пороги допустимих помилок. Звісно, що чим більше поріг допустимих помилок – тим простіше буде система. А це зазвичай означає, що вона буде дешевшою зі сторони розробки, підтримки, так буде менше вимагати із технічної сторони

На сьогодні вже є певні шаблони для виконання цієї задачі. Для оцінки ефективності систем розпізнавання виділяють два основні параметри. У

біометрії їх називають *коефіцієнтом помилкового доступу* (FAR - False Acceptance Rate) і *коефіцієнтом помилкового відмови* (False Rejection Rate - FRR). Коефіцієнт помилкової відмови показує ймовірність того, як часто особи, бувають помилково відкинуті, або іншими словами - система не знайшла в базі людину, яка насправді зображена на фотографії чи кадрі з відеозапису. Чим цей коефіцієнт менше, тим вище точність розпізнавання. Коефіцієнт помилкового допуску - це навпаки, ймовірність того, як часто система помилково робить висновок. У результаті такого висновку, замість людини яка насправді була об'єктом ідентифікації система зробить хибне припущення та покаже іншу людину. Існує ще один коефіцієнт, що дозволяє порівнювати біометричні системи - *коефіцієнт EER* (рівний рівень помилок). Це коефіцієнт, при якому обидві помилки (помилка прийому і помилка відхилення) еквівалентні. Чим нижче коефіцієнт EER, тим вище точність біометричної системи. На рисунку 3 показані взаємозв'язку характеристик FAR, FRR і EER.

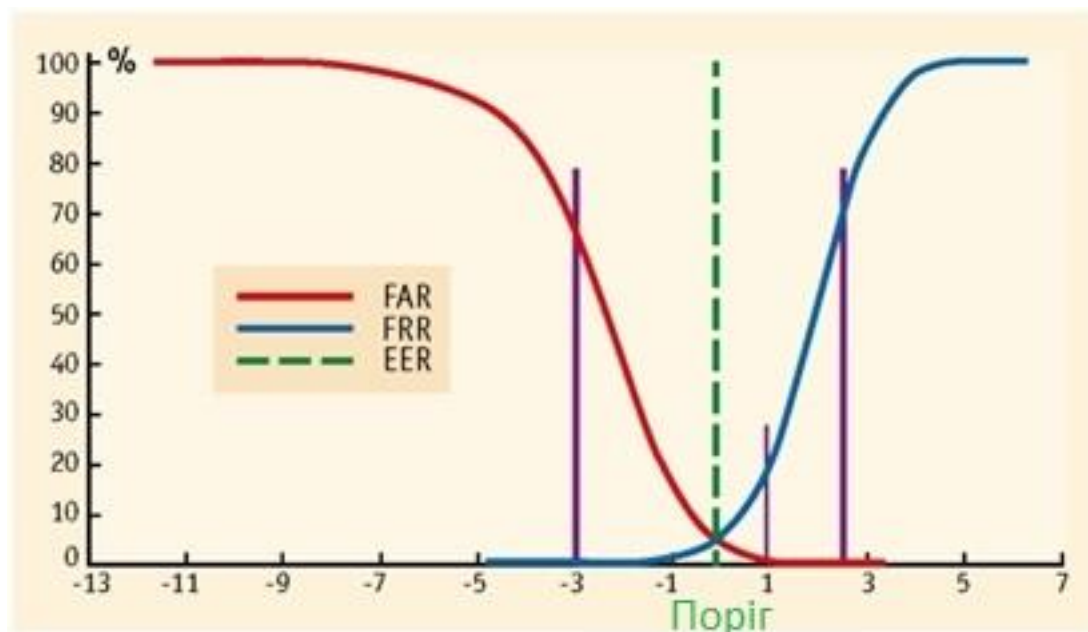


Рисунок 3 Характеристики біометричних систем

Для порівняння систем і візуалізації залежності між помилками використовують графік ROC (Receiver Operating Characteristic).

Крива ROC визначає, наскільки зразок повинен бути близький до шаблону, щоб система могла вважати це збігом. При регулюванні порогів помилок, варто не забувати, що величини FAR і FRR взаємодоповнюють і якщо

зменшити, то буде менше помилкових розбіжностей, але більше помилкових прийомів.

1.5 Сучасні системи розпізнавання обличч

З кожним роком зростання інтересу до технологій розпізнавання обличч помітно збільшується. На сьогоднішній день вже існує кілька десятків систем розпізнавання, які застосовуються в різних сферах людського життя.

Лідерами серед компаній розробляють такі системи вважаються:

корпорація NEC (Японія), Cognitec Systems GMBH (Німеччина), «VisionLabs» (Росія) «FACE ++» і Neurotechnology (Литва). Розглянемо їх короткий опис та технічні особливості.

Система «FaceVACS» компанії «Cognitec Systems»

«FaceVACS-VideoScan» - просте у використанні та налаштуванні програмне забезпечення розпізнавання осіб по відеопотоку в реальному часі, пропонуване компанією «Cognitec Systems».

Система «FaceVACS-VideoScan» складається з декількох системних компонентів: відео серверу, який керує відеопотоками; сервера відеосканування, що координує всі компоненти системи і виконує основні біометричні операції; обчислювального вузла, який використовується для розподілу обчислювального навантаження; інтерфейсу користувача; диспетчеру сигналів що отримує повідомлення про події та обслуговує мобільні пристрої; операційної бази даних; комплекту інтеграторів.

На сьогодні технологія FaceVACS використовує алгоритм розпізнавання осіб B10T9. Цей алгоритм стійкий до змін міміки, поворотам особи (до 15°) часткового його закриття, використання окулярів і зміни освітлення.

Крім того, система FaceVACS має такі особливості:

- можливість одночасного відстеження декількох обличч;
- порівняння обличч відбувається в реальному часі;
- можливість відображення і відправки статистики про потоки;

- підтримка інтерактивної реєстрації з нерухомого;
- застосування C ++ API і Web Services API.

Система «NEC's Face Recognition» компанії «NEC»

«NEC's Face Recognition» - одна з передових систем розпізнавання обличч, розроблена японською компанією «NEC», що дозволяє ідентифікувати людей по кадрам багаторічної давності і навіть, якщо людина знаходиться в окулярах або сильно змінює міміку обличчя. Всі розпізнані обличчя зберігаються в базі даних, тому якщо буде потреба можна підняти всю історію відеореєстрації та переглянути дату і час будь-якого збереженого зображення. Технологія NEC перевершує безліч інших систем розпізнавання своєю точністю і швидкістю. Вона має гарні показники продуктивності в різних ситуаціях, у тому числі при роботі з відео низької якості і сильно стислими зображеннями. NEC аналізує індивідуальні особливості обличчя (Розмір, форму зіниць, лінії носа і рота), їх взаємне розташування, і знаходить потім по цій інформації відповідну людину в базі даних.

Система включає в себе декілька модулів, що реалізують такі алгоритми:

1. Використовується метод узагальненого відповідності (GMFD), який забезпечує високу швидкість знаходження і високу точність розпізнавання особи. Метод GMFD заснований на нейронних мережах і здійснює попередній пошук пар очей;

2. алгоритм PSM (Perturbation Space Method), дозволяє ефективно вирішувати проблеми, що пов'язані зі складнощами що виникають через незручне розташування 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 особи, наявність окулярів, головного убору, волосся на обличчі, тощо) на точність розпізнавання.

Система розпізнавання облич NeoFace володіє наступними особливостями:

- можливість спостереження і контролю в реальному часі;
- ідентифікація на основі індивідуальних рис обличчя;
- множинне розпізнавання;
- можливість пошуку подій по базі даних;
- ведення журналу зображень обличчя;
- стійкість до повороту обличчя до 15° і нахилу голови до 45° в

будь-якому напрямку від фронтального положення;

- масштабований і необмежений розмір бази даних;
- незалежне розпізнавання напрямку погляду і характеристик обличчя (окуляри, борода і вираз обличчя).

Система «LUNA SDK» компанії «VisionLabs»

«LUNA SDK» - спеціалізоване ПО, супроводу і розпізнавання обличчя людей на цифрових фотографіях або в відеопотоці від компанії «VisionLabs». Її двигун має одні з кращих в світі показників повноти і точності розпізнавання в реальних умовах.

Процес розпізнавання обличчя в LUNA SDK працює на основі глибинних нейронних мереж і складається з декількох ключових етапів: визначення положення і розмірів всіх обличчя в кадрі (детекції); визначення розташування характерних рис обличчя і трансформації його в стандартизовану форму (вирівнювання); отримання дескрипторів (числових векторів, які підсумовують характерні ознаки особи); порівняння осіб з базою зображень і запобігання підміни осіб.

Технічні характеристики і особливості LUNA SDK:

- повністю розроблена на C ++;
- можливість спостереження і контролю в реальному часі;
- можливість працювати в режимі великої мультисредовості;
- використання алгоритму розпізнавання осіб на основі глибинних нейронних мереж;

- оптимізована управління пам'яттю;
- компактні дескриптори, які забезпечують обчислювальну ефективність;
- підтримка різних платформ, включаючи Windows і Linux.

Дескриптори LUNA навчені на мільйонах осіб з різних джерел і забезпечують високу точність в різних умовах, наприклад, в банках, в системах відеоспостереження і в соціальних мережах.

Система «VeriLook SDK» компанії «Neurotechnology»

«VeriLook SDK» - технологія ідентифікації обличь, розроблена компанією «Neurotechnology». Являє собою систему виявлення обличь з можливістю одночасного множинного розпізнавання людей присутніх в кадрі і швидкої ідентифікації обличь (знаходить до 100000 осіб за секунду). VeriLook SDK доступна у вигляді комплекту для розробки ПО і підтримує широкий вибір пристроїв на Windows Linux, Mac OS X, iOS і Android.

Алгоритм VeriLook реалізує локалізацію обличчя з використанням алгоритмів обробки цифрових зображень заснованих на глибоких нейронних мережах.

Основними перевагами системи VeriLook є відсутність необхідності контакту із засобами сканування і швидке впровадження функцій біометричної ідентифікації в прикладні системи замовника, а також існує ряд інших переваг:

- одночасна обробка декількох осіб;
- гендерна класифікація. За бажанням, стать може бути визначена для кожної людини на зображенні;
- живе розпізнавання обличчя. VeriLook визначає, чи є обличчя в відеопотоці обличчям реальної людини або фотографією;
- розпізнавання емоцій. VeriLook аналізує шість основних емоцій: гнів, відраза, страх, щастя, смуток і здивування;
- атрибути обличчя. VeriLook може бути налаштоване для виявлення визначених 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 будуть зберігатися тільки шаблони кращої якості.
- кілька зразків одного і того ж обличчя. Ці зразки можуть бути створені та отримані з різних джерел і в різний час, що дозволяє поліпшити якість зіставлення;
- ідентифікаційна здатність. Функції VeriLook можуть використовуватися в зіставленні один-до-одного (перевірка), а також в режимі один-до-багатьох (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 і того ж об'єкта;
- алгоритм VeriLook здатний зіставляти межі, які були захоплені в інфрачервоному спектрі.

Розділ 2

У цьому розділі детально описані етапи обробки кадрів з відеопотоку та методології їх обробки. Проведено аналіз методів обробки та попередньої обробки зображень для ідентифікації обличчя людини на них.

2.1 Етапи обробки кадрів відеопотоку

Як було зазначено раніше, обробка кадрів відеопотоку складається з двох основних етапів. На першому етапі відбувається виявлення обличчя в кадрі, а на другому – безпосередньо розпізнавання виявленого обличчя. Під час розробки системи, був використаний метод виявлення осіб Віюлі-Джонса, а для розпізнавання – методи Eigenface і FisherFace, які було розглянуто раніше. Розпізнавання обличчя в системі проводиться на основі пошуку найбільш відповідного шаблону базових зображень, що зберігається в базі даних.

Також варто одразу зазначити, що продуктивність обраних алгоритмів істотно залежить від різних чинників, наприклад: освітлення, положення особи, заднього фону і т.д. Так, метод FisherFace показав себе кращим при використанні штучного освітлення, несприятливого освітлення, нахилу обличчя або голови. А метод Eigenface показав кращий результат при знаходженні обличчя на відстані від камери, та якщо на обличчі присутні емоції

Крім основних етапів виявлення і розпізнавання, в розробленому програмному забезпеченні для розпізнавання обличчя існують проміжні етапи обробки знайдених осіб: фільтр Гаусса, який після виявлення обличчя допомагає знизити вплив шумів, а також маска значимих областей, яка дозволяє прибрати вплив кутових областей зображення, що містять задній план.

В результаті узагальнений алгоритм обробки кадрів в програмному застосунку містить наступні етапи: виявлення обличчя, обробка знайденого обличчя за допомогою фільтра Гаусса,. Також, за складеним узагальненому

алгоритму обробки відеопотоку, був складений необхідний функціонал, що розробляється:

- висновок відеопотоку з підключеною до комп'ютера камери в реальному часі або можливість завантаження відео з файлу;
- виведення інформації про приналежність розпізнається особи до певного класу;
- можливість навчання та додавання класів осіб з використанням камери через інтерфейс програми;
- можливість налаштування параметрів роботи алгоритмів.

2.2 Виявлення обличч методом Віюлі-Джонса

Для розробки системи розпізнавання, був обраний метод виявлення осіб Віюлі-Джонса. Цей метод був розроблений в 2001 році, але завдяки своїй високій швидкості, а також вкрай низьку ймовірність помилкового виявлення обличчя до сих пір є одним з основних методів пошуку об'єктів на зображенні. Основні принципи, на яких заснована робота даного методу:

- представлення зображення в інтегральному вигляді;
- пошук осіб за допомогою ознак Хаара;
- каскадна класифікація;
- навчання системи розпізнавання об'єктів на основі методу «AdaBoost»;

Для підвищення ефективності роботи з даними, в методі Віюлі-Джонса застосовується метод інтегрального уявлення, який дозволяє швидко розраховувати суму яскравості пікселів довільного прямокутника на заданому кадрі. Таке уявлення зображення являє собою матрицю однакової за розмірами з вихідним зображенням, кожен елемент якої зберігає в собі суму інтенсивностей всіх пікселів, що знаходяться над ним і зліва від нього плюс його власну вагу.

У розширеному методі віюлі-Джонса, який використовується в бібліотеці комп'ютерного зору OpenCV, використовуються додаткові ознаки Хаара. Каскади Хаара представляють собою прямокутні області, які складені з

декількох сусідніх прямокутних областей, позначених як світлі чи темні.

Коефіцієнтом присутності ознак є різниця між сумою значень яскравості точок що закриваються світлою частиною ознаки і точок, які закривають темною частиною ознаки.

Пошук обличчь відбувається за допомогою так званого скануючого вікна, алгоритм сканування якого виглядає так:

- є досліджуване зображення, обрано вікно сканування, обрані ознаки;
- далі вікно сканування починає послідовно рухатися по зображенню з кроком в 1 осередок вікна (припустимо, розмір самого вікна є $24 * 24$ осередки);
- при скануванні зображення в кожному вікні обчислюється приблизно 200 000 варіантів розташування ознак, за рахунок зміни масштабу ознак і їх положення в вікні сканування;
- сканування проводиться послідовно для різних масштабів; масштабується не саме зображення, а скануюче вікно (змінюється розмір осередку);
- всі знайдені ознаки передаються класифікатору, який визначає за їх значенням, є чи область зображення, відповідна вікну, обличчям чи ні.

Оскільки для опису об'єкта з достатньою точністю необхідна велика кількість ознак Хаара, вони не дуже підходять для навчання або класифікації. У зв'язку з цим, для прискорення процесу виявлення, в методі Віоли-Джонса використовується каскадний класифікатор, який дозволяє прискорити виявлення осіб, фокусуючи роботу на найбільш цікавих областях зображення.

Таким чином при малих обчислювальних витратах можна на ранніх етапах розпізнавання відкинути зображення, з великою часткою вірогідності не містять шуканий об'єкт (в даному випадку особа). Кожен рівень каскаду навчається за допомогою алгоритму AdaBoost.

AdaBoost

Для вирішення проблеми навчання існує технологія бустінга. Бустінг - це комплекс методів, які сприяють підвищенню точності аналітичних моделей. Ефективна модель, яка припускає мало помилок класифікації, називається «сильною». «Слабка» ж, навпаки, не дозволяє надійно розділяти класи або давати точні прогнози, робить велику кількість помилок під час опрацювання зображень. Тому бустінг (від англ. Boosting - підвищення, посилення, поліпшення) означає дослівно «посилення» «слабких» моделей - це процедура послідовної побудови композиції алгоритмів машинного навчання, коли кожен наступний алгоритм прагне компенсувати недоліки композиції всіх попередніх алгоритмів.

Ідея бустінга була запропонована Робертом Шапіро (Scharire) в кінці 90-х років, коли актуальною була проблема, як при наявності безлічі слабких (незначно відрізняються від випадкових) алгоритмів навчання, отримати один сильний. В основі такої ідеї лежить побудова ланцюжка (ансамблю) класифікаторів, який називається каскадом, кожен з яких (крім першого) навчається на помилках попереднього. Наприклад, один з перших алгоритмів бустінга Boost1 використовував каскад з трьох моделей, перша з яких навчалася на всьому наборі даних, друга - на вибірці прикладів, в половині з яких перша дала правильні відповіді, а третя - на прикладах, де «відповіді» перших двох розійшлися. Таким чином, має місце послідовна обробка прикладів каскадом класифікаторів, причому так, що завдання для кожного наступного стає важче. Результат визначається шляхом простого голосування: чи відноситься приклад до того класу, який виданий більшістю моделей каскаду.

Бустінг є жадібним алгоритмом побудови композиції алгоритмів (greedy algorithm) - це алгоритм, який на кожному кроці робить локальний вибір на його розсуд, найкращого рішення, вважаючи, що підсумкове рішення буде оптимальним. Бустінг над вирішальними деревами вважається одним з найбільш ефективних методів з точки зору якості класифікації. У багатьох експериментах спостерігалось практично необмежене зменшення частоти

помилки на незалежній тестовій вибірці при нарощуванні композиції. Більш того, якісний результат на тестовій вибірці часто поліпшувався навіть після досягнення безпомилкового розпізнавання всієї навчальної вибірки. Це змінило уявлення про те, що для підвищення узагальнюючої здатності необхідно обмежувати складність алгоритмів. На прикладі бустінга стало зрозуміло, що якісний результат можуть показувати будь-які композиції незалежно від їх складності, якщо їх правильно налаштовувати.

Плюси AdaBoost:

- гарна узагальнююча здатність. В реальних задачах практично завжди будуються композиції, що перевершують за якістю базові алгоритми. Узагальнююча здатність може поліпшуватися в міру збільшення числа базових алгоритмів;
- простота реалізації;
- власні накладні витрати бустінга невеликі. Час побудови композиції практично повністю визначається часом навчання базових алгоритмів;
- можливість ідентифікувати об'єкти, які є шумовими викидами. Це найбільш «важкі» об'єкти x_i , для яких в процесі нарощування композиції ваги w_i приймають максимальні значення.

Мінуси AdaBoost:

- Буває перенавчання при наявності значного рівня шуму в даних. Експоненціальна функція втрат занадто сильно збільшує ваги «найбільш важких» об'єктів, на яких помиляються багато базових алгоритмів. Однак саме ці об'єкти найчастіше виявляються шумовими викидами. В результаті AdaBoost починає налаштовуватися на шум, що веде до перенавчання. Проблема вирішується шляхом видалення викидів або застосування менш «агресивних» функцій втрат. Зокрема застосовується алгоритм GentleBoost;

- AdaBoost вимагає досить довгих навчальних вибірок. Інші методи лінійної корекції, зокрема, беггінг, здатні будувати алгоритми порівнянної якості за меншими вибірками даних;
- буває побудова неоптимального набору базових алгоритмів. Для вдосконалення композиції можна періодично повертатися до раніше побудованим алгоритмам і навчати їх наново.
- Бустінг може призводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Такі композиції виключають можливість змістовної інтерпретації, вимагають великих обсягів пам'яті для зберігання базових алгоритмів і істотних витрат часу на обчислення класифікацій.

В наші дні підхід посилення простих класифікаторів є популярним і, ймовірно, найбільш ефективним методом класифікації за рахунок високої швидкості та ефективності роботи і відносну простоту реалізації.

2.3 Фільтр Гаусса

Розмиття є невід'ємною частиною різних технік корекції зображення, спрямованих на усунення специфічних дефектів (зайва деталізація, дефекти сканування, пил, і т.д). Одним з їх можливих застосувань є шумозаглушення, тобто задача відновлення вихідного зображення, до пікселів якого доданий випадковий шум.

Розмиття по Гаусу - це характерний фільтр розмиття зображення, який використовує нормальний розподіл (гаусовський розподіл) для обчислення перетворення, що застосовується до кожного пікселя зображення.

Шум в зображенні змінюється незалежно від пікселя до пікселя і, за умови, що математичне очікування значення шуму дорівнює нулю, шуми сусідніх пікселів компенсують один одного. Чим більше вікно фільтрації,

тим менше буде усереднена інтенсивність шуму, однак при цьому буде відбуватися і суттєве розмиття значущих деталей зображення.

У разі двох вимірів - ця формула задає поверхню, що має вигляд концентричних кіл з розподілом Гаусса від центральної точки. Пікселі, де розподіл відмінно від нуля використовуються для побудови матриці згортки, яка застосовується до вихідного зображення. Значення кожного пікселя стає середньо зваженим для околиці. Початкове значення пікселя приймає найбільшу вагу (має найвище гаусове значення), і сусідні пікселі приймають менші ваги, в залежності від відстані до них. У теорії, розподіл в кожній точці зображення буде ненульовим, що вимагало б обчислення вагових коефіцієнтів для кожного пікселя зображення. Але, на практиці, коли розраховується дискретне наближення функції Гаусса, не враховують пікселі на відстані понад 3 σ , тому що вони досить малі.

2.4 Emgu.CV

OpenCV (англ. Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим вихідним кодом) - бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов.

Emgu CV є крос-платформних .Net доповненням для бібліотеки OpenCV для обробки зображень. Розроблено для роботи з .NET сумісними мовами, такими як C #, VB, VC ++, IronPython і т.д., може бути використано в Visual Studio, Xamarin, працює з Windows, Linux, Mac OS X, IOS, Android і Windows Phone. Основні модулі бібліотеки:

sxcore - ядро

* Містить базові структури даних і алгоритми:

- базові операції над багатовимірними числовими масивами;

- матрична алгебра, математичні функції, генератори випадкових чисел;
- запис / відновлення структур даних в / з XML;
- базові функції 2D графіки.

CV - модуль обробки зображень і комп'ютерного зору:

- базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення кольірних просторів і т.д.);
- аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми);
- аналіз руху, спостереження за об'єктами;
- виявлення об'єктів, зокрема обличч;
- калібрування камер, елементи відновлення просторової структури.

Highgui - модуль для введення / виведення зображень і відео, створення призначеного для користувацького інтерфейсу:

- захоплення відео з камер і з відео файлів, читання / запис статичних зображень;
- функції для організації простого UI .

Сваux - експериментальні і застарілі функції:

- просторовий зір: стерео калібрації, прості калібрації;
- пошук стерео-відповідності, кліки в графах;
- знаходження і опис рис обличчя.

2.5 EigenFaces

Як і будь-який алгоритм із серії machine learning, eigenface необхідно спочатку навчити. Для цього використовується навчальна вибірка (training set), що представляє собою зображення осіб, які ми хочемо розпізнати. Після того як модель навчена, ми подамо на вхід деяке зображення і в результаті отримаємо відповідь на питання: якому зображенню з навчальної вибірки з

найбільшою ймовірністю відповідає приклад на вході, або не відповідає жодному.

Завдання алгоритму уявити зображення як суму базисних компонент (зображень):

$$\Phi_i = \sum_{j=1}^K w_j u_j$$

Де Φ_i - інтегроване (тобто за вирахуванням середнього) i -е зображення вихідної вибірки, w_j є ваги і u_j власні вектора (eigenvectors або, в рамках даного алгоритму, eigenfaces).

Навчальну вибірку необхідно спроектувати в новий простір (причому простір, як правило, набагато більше розмірності, ніж вихідне 2х мірне зображення), де кожна розмірність буде давати певний внесок в загальне уявлення. Метод головних компонент дозволяє знайти базис нового простору таким чином, щоб дані в ньому розташовувалися, в деякому сенсі, оптимально. Щоб зрозуміти, просто уявіть, що в новому просторі деякі розмірності (ака головні компоненти або власні вектора або eigenfaces) будуть «нести» більше загальної інформації, тоді як інші будуть нести тільки специфічну інформацію. Як правило, розмірності більш високого порядку (що відповідають меншим власним значенням) несуть набагато менше корисної (в нашому випадку під корисною розуміється щось, що дає узагальнене уявлення про всій вибірці) інформації, ніж перші розмірності, відповідні найбільшим власним значенням. Залишаючи розмірності тільки з корисною інформацією, ми отримуємо простір ознак, в якому кожне зображення вихідної вибірки представлено в узагальненому вигляді. У цьому, дуже спрощено, і полягає ідея алгоритму. Далі, маючи на руках деяке зображення, ми можемо відобразити його на створений заздалегідь простір і визначити до якого зображення навчальної вибірки наш приклад розташований найближче. Якщо він знаходиться на

відносно великій відстані від усіх даних, то це зображення з великою імовірністю взагалі не належить нашій базі.

2.6 FisherFace

Попередній алгоритм базується на припущенні, що, за загальновідомими ідеалізованими умовами, варіація в середині класу лежить у лінійному підпросторі простору зображень. Отже, класи опуклі і, отже, лінійно відокремлювані. Один може виконувати зменшення розмірності за допомогою лінійної проєкції і при цьому зберігати лінійну роздільність. Це сильний аргумент на користь використання лінійних методів для зменшення розмірності, принаймні, в задачі розпізнавання облич. коли хтось прагне нечутливості до умов освітлення. Оскільки навчальний набір позначений, має сенс використовувати цю інформацію для побудови більш надійного методу зменшення розмірності простору об'єктів. Використання специфічних для класу лінійних методів зменшення розмірності та простих класифікаторів у зменшених просторах можливостей, дозволяє отримати кращі показники розпізнавання, ніж за допомогою методу лінійного підпростіру або Eigenface методу. Лінійний дискримінант Фішера є прикладом методу класифікації. Цей метод вибирає W таким чином, що співвідношення розсіювання між класами та розсіювача всередині класу максимізоване. Нехай матриця розсіювання між класами визначається як

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

а матриця розсіювання всередині класу визначається як

$$S_W = \sum_{i=1}^c \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T$$

Де μ_i - середнє зображення класу X_i , а N_i - кількість зразків класу X_i .
 Якщо S_w не є одиничним, оптимальним проєкція W_{opt} вибрана як матриця з ортонормальних стовпців, що максимізує співвідношення визначника матриці розсіювання між класами проєктованих зразків до визначник матриці прогнозованих зразків розсіювання всередині класу.

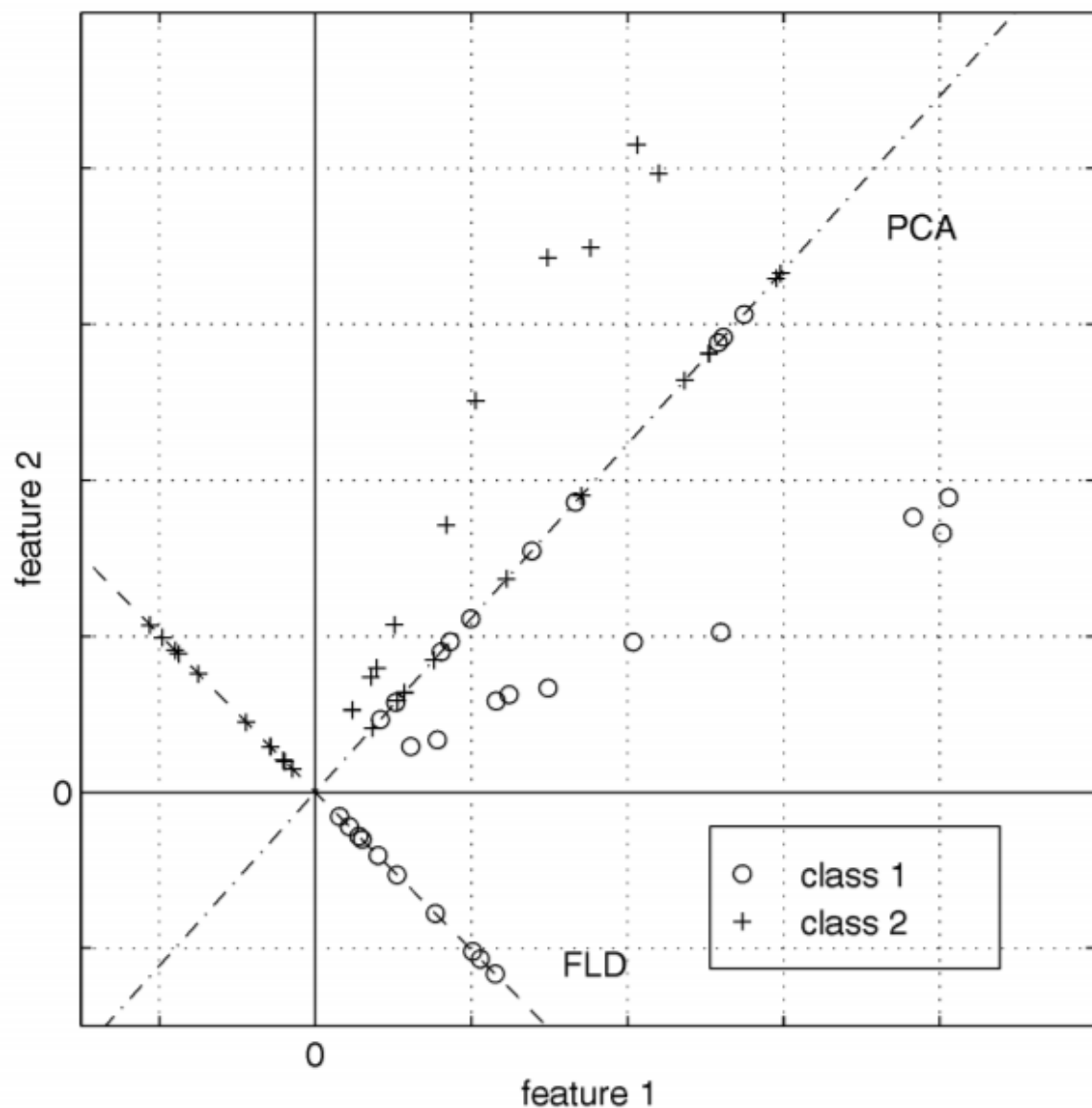


Рисунок 4 Порівняння PCA та FLD (із статті Eigenfaces vs. Fisherfaces: Recognition)

Рисунок 4 демонструє порівнянням PCA і FLD для двокласної задачі, в якій кожен клас випадково збурений у напрямку, перпендикулярному до лінійного підпростору.

Розділ 3

У цьому розділі буде розглянуто процес розробки власної реалізації програмного забезпечення з розпізнавання обличь, використовуючи аналітичну інформацію подану раніше. Основними критеріями до програмного забезпечення буде можливість розпізнавати обличчя в реальному час. Тобто програма має можливість отримувати відео потік з зовнішнього апаратного елементу. Наприклад, камери ноутбуку. Другим не менш важливим параметром є необхідність розпізнавання обличь різних людей.

Важливо зазначити, що в даній реалізації не було цілі розробити програмний застосунок, що може визначити стать, може ідентифікувати або просто знайти в кадрі більше одного обличчя, може знайти різницю між живими обличчями, та обличчями з фото або з екрану мобільного телефону, ноутбуку тощо.

3.1 Інструментарій

Для реалізації застосунку було обрано мову програмування C# та платформу .NET Core. C# - це мова програмування загального призначення з багатьма парадигмами, що охоплює статичну типізацію, сильну типізацію, лексичний масштаб, імператив, заяву, функціональність загальні, об'єктно-орієнтовані (на основі класів) та компонентно-орієнтовані дисципліни програмування. C # був розроблений Андерсом Хейлсбергом та компанією Microsoft у 2000 році в рамках його ініціативи .NET, а згодом затверджений як міжнародний стандарт Ecma (ECMA-334) у 2002 році та ISO (ISO / IEC 23270) у 2003 році.

Обрана мова програмування є досить популярною, тому має реалізацію багатьох відомих бібліотек та фреймворків, які знадобляться для реалізації програмного застосунку з розпізнавання обличчя людини в реальному часі.

В якості IDE було обрано Microsoft Visual Studio Enterprise 2019. Ця середа розробки є найбільш популярною для роботи з мовою C# та платформою .NET в цілому, адже містить дуже багатий інструментарій, гнучкі можливості до розробки, форматування та аналізу коду та, що є дуже важливим про роботі над застосуванням з розпізнавання обличь – засобами дебагу та взаємодії з сторонніми програмами та апаратними частинами комп'ютера.

Так як застосунок має використовувати камеру ноутбуку для пошуку та ідентифікації обличчя було вирішено розробляти застосунок з графічним інтерфейсом. Наразі, на платформі .Net найоптимальнішим варіантом для цього є Windows Presentation Foundation (WPF).

Windows Presentation Foundation (WPF) - це безкоштовна графічна підсистема (фреймворк) з відкритим вихідним кодом (подібна до WinForms), але більш нова та актуальна, розроблена корпорацією Майкрософт для візуалізації користувацьких інтерфейсів у додатках на базі Windows. WPF використовує DirectX і намагається надати послідовну модель програмування для побудови додатків. Він відокремлює користувацький інтерфейс від знес логіки і нагадує подібні XML-орієнтовані об'єктні моделі, такі як реалізовані в XUL та SVG.

У якості платформи для розробки застосування на C# з використанням WPF технології графічного інтерфейсу було обрано платформу .NET Core – платформу з відкритим кодом, а саме найактуальнішу її реалізацію - .Net Core 3.1. Переваги використання .NET Core очевидні – ця платформа є кросплатформерною, тобто розроблений додаток можна з легкістю перенести на будь яку іншу платформу чи операційну систему окрім Windows.

Використанні бібліотеки

System.Drawing – бібліотека, що є частиною платформи .NET, але має встановлюватись окремо так як не входить в стандартний набір бібліотек даної платформи. Її встановлення було необхідне через використання класу *Bitmap*,

що надає можливість представляти зображення з фрейму (не важливо, обличчя це чи просто загальний план) у вигляді двійкової карти.

Emgu.CV – величезна бібліотека, реалізація OpenCV для платформи .NET та мови C#. Ця бібліотека з відкритим кодом. Вона містить багато основних функцій та об'єктів для роботи з комп'ютерним зором. Реалізація на C# є досить гнучкою і містить достатньо функціональних можливостей для реалізації застосування для розпізнавання обличь в реальному часі. Головні використані класи з бібліотеки: *FisherFaceRecognizer* та *EigenFaceRecognizer* – класи, що реалізують методи розпізнавання обличь, описані раніше. Містять методи Train та Predict. Метод для тренування приймає на вхід список зображень обличчя, та список відповідних ідентифікаторів особи. У реалізації в якості ідентифікатора використовується просто ім'я.

3.2 Реалізація

По-перше, необхідно створити нове рішення з проектом типу WPF на .NET Core 3.1. IDE створить проект з вже існуючим елементом вікна MainWindow. З ним і будемо працювати. Цей клас буде основою застосунку для розпізнавання обличь, адже в ньому буде реалізована основна логіка програми. В першу чергу необхідно розробити простий UI у файлі MainWindow.xaml.

Для того, щоб активізувати камеру пристрою та виводити зображення з неї на застосування використовується клас VideoCapture. Цей клас є частиною бібліотеки для роботи з комп'ютерним зором Emgu.CV, тож його одразу необхідно проініціалізувати в конструкторі класу.

Інтерфейс користувача

Так як для реалізації було обрано застосунок типу WPF, інтерфейс користувача представляє собою десктопний застосунок. У WPF побудова інтерфейсу можлива двома способами. Перший – через код на C#. Перевагою

такого підходу є зрозумілий і звичний, об'єктно-орієнтований, метод створення елементів інтерфейсу користувача. Другий – використання XAML коду для побудови інтерфейсу. В даному програмному застосунку інтерфейс реалізовано саме за цим методом. Це дає змогу відділити частину представлення інтерфейсу та частину логіки. До того ж, Visual Studio 2019 Enterprise має зручні інструменти для роботи з компонентами користувацького інтерфейсу побудованого на XAML. Наприклад, можна використовувати drag-and-drop функціональність для створення елементів, змінення розмірів елементів через UI – IDE сама сгенерує необхідний код.

Інтерфейс застосунку буде представляти собою одне вікно, в якому одразу буде транслюватися відеопотік за камери пристрою. Крім того, у вікні буде присутня кнопка для додавання нової особи, якщо обличчя знайдено в кадрі і його немає в базі або якщо програма невірно ідентифікувала обличчя у кадрі. Таким чином можна буде тренувати модель «на льоту».

Логіка

Головними методами класу будуть RetrainRecognizer який дозволить тренувати клас для розпізнавання обличч при ініціалізації (запуску застосування) та при додаванні нового обличчя та ProcessFrame – метод, який викликається для кожного кадру з відео. Він буде відповідати за пошук обличчя на кадрі і якщо воно знайдено, передавати фрагмент кадру з обличчям на метод розпізнавання - RecognizeFace.

RetrainRecognizer

Цей клас відповідає за тренування моделей та ініціалізацію (переініціалізацію) основних компонентів системи. По-перше він перевіряє чи дійсно існує файл з даними про каскад Хаара. Якщо так, то вчитує його та ініціалізує класифікатор каскаду Хаара.

Після цього він перевіряє коректність вказаних шляхів до файлів з фотографіями для навчання (шаблонів) та відповідного їм текстового файлу з

інформацією про особу, яку ідентифікує фотографія (відповідність назви файлу з фото обличчя та імені людини. Якщо шлях вірний, але файлів не існує (можливо при першому запуску програмного застосунку на новому пристрої) – створює пустий текстовий файл.

Далі формуються два списки – перший з фотографіями, а другий з відповідними їм іменами людей, що на них зображено. Ці два списки надаються рекогнайзеру для тренування. Особливість цього методу полягає в тому, що він не тільки викликається при ініціалізації класу, а й при кожному додаванні нової особи до списку. Тобто тренування відбувається в процесі роботи з програмним застосунком.

ProcessFrame

Цей метод викликається кожного разу, як на відео з’являється новий кадр. Тобто для кожного кадру на відеопотоці буде викликано цей метод для його попередньої обробки. Кількість кадрів в відеопотоці залежить від технічних характеристик камери (для більшості це 30 кадрів в секунду) та налаштуванні класу VideoCapture. При необхідності кількість кадрів можна зменшити – це буде мати вплив на швидкодію застосування (актуально для машин із слабкими технічними характеристиками).

Так як цей метод викликається на кожен фрейм – основна його задача визначити чи є обличчя на кадрі чи ні. Якщо є – то він передає відповідальність за його розпізнавання іншому методу, а якщо ні – то не передає. Це звісно поліпшує навантаження на систему, адже їй непотрібно розпізнавати обличчя на кожному фреймі.

Цей метод спочатку оброблює кадр з використанням сірого фільтру. Потім, за допомогою бібліотеки Emgu.CV та проініціалізованого раніше класифікатору каскаду Хаара, знаходить на зображенні об’єкти, що можуть бути обличчями людини. На виході після роботи класифікатора ми маємо вже готові прямокутники, які вказують на положення обличчя на кадрі. Якщо

обличчя знайдено, то метод малює прямокутник на екрані та викликає метод для обробки обличчя з метою його розпізнавання.

RecognizeFace

Цей метод є основним ядром застосунку, адже саме тут відбувається розпізнавання обличчя та пошук його власника. Незважаючи на це його реалізація є не складною і досить зрозумілою. Тут просто викликається метод Predict на об'єкті рекогнайзеру, що було проініціалізовано раніше. Ця функція відповідає за передбачення, яке надає модель на основі тренувальних даних, наданих їй раніше. Потім результат зіставляється із списком імен персон і ім'я виводиться на користувацький інтерфейс застосування.

Висновки

В результаті дипломної роботи було проведено аналіз існуючих рішень з розпізнавання обличчя на відеопотоці. Були проведено порівняння так аналітика їх переваг. Розглянуто сучасні методи пошуку обличчя на кадрі (фото) та його ідентифікації. Проведено якісне порівняння та опис сильних і слабких сторін того чи іншого підходу. Наразі існує досить багато різних методів та підходів. І виділити з них більш доміновані або домінуючі складно, адже кожен має свої переваги так недоліки. Використання того чи іншого методу напряду залежить від поставленої задачі, бюджету, технічних обмежень системи.

За результатами аналізу було спроектоване власне програмне застосування, яке мало задачі розпізнавання обличчя в реальному часі з використанням апаратної частини комп'ютера. Застосунок було реалізовано, протестовано та відлагоджено. В результаті розроблене програмне забезпечення відповідає вимогам, що були сформовані на початку.

За результатами розробки, було досягнуто мети роботи. За допомогою сучасних методів та підходів до розпізнавання обличчя, було реалізовано застосунок на платформі .NET. Після реалізації, за допомогою тренування

моделі та тренування, аналізу ефективності і швидкодії ми дійшли висновку, що цей застосунок є ефективним у вирішенні поставленої задачі.

Література

1. P. Viola and M.J. Jones, «Rapid Object Detection using a Boosted Cascade of Simple Features», proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2001), 2001
2. P. Viola and M.J. Jones, «Robust real-time face detection», International Journal of Computer Vision, vol. 57, no. 2, 2004., pp.137–154
3. Р.Гонсалес, Р.Вудс, «Цифрова обробка зображення», ISBN 5-94836-028-8, 2005. – 1072 с.
4. Yan ˘Sochman, Jiřı Matas, «AdaBoost», Center for Machine Perception, Czech Technical University, Prague, 2010
5. Yoav Freund, Robert E. Schapire, «A Short Introduction to Boosting», Shannon Laboratory, USA, 1999., pp. 771-780
6. D. Pissarenko (2003). Eigenface-based facial recognition
7. "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projectio" - Peter N. Belhumeur, Joao~ P. Hespanha, and David J. Kriegman
8. <https://habr.com/ru/post/68870/>
9. www.emgu.com/wiki/index.php/Main_Page
<https://ru.wikipedia.org/wiki/OpenCV>
10. <https://github.com/opencv/opencv/tree/master/data/haarcascades>
11. http://www.sigma-is.ru/articles/art_ab_nic_5_2015.html
12. <https://habr.com/ru/company/synesis/blog/238129/>
13. <http://robocraft.ru/blog/computervision/264.html>
14. <https://en.wikipedia.org/wiki/Eigenface>
15. <https://www.bytefish.de/blog/fisherfaces.html#algorithmic-description>
16. <https://habr.com/ru/post/133826/>
17. <https://towardsdatascience.com/video-facial-expression-detection-with-deep-learning-applying-fast-ai-d9dcfd5bcf10>

18. <https://www.codeproject.com/Articles/239849/Multiple-Face-Detection-and-Recognition-in-Real-2>
- 19.. https://camera-sdk.com/p_6767-how-to-implement-face-detection-in-c-sharp.html
20. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
21. <https://www.mygreatlearning.com/blog/real-time-face-detection/>