

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

**Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра системного аналізу та кібербезпеки**

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Системний аналіз»

СПЕЦІАЛЬНІСТЬ

124 «Системний аналіз»

Форма навчання: денна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему «Аналіз систем комп'ютерного розпізнавання обличч»

здобувача Романюка Володимира Олександровича _____

(підпис)

**Науковий керівник: кандидат фізико-математичних наук,
доцент, Гладка Ю.А.**

(підпис)

**Робота допущена до захисту перед екзаменаційною
комісією з атестації здобувачів вищої освіти (ЕК)**

**Завідувач кафедри: доктор фізико-математичних наук,
професор Джалладова Ірада Агаверді-кизи**

(підпис)

Київ 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
СПЕЦІАЛЬНІСТЬ

«Системний аналіз»
124 «Системний аналіз»

ПОГОДЖЕНО

Керівник проектної групи (гарант)
освітньо-професійної програми
д.фіз.–мат.н., проф.Джалладова Ірада
Агаверді-кизи

(підпис)

2024 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри
д.фіз.–мат.н., проф.Джалладова Ірада
Агаверді-кизи

(підпис)

2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Романюка Володимира Олександровича
очної (денної) форми навчання

на підготовку кваліфікаційної магістерської роботи
на тему «Аналіз систем комп'ютерного розпізнавання облич»
Тему затверджено наказом ректора Університету від "____" _____ 2024 р. № _____

Кваліфікаційна магістерська робота виконується на нормативно-правових документах, експериментах у вільному доступі.

Розділ 1	АНАЛІЗ СФЕРИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ
Розділ 2	ОЦІНКА СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ, ОБГРУНТУВАННЯ ПІДХОДУ ДО РОЗРОБКИ ТА АЛГОРИТМІВ ШЛЯХОМ ПРОВЕДЕННЯ ПРАКТИЧНОГО ЕКСПЕРИМЕНТУ
Розділ 3	ПРАКТИЧНИЙ ЕКСПЕРИМЕНТ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ, ОЦІНКА РІШЕНЬ, ОГЛЯД РИЗИКІВ ТА ПОШУК МОЖЛИВОСТІ ВДОСКОНАЛЕННЯ СИСТЕМ

Об'єкт дослідження:	Комп'ютерне розпізнавання облич.
Предмет дослідження:	Методи розпізнавання облич та їх імплементація на мові Python.
Мета кваліфікаційної магістерської роботи:	Аналіз та порівняння ефективності моделей розпізнавання обличчя на основі OpenCV (cv2) та MTCNN. Оцінка точності, швидкодії та надійності цих моделей, а також виявлення можливих обмежень та шляхів їх подолання.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

Розділ 1	
	Провести огляд поточного стану розвитку машинного навчання у сфері розпізнавання облич, аналіз останніх досліджень, систем і підходів для вирішення завдання виявлення і розпізнавання облич. Визначити переваги та недоліки існуючих підходів. Проаналізувати правові аспекти використання систем розпізнавання облич в Україні та ЄС.
Розділ 2	
	Провести огляд та оцінку системи розпізнавання облич за допомогою бібліотеки cv2 у мові програмування Python. Провести огляд та оцінку системи розпізнавання облич за допомогою алгоритму MTCNN. Визначити особливості та переваги цього підходу.
Розділ 3	
	Провести експеримент з розпізнавання облич за допомогою системи cv2 у мові програмування Python та системи MTCNN. Оцінити ефективність та точність цього методу. Зібрати та проаналізувати результати експерименту. З'ясувати можливості вдосконалення системи MTCNN для розпізнавання облич. Визначити основні ризики та шляхи їх уникнення.

**Завдання підготував
науковий керівник**

Ю. А. Гладка

(підпис)

(ініціали, прізвище)

«____» _____ 2024 р.

**Завдання одержав
здобувач**

О.В. Романюк

(підпис)

(ініціали, прізвище)

«____» _____ 2024 р.

Реферат

Кваліфікаційна магістерська робота містить 66 сторінок, 19 рисунків, список використаних джерел з 57 найменувань, додатки.

«Аналіз систем комп'ютерного розпізнавання облич»

Об'єктом дослідження кваліфікаційної магістерської роботи є комп'ютерне розпізнавання облич.

Предметом дослідження є методи розпізнавання облич та їх імплементація на мові Python.

Мета і завдання дослідження: аналіз та порівняння ефективності моделей розпізнавання обличчя на основі OpenCV (cv2) та MTCNN з метою визначення їхньої придатності. Завдання включають оцінку точності, швидкодії та надійності цих моделей, а також виявлення можливих обмежень та шляхів їх подолання.

Відповідно до поставленої мети визначені такі *завдання*:

- провести огляд поточного стану розвитку машинного навчання у сфері розпізнавання облич. Зробити аналіз основних досягнень та напрямків досліджень у даній області;
- проаналізувати останні дослідження, систем і підходів для вирішення завдання виявлення і розпізнавання облич. Визначити переваги та недоліки наявних підходів;
- проаналізувати правові аспекти використання систем розпізнавання облич в Україні та ЄС. Визначити основні проблеми та виклики, пов'язані з впровадженням таких систем;
- провести огляд та оцінку системи розпізнавання облич за допомогою бібліотеки cv2 у мові програмування Python;
- провести огляд та оцінку системи розпізнавання облич за допомогою алгоритму MTCNN. Визначити особливості та переваги цього підходу;
- провести експеримент з розпізнавання облич за допомогою системи cv2 у мові програмування Python. Зібрати та проаналізувати результати експерименту;

Практичне значення отриманих результатів. У магістерській роботі проведено дослідження деяких з існуючих систем виявлення і розпізнавання облич. Було проведено аналіз поточного стану технологій розпізнавання облич та особливості розвитку цих технологій в Україні, що надалі дасть можливість розвивати побудовані системи зважаючи на правові аспекти українського законодавства.

Практичні результати дослідження полягають у виявленні, що обидві моделі можуть бути використані у різних сферах застосування. Модель cv2 може бути використана для ситуацій, де важлива швидкодія та простота використання, тоді як модель MTCNN відмінно підходить для завдань, де вимагається висока точність розпізнавання обличчя в умовах з великими викликами.

Рік виконання кваліфікаційної магістерської роботи – 2024.

Рік захисту роботи – 2024.

Ключові слова: штучний інтелект, розпізнавання облич, система, модель, алгоритм, Python, cv2, MTCNN.

В і д г у к

про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми «Системний аналіз»
Романюка Володимира Олександровича

на тему: «Аналіз систем комп'ютерного розпізнавання обличчя»

Актуальність теми. В сучасному світі зростає значущість систем розпізнавання обличчя як інструменту для різноманітних застосувань, від безпеки до розваг та особистої ідентифікації. Актуальність цієї проблематики в контексті України полягає в потребі удосконалення систем безпеки, ідентифікації осіб у громадських просторах та відповідності сучасним стандартам безпеки даних. Саме аналізу систем комп'ютерного розпізнавання обличчя присвячена ця робота.

Позитивні риси кваліфікаційної роботи. В основу КМР покладено системний підхід до розв'язання проблем, що розглядаються, а також сучасні інформаційні технології, зокрема технології машинного навчання, системи комп'ютерного зору.

Наявність самостійних розробок автора. В ході роботи описано архітектуру та методи, що використовувались для побудови комп'ютеризованої системи розпізнавання обличчя, проведено експериментальне порівняння різних моделей розпізнавання обличчя. У роботі було вирішено завдання комп'ютеризованого аналізу розпізнавання обличчя на основі різних моделей для забезпечення надійності використання і мінімізації можливості обходу системи розпізнавання обличчя.

Цінність теоретичних висновків та практичних рекомендацій. Запропоновано рекомендації щодо подальшого розвитку систем розпізнавання обличчя, які включають покращення точності і швидкодії, розширення функціональності, забезпечення конфіденційності і безпеки даних, інтеграцію з іншими технологіями.

Наявність недоліків. Суттєвих недоліків не виявлено, але є окремі зауваження щодо питання вибору технологій систем розпізнавання обличчя, а також до стилю викладу, що не впливає на позитивну оцінку роботи.

Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК. В роботі обґрунтовано актуальність завдання, яке вирішилося в повному обсязі. Робота виконана на сучасному рівні та показує, що автор орієнтується у даному питанні.

Кваліфікаційна магістерська робота Романюка В. О. може бути рекомендована до захисту в ЕК. Робота відповідає вимогам, що висуваються до кваліфікаційних робіт на здобуття освітнього ступеня магістр, заслуговує позитивної оцінки, а її автор, Романюк Володимир Олександрович, заслуговує на присвоєння йому освітнього ступеня магістр зі спеціальності 124 «Системний аналіз», галузі знань 12 «Інформаційні технології».

Науковий керівник

доцент кафедри комп'ютерної математики та
інформаційної безпеки Інституту інформаційних
технологій в економіці

канд. фіз.-мат.наук, доцент



Гладка Ю.А.

20 травня 2024 р.

Рецензія

на кваліфікаційну магістерську роботу
здобувача вищої освіти
Романюка Володимира Олександровича

*Тема **Аналіз систем комп'ютерного розпізнавання облич***

Актуальність теми кваліфікаційної магістерської роботи і доцільність її розроблення

Задача виявлення і розпізнавання облич на сьогодні являється актуальною. Багато компаній звертається до цього підходу. До того ж автоматизація процесів полегшує функціонування компанії взагалі. Розглянуті питання ефективності і надійності використання систем виявлення і розпізнавання. Актуальність теми полягає в тому, що такі системи є поширеними і їхній аналіз являється ефективним способом розвитку.

Якість проведеного дослідження

Робота проводилася значною мірою самостійно. Графік консультацій не порушувався. Завдання на роботу виконано повністю. Під час виконання магістерської роботи студент глибоко вивчив питання виявлення і розпізнавання облич, технології які використовуються для рішення відповідних питань, їхні сильні та слабкі сторони.

Позитивні риси кваліфікаційної магістерської роботи

Тема даної роботи присвячена аналізу наявних рішень. У роботі проведено практичні експерименти націлені на аналіз вразливостей відповідних систем та запропоновані покращення. Практичне значення отриманих автором роботи результатів полягає в можливості їх впровадження і використання за реальних обставин.

Зауваження

Недоліки роботи :

- 1) в роботі проаналізовані рішення створенні лише за допомогою мови програмування Python;
 - 2) нерозкриті питання вибору технологій систем розпізнавання і виявлення облич.
- Але названі недоліки не знижують цінності виконаної роботи

Практична значимість висновків і рекомендацій

Результати роботи дають можливість оцінити доцільність вибору того чи іншого підходу для виконання конкретних задач. Було розроблено ПЗ яке дає можливість отримати швидкий результат і має ймовірність подальшого застосування для виконання простих задач у даній сфері.

Рецензент:

Директор ТОВ «Кабельний завод»

Сокол-Пруський Ян Георгійов



ЗМІСТ

<u>ВСТУП</u>	3
<u>РОЗДІЛ 1 АНАЛІЗ СФЕРИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ</u>	7
1.1 Поточний стан розвитку машинного навчання у сфері розпізнавання облич	7
1.2 Аналіз останніх досліджень, наявних і доступних систем і підходів для розв'язування задачі виявлення і розпізнавання облич.....	19
1.3 Правові аспекти використання системи розпізнавання облич в Україні та ЄС	26
<u>РОЗДІЛ 2 ОЦІНКА СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ, ОБГРУНТУВАННЯ ПІДХОДУ ДО РОЗРОБКИ ТА АЛГОРИТМІВ ШЛЯХОМ ПРОВЕДЕННЯ ПРАКТИЧНОГО ЕКСПЕРИМЕНТУ</u>	30
2.1 Огляд та оцінка системи розпізнавання облич cv2 python	30
2.2 Огляд та оцінка у системі розпізнавання облич MTCNN.....	34
<u>РОЗДІЛ 3 ПРАКТИЧНИЙ ЕКСПЕРИМЕНТ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ, ОЦІНКА РІШЕНЬ, ОГЛЯД РИЗИКІВ ТА ПОШУК МОЖЛИВОСТІ ВДОСКОНАЛЕННЯ СИСТЕМ</u>	40
3.1 Експеримент з розпізнавання облич за допомогою системи cv2 python....	40
3.2 Експеримент з розпізнавання облич за допомогою системи MTCNN	46
3.3 Результати експерименту та шляхи вдосконалення системи MTCNN для розпізнавання облич	55
<u>ВИСНОВКИ</u>	59
<u>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</u>	61

ВСТУП

Актуальність теми. Сьогодні великі міста не мають місць, де б не було обладнано традиційними системами відеоспостереження - відеокамерами. Вони допомагають розслідувати інциденти, але мало що роблять для їх запобігання.

Біометричні відеоідентифікаційні системи наразі перебувають у швидкому розвитку. Ця технологія дозволяє, порівнюючи зображення, отримані камерами, з тисячами записів у базах даних різних структур і відомств за різними параметрами, знайти необхідну ідентифікацію особи.

В сучасному світі зростає значущість систем розпізнавання обличчя як інструменту для різноманітних застосувань, від безпеки до розваг та особистої ідентифікації. Критичний аналіз існуючих методів, зокрема, моделей на базі OpenCV (cv2) та MTCNN, виявляє як їхні переваги, так і обмеження. Актуальність цієї проблематики в контексті України полягає в потребі удосконалення систем безпеки, ідентифікації осіб у громадських просторах та відповідності сучасним стандартам безпеки даних.

Але всі ці передові рішення в основному використовуються урядами або високотехнологічними компаніями. Що можуть досягти звичайні люди і використовувати для своїх щоденних зручностей чи дослідницьких цілей? Які це рішення і які їхня сильна і слабка сторони? Розв'язування цих питань лягли в основу магістерської кваліфікаційної роботи, основною проблематикою якою було обрано дослідження та створення системи розпізнавання обличчя, шляхи обходу цих систем, дослідження їх переваг та недоліків за умови використання звичайного технічного забезпечення, доступного студенту.

Аналіз останніх досліджень та публікацій. Останні дослідження та публікації в працях українських та зарубіжних дослідників Goodfellow I., Kurakin A., Parkhi O., Redmon J., Everingham M., Esler T., Schroff F., Kalenichenko D., Bishop C., Deng J., Dong, W., Socher R, Sun, Y., Wang, X., Tang, X., Фрідман Ю., Черниш О., Михайленко В, Денисенко С., Гаврилюк І., Попов Д., Іванов О., Романов П., Russel P., Norvig P., Koller D. продовжують підтверджувати важливість систем розпізнавання обличчя, особливо в контексті застосування штучного інтелекту та комп'ютерного зору. Проаналізувавши дані у їх роботах вибір системи розпізнавання обличчя на основі OpenCV (cv2) та MTCNN (Multi-task Cascaded Convolutional Networks) для виконання магістерської роботи є досить актуальним і обґрунтованим з кількох причин:

- Ефективність: MTCNN є однією з найефективніших моделей для виявлення та розпізнавання обличчя в реальному часі. Вона забезпечує високу точність та швидкодію, що особливо важливо для багатьох застосувань.
- Наявність в OpenCV: OpenCV (cv2) - це потужна бібліотека для обробки зображень та комп'ютерного зору, яка надає різноманітні інструменти для роботи з обличчям. Це робить використання MTCNN зручним і доступним.
- Різноманітність застосувань: Системи розпізнавання обличчя на основі MTCNN та OpenCV можуть бути використані в різних сферах, включаючи безпеку, медицину, розпізнавання емоцій, автоматизовану аутентифікацію.
- Можливості дослідження: Використання цих технологій для експериментів дозволяє швидко прототипувати та тестувати різні ідеї в галузі розпізнавання обличчя.

Базуючись на даних наявних у працях вище названих авторів можна зробити висновок, що вибір системи розпізнавання обличчя на основі cv2 та MTCNN залишається актуальним та обґрунтованим для виконання експерименту з огляду на їх ефективність, доступність та широкий спектр застосувань.

Мета і завдання дослідження. Аналіз та порівняння ефективності моделей розпізнавання обличчя на основі OpenCV (cv2) та MTCNN з метою визначення їхньої придатності. Оцінка точності, швидкодії та надійності цих моделей, а також виявлення можливих обмежень та шляхів їх подолання.

Для досягнення поставленої цілі потрібно вирішити взаємопов'язані завдання:

1. Провести огляд поточного стану розвитку машинного навчання в сфері розпізнавання облич. Зробити аналіз основних досягнень та напрямків досліджень у даній області;
2. Проаналізувати останні дослідження, систем і підходів для вирішення завдання виявлення і розпізнавання облич. Визначити переваги та недоліки існуючих підходів;
3. Проаналізувати правові аспекти використання систем розпізнавання облич в Україні та ЄС. Визначити основні проблеми та виклики, пов'язані з впровадженням таких систем;
4. Провести огляд та оцінку системи розпізнавання облич за допомогою бібліотеки cv2 у мові програмування Python;
5. Провести огляд та оцінку системи розпізнавання облич за допомогою алгоритму MTCNN. Визначити особливості та переваги цього підходу;

6. Провести експеримент з розпізнавання облич за допомогою системи cv2 у мові програмування Python. Зібрати та проаналізувати результати експерименту;

Об'єкт дослідження: комп'ютерне розпізнавання облич.

Предмет дослідження: методи розпізнавання облич та їх імплементація на мові Python.

Методи дослідження: аналіз літературних джерел, машинне навчання, нейронні мережі, комп'ютерний зір, метод Віоли-Джонса, експериментальне порівняння різних моделей розпізнавання обличчя.

Теоретична, методична та практична значущість отриманих результатів. Отримані результати мають практичне значення для розробників систем безпеки, а також теоретичне значення для подальших досліджень у галузі розпізнавання обличчя. Вони надають важливі вказівки щодо вибору оптимального методу та моделі для проведення експериментів пересічному користувачу, що цікавиться даною темою.

Інформаційна база дослідження: включає широкий спектр джерел, від наукових публікацій до статистичних даних з вітчизняних та зарубіжних джерел, що забезпечує обґрунтування та достовірність отриманих результатів.

РОЗДІЛ 1

АНАЛІЗ СФЕРИ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ

1.1 Поточний стан розвитку машинного навчання у сфері розпізнавання облич

Останнім часом стає все цікавіше спостерігати за розвитком таких нових технологій, як розпізнавання обличчя. Понад 20 років минуло з моменту презентації перших розробок у цій галузі, і за цей час з'явилися нові перспективні можливості. При всьому цьому розвиток цієї технології зробив значний крок вперед, не зупиняється ні на хвилину і далекий від завершення. Для уряду розпізнавання обличчя є важливою складовою частиною системи безпеки та разючим пунктом бюджету. Розвинені країни зробили величезний прогрес у застосуванні цієї технології в різних галузях (див. Рисунок 1.1). Для журналістів це або панацея, або інструмент світової змови. Для бізнесу - інструмент або продукт.



Рисунок 1.1 - Концепція розпізнавання обличчя

Джерело: розроблено автором на основі [1]

Звичайно, розвиток технологій, зокрема розпізнавання обличчя, породжує низку викликів, серед яких питання конфіденційності та захисту особистих даних стоїть особливо гостро (див. Рисунок 1.2). [1]

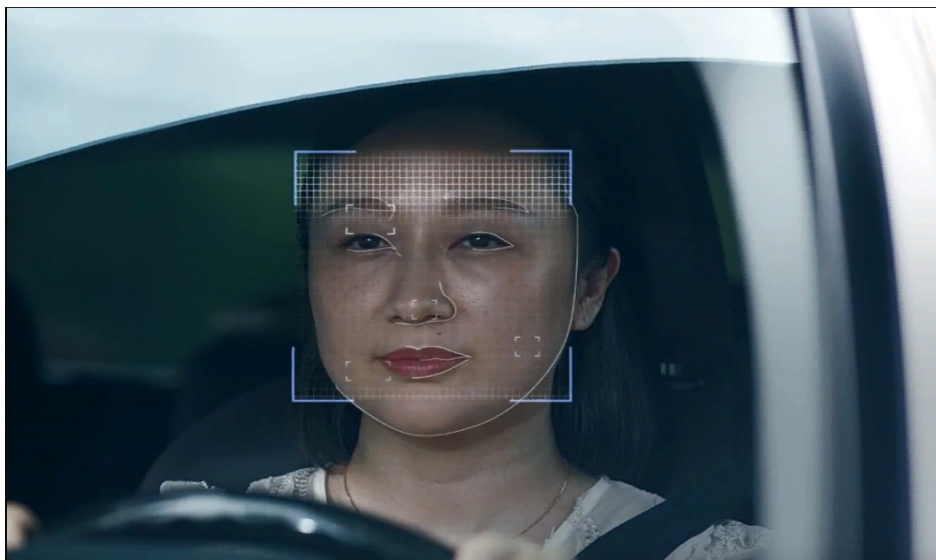


Рисунок 1.2 – Приклад захоплення обличчя

Джерело: розроблено автором на основі [1]

Чи є використання таких технологій порушенням приватності та обмеженням свободи людини - це питання, що залишається предметом дебатів. Також виникає занепокоєння стосовно можливості зловживання владою з метою шпигунства або переслідування політичних опонентів. Деякі експерти вважають, що загроза свободам людини та потенційна шкода переважають потенційні переваги технології. Багато міст у США вже заборонили використання розпізнавання обличчя на камерах поліцейських та ввели інші обмеження. Проте більшість занепокоєнь пов'язані виключно з відсутністю відповідного регулюючого законодавства та необхідністю якісного підходу до

використання таких технологій. У більшості країн Європи вже прийнято закони про захист персональних даних, які частково регулюють це питання. Додатковим важливим чинником у забезпеченні безпеки впровадження цих систем є правильний підхід до їх розгортання. Щоб захистити інформацію від перехоплення злоумисниками, передача даних повинна здійснюватися за допомогою шифрування протягом усього процесу передачі, а також персональні дані повинні бути анонімізовані. Це дозволяє забезпечити точну верифікацію біометричних та особистих даних лише на захищених серверах. Наприклад, один із провідних у цій галузі - Китай - встановив собі амбітну мету стати лідером за кількістю встановлених камер з функцією розпізнавання обличчя та вже встановив понад 170 мільйонів таких камер. Японія, з свого боку, впроваджує розпізнавання обличчя в автомобільному виробництві для виявлення ознак втоми водія та запобігання аваріям, у системах контролю доступу, створених у підготовці до Олімпіади 2020 року в Токіо, а також на робочих місцях для моніторингу стану працівників та підвищення їх продуктивності. Україна лише починає розвивати та впроваджувати технологію розпізнавання обличчя, проте можна стверджувати, що ця технологія вже з'явилася у корпоративному секторі, і багато підприємств починають впроваджувати її у свої системи безпеки. Особливо перспективним є використання розпізнавання обличчя в системах контролю доступу та обліку робочого часу персоналу, що дозволяє уникнути використання ідентифікаційних карток та підвищити рівень безпеки компаній.

Користувачі все частіше стикаються з технологією розпізнавання облич у своїх смартфонах, де використовується біометрична ідентифікація для розблокування пристрою та доступу до даних виключно власника. Камера

зберігає тривимірне представлення обличчя, що унеможливорює обман гаджету за допомогою фотографії.[3]

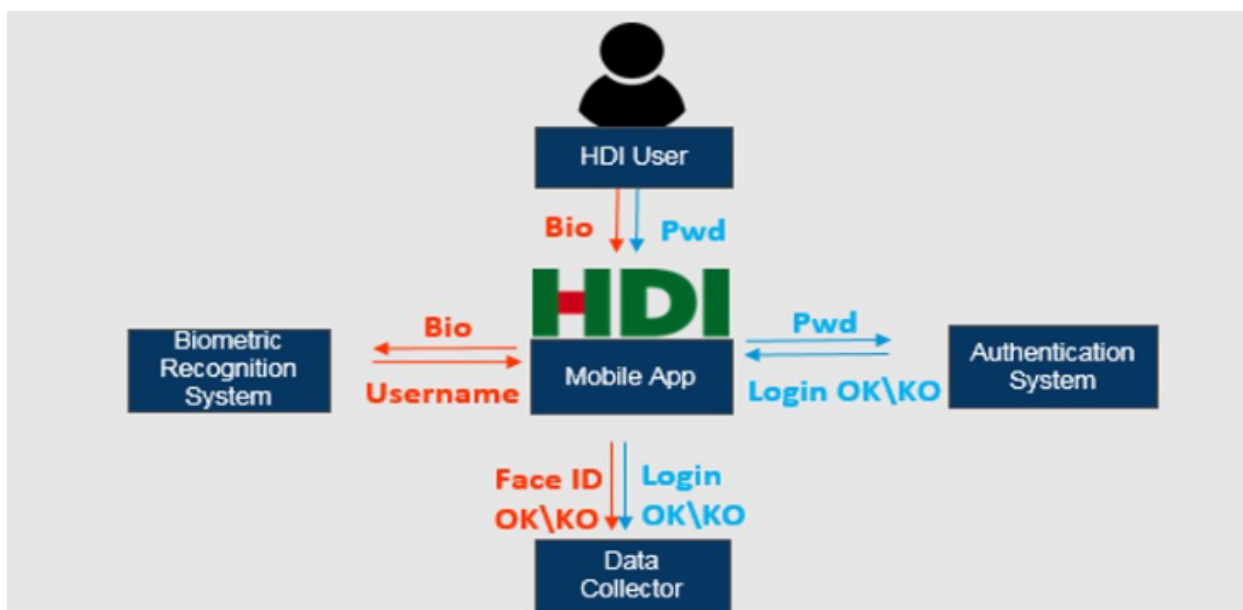


Рисунок 1.3 – Схема технології розпізнавання обличчя

Джерело: розроблено автором на основі [3]

Розпізнавання облич у реальному часі та у реальних умовах нерозривно пов'язане з системами відеоспостереження, де обличчя буквально "вириваються" з відеопотоку, знятого камерами.

Отримане відео може зберігатися в хмарному архіві. До камери підключений аналітичний модуль, який "вибирає" обличчя з відеопотоку, визначає стать і вік, і вводить дані в базу даних.

Суть технології полягає у використанні серверів з програмним забезпеченням або спеціальних камер з функцією розпізнавання обличчя для визначення вузлових точок на обличчі та вимірювання відстаней між ними, що дозволяє створити "відбиток обличчя".

Після отримання "відбитку обличчя" відбувається пошук у базі даних для порівняння обличчя з інформацією, що міститься у них. Швидкість

розпізнавання та порівняння з існуючими файлами може бути менше одної секунди, при цьому ймовірність помилки мінімальна. Система автоматично запам'ятовує всі визнані обличчя та записує їх в архів, а користувач з доступом додає додаткові дані. Коли особа з позначкою пройде перед камерою, система зафіксує це як важливу подію та надішле повідомлення зацікавленим користувачам.

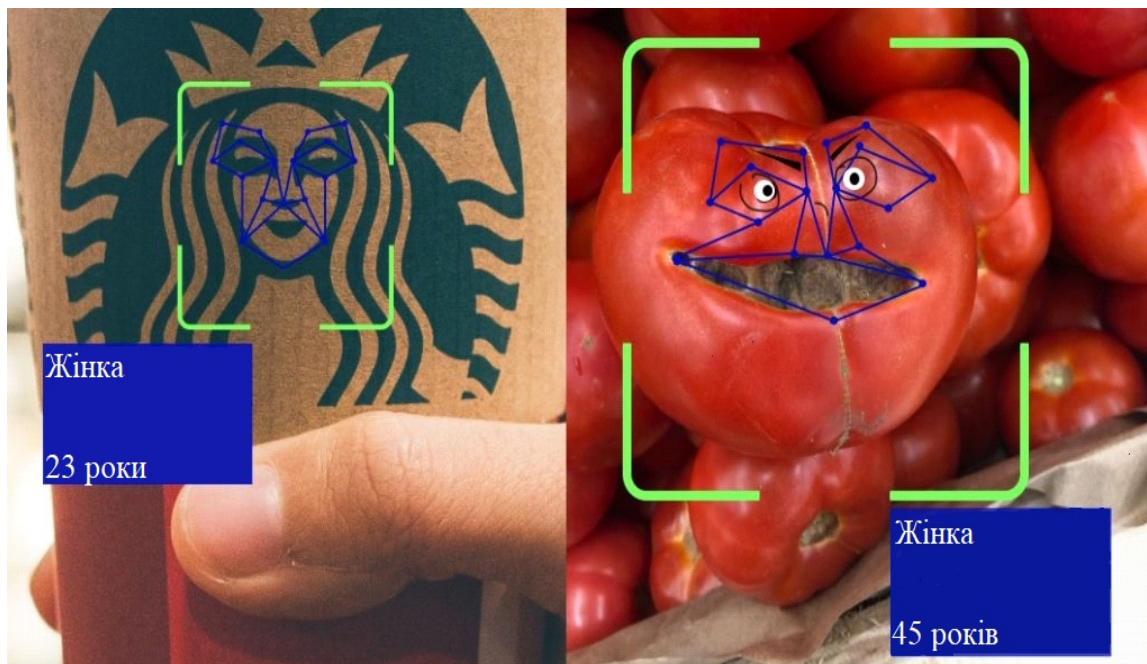


Рисунок 1.4 – False/Positive випадок

Джерело: розроблено автором на основі [3]

В контексті розпізнавання облич, виявлення - це ситуація, коли алгоритм, в принципі, розуміє, що це обличчя, а не помідор чи русалка з кружки від Starbucks (див. Рисунок 1.4). Спочатку для цього потрібна обчислювальна потужність, і лише потім можливий пошук в базі даних або її запам'ятовування.[3]

Інтернет - це парадоксальне місце: люди можуть одночасно хвилюватися про те, чи визначає кожна друга камера на вулиці їх особистість,

і щиро прагнути "розпізнавати обличчя інших людей на фотографіях в Інтернеті".

Сервіс FindFace.ru, який допомагав знаходити людей в соціальній мережі ВКонтакте за їх фотографіями, був заснований 18 лютого 2016 року. Серед іншого, завдяки йому, кожен міг знайти профілі дівчат, які знімалися у порнографічних фільмах. Дуже скоро сервіс став використовуватися для багатьох флешмобів з виявлення облич, які мали усі підстави, щоб ніколи не бути виявленими кимось. Скандал вибухнув, що працювало як вірусна реклама: технологія, яка стала основою сервісу, отримала ряд престижних нагород і викликала інтерес з боку держави та бізнесу. З 1 вересня 2018 року ця платформа припинила свою роботу. FindFace оголосив про закриття, так як його було перетворено компанією NtechLab на лінійку рішень для різних галузей бізнесу.[4]

Розпізнавання облич є досить типовим завданням. Тому багато компаній надають готові сервіси у вигляді хмарних API для якісного вирішення цих завдань. Окрім гігантів IT, таких як Microsoft та Google, розпізнаванням облич займаються також спеціалізовані міжнародні компанії. Їхні продукти швидко розвиваються і надають ще більше унікальних функцій, таких як ідентифікація облич і силуетів в натовпі.

Набагато складніше навчити нейронну мережу з нуля. Для цього потрібен великий і якісний набір початкових даних - десятки і сотні тисяч фотографій людей. Крім того, потрібні значні обчислювальні ресурси та знання штучного інтелекту та машинного навчання. Великі компанії мають для цього спеціальні додатки, що дозволяє вирішувати це питання значно швидше.

Компанія Tencent відкрила шлях у використанні технології в медичній галузі, створивши роботів, які аналізують симптоми пацієнтів, дивлячись на їх обличчя, і рекомендують відповідні методи лікування. Крім того,

розпізнавання облич може бути використане для ідентифікації рідкісних генетичних захворювань. Китай, без перебільшення, можна назвати лідером у цій галузі. У той час як у інших країнах подібні проекти лише у стадії планування, у Китаї такі системи вже давно успішно застосовуються в реальному житті.[5]

Крім того, можливості розпізнавання облич банками та платіжними системами розглядаються для забезпечення безпеки платежів. Наприклад, ПриватБанк разом з Visa використовують систему FacePay24, що дозволяє користувачам відмовитися від не тільки гаманця, але і від смартфона. Усі покупки будуть оплачуватися автоматично за допомогою системи розпізнавання облич. [6]

Серед великих світових компаній Amazon вже активно використовує це рішення у своїх автоматизованих магазинах Amazon Go, технології розпізнавання облич також використовуються в галузі "розумної" роздрібної торгівлі.

У роздрібній торгівлі все більше прозора. В цій галузі є три основні проблеми, які вирішує розпізнавання облич.

Перше місце займає крадіжка. Шахраї докучають магазинам, і часто ті ж самі люди в тій же самій мережі. Розпізнавання облич дозволяє виявити "блукаючих злодіїв" та інших людей, які раніше порушували порядок. Як тільки потенційний порушник, збережений у базі даних, увійде в магазин, охорона отримає сповіщення у месенджері або іншим зручним способом.

Потім виникає складність у роботі з постійними клієнтами. Просто недостатньо даних про покупки та дні народження для персоналізації пропозицій для важливих клієнтів та фанатів бренду. Розпізнавання облич може бути інтегроване з CRM - це програмне забезпечення, де менеджери вводять усю інформацію про всі транзакції організації. У разі злодіїв та VIP-

клієнтів розпізнавання облич працює приблизно таким же способом: обличчя вводиться у чорний або білий список, і коли воно знову з'являється, система надсилатиме сповіщення особі з доступом.

Зрештою, ідентифікація клієнтів у роздрібній торгівлі використовується для цільової реклами. Наприклад, деякі магазини мережі X5 Retail Group встановили камери для розпізнавання виразів обличчя та віку клієнтів. Аналізуючи ці дані, система відображає на моніторах в торговому залі товари, які можуть сподобатися людині. Інший яскравий приклад - це випадок магазину Lolli & Pops, великого кондитерського магазину у США. Система розпізнавання облич визначає постійних клієнтів та надсилає сповіщення на їх смартфони з продуктами, які їм можуть сподобатися, враховуючи індивідуальні уподобання, навіть алергії на їжу. Ще один вражаючий приклад використання технологій у роздрібній торгівлі - це магазини без продавців та кас. Наприклад, Alibaba Тао - це кафе та магазин самообслуговування, розташований у Ханчжоу. Тут продаються напої, закуски, продукти, іграшки, рюкзаки та інше. Під час покупки напоїв система камерного розпізнавання облич автоматично ідентифікує клієнта, підключається до їх онлайн-акаунту магазину та обробляє платіж. Покупці виходять через простір, обладнаний кількома датчиками, які визначають як клієнта, так і товари. Сканування працює навіть, якщо людина кладе покупку у кишеню чи сумку.

Системи відеоспостереження з розпізнаванням облич постійно займають все більш важливі позиції в світі.

Тільки 6 країн не мали доказів використання систем розпізнавання осіб. Але, швидше за все, це пов'язано з бюджетами/відсутністю технологій на відміну від суворого законодавства/протидії технології.[7]

Відомо, що тільки дві країни світу заборонили розпізнавання осіб - Бельгія та Люксембург. Останній мораторій у Марокко закінчився в грудні 2020 року, і докази технологій, що впроваджуються в містах, вже з'являються:

- 7 з 10 урядів використовують розпізнавання облич на масштабній основі;
- 70% поліціантів сил мають доступ до певної форми технології;
- 60% країн використовують систему розпізнання облич в аеропортах;
- Майже 20% країн розпізнають обличчя в своїх школах;
- Майже 80% країн використовують розпізнавання облич в рамках деяких своїх банківських чи фінансових установ;
- Близько 40% країн реалізували розпізнавання облич на деяких робочих місцях
- 20% країн мають розпізнавання облич на деяких автобусах в той час, як 30% застосовують його в поїздах / метро;
- Понад 40% країн використовували розпізнавання осіб в тій чи іншій формі, щоб спробувати відстежувати, контролювати або зменшувати передачу COVID-19.[8]

Топ-10 країн з найбільш поширеним і застосуванням розпізнавання обличчя.

Кожна країна отримала по 40 балів, що свідчить про те, що використання розпізнавання облич є найбільш поширеним та з посяганням на права особистості, а результати нижчими є більш поширеними та з несанкціонованим використанням. Найнижчий бал отримали такі країни:

Китай = 5 з 40: не дивно, що Китай очолює список, і його часто називають найбільшим постачальником технології розпізнавання обличчя.

Його уряд і поліція широко використовують цю технологію, часто з тактикою інвазивного спостереження. Наприклад, місто Сучжоу використовувало цю технологію, щоб публічно присоромити сімох людей, які залишили свої будинки в піжамах. Використовуючи розпізнавання облич, щоб ідентифікувати їх, місто опублікувало зображення на своєму обліковому записі WeChat. Один китайський парк навіть використовував розпізнавання облич, щоб запобігти крадіжці туалетного паперу. Та й діти також не уникають технологій, які загрожують конфіденційності, оскільки школи часто використовують технологію, щоб побачити, наскільки уважні учні. Якщо діти виглядають незосередженими, це відбивається на їхніх оцінках.

Росія = 9 з 40: поява Росії на вершині списку, мабуть, теж не дивує. Росія є ще однією країною, яка звертається до розпізнавання обличчя в різних областях. [9]

Об'єднані Арабські Емірати = 10 з 40: як і в багатьох інших топ-10 країнах, ОАЕ розгортають технології розпізнавання осіб у багатьох сферах, щоб допомогти «прискорити процеси» та «усунути шахрайство». Її використання в ОАЕ широко поширене: від використання технології для отримання доступів до державних послуг до реєстрації відвідуваності шкіл. Поліція в Абу-Дабі також модернізувала свої патрульні автомобілі, щоб включити розпізнавання облич в заявку, аби допомогти їм визначити «підозрілих і розшукуваних людей».

Японія, Індія та Чилі = 12 з 40: всі ці країни використовують розпізнавання облич у всіх категоріях, які були вивчені. Найбільше занепокоєння викликає використання Японією розпізнавання облич разом із повідомленнями громадян у соціальних мережах для відстеження злочинців. Національна комісія з громадської безпеки (NPC), яка відповідає за зберігання зображень облич близько 10 мільйонів японських громадян, також надала

поліції доступ до цієї бази даних, щоб вони могли використовувати її з розпізнавання облич. У Чилі до 2022 року більшість громадян отримали електронні посвідчення особи, які використовують розпізнавання осіб. А в Індії існує близько 16 різних систем розпізнавання облич, що використовуються в урядах Центральної та Державної систем, з подальшим розвитком.[10-13]

Австралія і Бразилія = 13 з 40: Австралійська поліція використовує технологію розпізнавання обличчя від Clearview (яка використовувала зображення соціальних мереж для створення бази даних). Щоправда, поліція штату Вікторія дистанціювалася від цих технологій. Незважаючи на те, що використання розпізнавання облич урядом і поліцією не настільки поширене в Бразилії, він знаходиться на підйомі. А технологія вже широко впроваджена в школах і громадському транспорті.

Аргентина = 16 з 40: У Буенос-Айресі, за повідомленнями, розпізнавання облич був пов'язаний з базою даних, яка включала неповнолітніх підозрюваних. Є також кілька історій про те, як людей незаконно затримували через неправильну ідентифікацію через систему розпізнавання осіб в режимі реального часу. Розпізнавання облич також знаходиться в більшості інших регіонів Аргентини (крім автобусів і шкіл).[13]

Франція, Угорщина, Малайзія, і Сполучене Королівство = 17 з 40: Всі чотири з цих країн мають технології розпізнавання осіб у кожній категорії. Франція, Велика Британія та Угорщина не мають відомого розпізнавання облич в школах, а Малайзія, здається, не встановила розпізнавання облич на автобусах. Проте вони досить широко використовують технологію розпізнавання облич в уряді, поліції, банківській справі та аеропортах та технологія також зростає в системах громадського транспорту. Проте, на

початку 2020 року, Французький Високий суд постановив, що розпізнавання обличчя не слід використовувати в середніх школах.[14]

Мексика і Сполучені Штати = 18 з 40: Зараз в школах Мексики немає відомих технологій розпізнавання осіб. Але є зростаюче або широке застосування у всіх інших районах барних автобусів, де системи були запропоновані. Існує, однак, все більше використання цієї технології в США, але автобуси, здається, не мають розпізнавання обличчя, встановлених в даний час.[15]

Румунія, Іспанія, і Тайвань = 19 з 40: Кожна з цих чотирьох країн все більше використовує технологію розпізнавання обличчя в більшості галузей, але всі вони також мають різні області, які залишаються недоторканими технологією в даний час. Наприклад, румунські системи громадського транспорту, іспанські школи та тайванські автобуси, схоже, не мають розпізнавання обличчя.

Казахстан, Швеція, Таїланд та Південна Африка = 20 з 40: У більшості випадків ці країни мають різну ступінь розпізнавання обличчя у кожній категорії. Так наприклад, в Швеції технологія розпізнавання обличчя була заборонена в школах.[16]

Компетентне і правильне використання розпізнавання обличчя перш за все спрямоване на покращення безпеки та комфорту. Люди зазвичай швидко починають довіряти технологіям, які, наприклад, допомагають уникати їм черг на футбольний матч, запобігають крадіжкам і хуліганству або допомагають їм заощаджувати при покупках. Все це, звичайно, потребує певного регулювання - саме тому приймаються закони про захист персональних даних.

У майбутньому, ймовірно, область розпізнавання обличчя в системах відеоспостереження буде регулюватися аналогічно до поточної практики роботи з розпізнаванням обличчя в Інтернеті. Прихильники конфіденційності

просто не завантажують забагато в мережу - часткове фіаско сервісу SearchFace свідчить про те, що така стратегія є ефективною. Незважаючи на всі виклики та питання щодо технології, ринок камер штучного інтелекту з розпізнаванням облич прогнозується досягти майже 8 мільярдів доларів менш ніж за три роки.

1.2 Аналіз останніх досліджень, наявних і доступних систем і підходів для вирішення задачі виявлення і розпізнавання облич

Наразі очевидно, що машинне навчання та системи розпізнавання облич вже не є невідомими поняттями. Ці системи дуже вдосконалені сьогодні і широко використовуються. Можна припустити, що передові рішення, здатні розпізнавати особу серед тисяч, доступні лише для урядових або корпоративних структур, і це буде частково правильно. Але для звичайних людей, не агентів ФБР, доступні обмежені функціональності систем розпізнавання облич - що можна легко створити, розвивати та отримувати.

Цю гіпотезу підтвердило багато фахівців з комп'ютерних наук, ентузіастів, аматорів тощо, які зацікавлені у технологіях та у виявленні своїх можливостей. Чимало людей працюють над удосконаленням існуючих алгоритмів або розробкою нових, використовуючи різні технологічні рішення, такі як MTCNN, Haar Cascades, MobileNet, FaceNet та інші.

Також існує безліч мов програмування, бібліотек і фреймворків, які підтримують машинне навчання та розпізнавання облич. Серед них C++ або C# з реалізаціями CV2 або популярний Python з Tensorflow та Keras.[17]

Незважаючи на випадкові помилки, точність розпізнавання облич та позитивний результат від використання таких технологій перевищує можливості людей. Китай незабаром запустить систему, яка зможе знайти певну особу серед 1,3 мільярда мешканців за 3 секунди з точністю 90%.

Але неминучі помилки, оскільки ідеального алгоритму розпізнавання облич не існує. Великі окуляри, накладна борода, кепка, велика швидкість руху, особливий макіяж: усе це може заплутати алгоритм. З іншого боку, використання таких хитрощів у звичайному світі - це складне завдання.

Розповсюдження нових технологій спричинить нові методи злому, спрямовані на особистість людини. Шахраї можуть намагатися або обдурити систему так, щоб вона зовсім не впізнавала їх, або сплутати їх з іншою особою.

Сучасні алгоритми добре впораються з перешкодами, такими як окуляри, кепки, рух голови. Зазвичай для успішного розпізнавання достатньо лише 70% обличчя. Наприклад, NTechLab успішно використовує свій алгоритм у Китаї, де медичні маски дуже поширені. Проте наявність окулярів і кепки на обличчі знижує точність розпізнавання. Вплив бороди або окулярів на обличчя вивчається Національним інститутом стандартів та технологій (NIST) США, який проводить тести на розпізнавання. І такі атрибути збільшують ймовірність помилкового визначення з одного разу на 50 000 до одного разу на 10 000.[18]

Очі є найбільш значущою частиною обличчя, не лише для людей, але й для машин, тому приховання очей за сонцезахисними окулярами значно зменшує можливість розпізнавання. Однак на сьогоднішньому рівні розвитку алгоритмів людина з окулярами та кепкою зможе приховатися від розпізнавання лише якщо вона постійно дивиться вниз. Втім, така поведінка викличе підозру. Окрім очей, алгоритм звертає увагу на геометрію обличчя, положення та форму носа, губ, і відстань між очима. Товста борода, яка

приховує губи, та інші способи спотворення параметрів обличчя ускладняють роботу алгоритму - людина з фальшивим носом, збільшеними губами та яскравим макіяжем, ймовірно, не буде розпізнана алгоритмом.[19]

Алгоритми розвинулися і почали враховувати принаймні 68 точок на обличчі - вони розташовані по контуру обличчя, визначаючи положення та форму підборіддя, очей, носа та рота, і відстань між ними. Отже, для того, щоб залишитися нерозпізнаним, шахраю доведеться якимось чином змінити всі ці точки.

Люди намагаються допомогти алгоритмам. До прикладу системи контролю доступу на стадіонах вимагають від охорони видалити всі перешкоди з обличчя інших людей - чи то кепку, чи бандану. Якщо обличчя захищене, система надсилатиме сигнал співробітнику, який попросить відвідувача відкрити обличчя та подивитися на камеру. Після того, як алгоритм виявить обличчя, він порівняє його з наявними базами даних, наприклад, з чорним списком вболівальників, і вирішить, чи пропустити людину через турнікет. Наприклад, така система відеоспостереження встановлена у багатьох стадіонах по всьому світі.

Масштабний експеримент щодо злому біометричного алгоритму ненавмисно було організовано компанією Apple. Вони оснастили свій флагманський iPhoneX системою Face ID, яка може розблоковувати пристрій за допомогою обличчя власника. Спроби обійти її почалися негайно і скоро були успішними. Наприклад, телефон було розблоковано близнюком власника iPhone, а в'єтнамська кібербезпекова фірма Вкав змогла зламати пристрій за допомогою маски, вартістю всього \$ 200.

Маски або професійний макіяж, що використовують актори, можна назвати найефективнішим способом обману. Наприклад, у тесті розпізнавання облич "Мітки обличчя на волі" [20] є контрольна пара фотографій акторки

Ніколь Кідман - у реальному житті і на зйомках одного з фільмів. Через макіяж людині важко впізнати її, не кажучи вже про алгоритми.



Рисунок 1.5 – Використання маски для обходу системи розпізнавання обличчя

Джерело: розроблено автором з використанням AI на основі [20]

Щоб переконати систему, що перед камерою знаходиться конкретна людина, потрібне ідеальне зображення. Це може бути фотографія людини на папері або цифровому пристрої, маска або 3D-модель обличчя (див. рисунок 1.5). Чим детальніше відтворені риси обличчя, наприклад, за допомогою 3D-комп'ютерної моделі, яка імітує м'язи обличчя, тим більше ймовірність обману алгоритму. За допомогою рухомих 3D-масок, створених із випадкових фотографій з Facebook, представники Університету Північної Кароліни змогли обдурити чотири з п'яти систем розпізнавання на конференції з безпеки Usenix у 2016 році.

Дослідники також випробували інші методи. У липні 2017 року блогер Григорій Бакунов, якого прозвали "bobuk", розповів про макіяж, який допомагає не тільки уникнути розпізнавання, а навіть видати себе за іншу

особу. Система накладає випадкові риси на фото і, цикл за циклом, погіршує можливість розпізнавання обличчя, пояснює Бакунов. Теоретично, один точний ризок може перетворити одну особу на іншу для системи. Але Бакунов не пішов на створення такого макіяжу - він вирішив не створювати нові можливості для зловмисників.

Ті самі вчені з Університету Карнегі-Меллона виконали схожий експеримент, в якому створили спеціальне покриття для оправ окулярів. Це покриття забезпечило невидимість людини перед алгоритмом у 80% випадків, як показано на малюнку 1.6. Також виявлено, що колір окулярів може ввести систему в оману, змушуючи її помилятися при ідентифікації людини на фото. Наприклад, алгоритм помилявся у 87,87% випадків, вважаючи білого чоловіка з окулярами за актрису Мілу Йовович, та в 88% випадків помилявся, вважаючи жінку південноазійського походження за чоловіка з Близького Сходу.

Біометричні системи ґрунтуються на статистичних методах, тому можуть виникати помилки. Помилки можуть бути двох типів: система може пропустити "знайомого" або відмовити "незнайомому" у доступі. Чутливість системи регулюється її власником. Система з високим порогом може впустити господаря в квартиру після кількох спроб, але не пропустить зловмисників. На відміну від неї, менш сувора система, наприклад, у колл-центрі банку, завжди розпізнає власника, але може частіше допускати "фальшиві позитивні" помилки.

Зловмисники намагаються обійти ці системи, наприклад, за допомогою макіяжу, що імітує звичайний зовнішній вигляд. В такому разі алгоритм розпізнавання може помилятися, а рішення перекладати на людину, яка, в свою чергу, може частіше помилятися в ідентифікації. Цей метод обходу алгоритмів вже вивчається на прикордонному контролі у європейських та

американських аеропортах, де впроваджуються біометричні системи ідентифікації.

Однак розробники вже знають про такі хитрощі. Спеціальний макіяж, як і той, що був розроблений Бакуновим, працює лише з конкретним алгоритмом, для якого він був створений. Щоб імітувати іншу особу, шахраю потрібно поступово коригувати макіяж, але через кілька спроб алгоритм може заблокувати доступ. Так само ви можете тренувати алгоритм розпізнавання окулярів на прикладі експерименту від Карнегі-Меллона.

Можна впізнати спроби обману за допомогою фотографій, масок і заздалегідь записаних відеороликів, застосовуючи спеціальні технології, які визначають, чи насправді жива людина перед камерою. Ці технології вимагають від людини виконання певних дій, таких як посмішка, моргання або наближення обличчя до камери або смартфона. Комбінації цих системних перевірок випадкові, що робить неможливим попереднє записування тисяч відеороликів з потрібними комбінаціями дій.

Також відомо, що деякі системи розпізнавання облич мають проблеми з ідентифікацією людей з темною шкірою. Це стосується не лише систем штучного інтелекту, які керують автомобілями, але й інших сфер. Дослідники з Технологічного інституту Джорджії недавно перевірили вісім моделей штучного інтелекту, що використовуються в системах виявлення об'єктів, і виявили, що всі вони мають проблеми з розпізнаванням пішоходів з темною шкірою. Навіть після корекції на такі фактори, як час доби або наявність перешкод перед камерою, точність цих моделей все одно знизилася на п'ять відсотків при розгляді фотографій пішоходів з темнішою шкірою.

Ці результати свідчать про можливі ризики для життя і безпеки темношкірих пішоходів, особливо в контексті широкого застосування автономних автомобілів. Проблема полягає в тому, що алгоритми засновані

лише на даних, які вони мають, і не можуть розпізнати об'єкти, яких немає в їх навчальних наборах. Тому потрібно включати більше зображень темношкірих людей у навчальні дані для систем штучного інтелекту, щоб підвищити їх точність і запобігти можливим негативним наслідкам.

Нещодавно дослідники з МАС і Стенфордського університету виявили неточності у роботі нейронних мереж. Вони встановили, що системи, що використовують штучний інтелект для обробки даних, часто роблять помилки у розпізнаванні осіб зі шкірою трохи темнішого відтінку: при ідентифікації жінок, програма, протестована експертами, допускала 20,8% помилок при розпізнаванні осіб з типом IV шкіри, 34,5% - з типом V, 34,7% - з типом VI. Помилки при розпізнаванні чоловіків зі світлою шкірою не перевищували 0,8%.

Хоча можна легко знайти багато досліджень про нові підходи і технології, мене найбільше цікавлять дослідження, які ставлять на порядок денний етичні питання. Це досить важливо для будь-якої технології, щоб продовжувати її розвиток. Проте, коли ми бачимо, що люди турбуються про свою приватність та конфіденційну інформацію, можна легко визначити, що такі технології швидко розвиваються.

Приватність в епоху мобільних додатків передачі даних охоплює як онлайн, так і офлайн сфери. Широке використання різноманітних додатків для збору особистої біологічної інформації створює нові проблеми в забезпеченні конфіденційності. З розвитком комп'ютерних алгоритмів, Інтернету речей та інших технологій поріг збору інформації стає все нижчим, а комп'ютерна інформація стає все доступнішою, що призводить до проблем, таких як вторинний видобуток даних та недостатня конфіденційність.

У дослідженні щодо конфіденційності Чха виявила негативну кореляцію між турботою користувачів про конфіденційність інформації на

технологічній платформі та частотою використання засобів масової інформації (Cha, 2020). Мак-Найт провів дослідження на Facebook і виявив, що чим більше проблем з конфіденційністю в середовищі, тим менше людей готові продовжувати використовувати середовище через страх неправильного використання особистої інформації (McKnight et al., 2022).

У контексті великих даних, проблеми конфіденційності користувачів пов'язані з розпізнаванням обличчя і включають ризик збору та використання інформації про зображення обличчя без особистого знання або згоди, або ризик обміну або витоку персональних біометричних даних. Іншими словами, рівень конфіденційності людини визначається уважним відбором програм розпізнавання обличчя.

1.3 Правові аспекти використання системи розпізнавання обличчя в Україні та ЄС

Право на конфіденційність є одним з основних прав людини і включає захист особистих даних, включаючи аудіовізуальну інформацію, таку як фотографії, відеозаписи та інші. Як сучасні системи спостереження з використанням штучного інтелекту можуть порушувати це право, і як цьому протидіють країни Європи аналізує юрист Максим Войнов з Центру стратегічних справ УГСПЛ.

«Збір особистих даних може призвести до соціальної дискримінації та можливих зловживань», зауважує Йоханнес Каспар.

Втручання у приватність особи стає все поширенішим, з фотографуванням і відеоспостереженням на вулицях. Наприклад, в Києві, за програмою "Безпечне місто", встановлено понад 7 тисяч відеокамер, деякі з них оснащені розпізнаванням облич.

Соціальні мережі допомагають в цьому процесі, дозволяючи швидко ідентифікувати осіб, але це також вносить ризики, особливо коли сторонні компанії використовують ці дані без дозволу користувачів.

Під час використання системи розпізнавання облич, наприклад в аеропортах чи для розблокування смартфонів, ми маємо на увазі не лише її переваги, але й можливість порушення прав людини, зокрема щодо приватності.

Технологія розпізнавання облич може посилити недоторканість приватного життя, особливо в сфері правоохоронної діяльності, де вона може бути використана для масового моніторингу без достатніх підстав.

Реальне розпізнавання облич дійсно може мати позитивний вплив у різних сферах, зокрема в розшуку злочинців та в безвісти зниклих осіб. Україна використовує систему розпізнавання облич Clearview AI для ідентифікації загиблих російських солдатів у контексті протистояння збройній агресії Росії.[22]

Однак, використання штучного інтелекту може порушувати права людини, зокрема право на конфіденційність та захист персональних даних. Технології, подібні Clearview AI, здійснюють збір та обробку особистих даних без належного повідомлення і згоди особи, що суперечить міжнародним стандартам захисту даних, зокрема загальному регламенту про захист даних (GDPR).

Компанії, які використовують Clearview AI, мають ризик отримання значних штрафів від європейських країн за порушення правил захисту даних.

Також українське законодавство потребує удосконалення для забезпечення більших гарантій захисту приватності осіб, зокрема захисту даних, зібраних камерами відеоспостереження. Проект Закону про захист персональних даних, який розглядається у Верховній Раді України, має на меті заповнити існуючі законодавчі прогалини в цій сфері.

Штучний інтелект, зокрема системи розпізнавання облич, стає предметом гострої дискусії щодо його можливих негативних впливів на права людини, зокрема на право на приватне життя та захист персональних даних. Законодавці та правозахисні організації у всьому світі, зокрема в Європейському Союзі, висловлюють серйозну турботу щодо можливих ризиків, пов'язаних з використанням цих технологій.[24]

Одним із ключових аспектів, який викликає обурення та обмеження використання систем розпізнавання облич, є їхній потенційний негативний вплив на приватне життя та право на конфіденційність. Такі системи можуть призвести до масового спостереження за громадськими місцями, що порушує будь-яку концепцію приватності та створює загрозу для особистої свободи та демократії.

Організації, які борються за права людини, такі як Amnesty International, ведуть глобальну кампанію із заборони використання систем розпізнавання облич для масового стеження. Вони підкреслюють, що невибіркове масове спостереження завжди порушує право людини на недоторканність приватного життя та інші основні права.

Європейський Союз реагує на ці турботи, розробивши проект Закону про штучний інтелект, який має стати новою правовою базою та посилити регулювання розвитку та використання штучного інтелекту. Проект встановлює різні вимоги до розробки та використання систем штучного

інтелекту, зокрема систем розпізнавання облич, і передбачає суворі покарання за їхнє недотримання.

Зокрема, встановлюється заборона на використання технологій штучного інтелекту в біометричному спостереженні, а також розробляється система класифікації ризику для різних типів технологій ШІ. Такий підхід спрямований на захист основних прав людини та забезпечення високого рівня захисту персональних даних.

Рішення Європейського Суду з прав людини в справах, пов'язаних з захистом приватного життя та персональних даних, також має велике значення для регулювання використання систем розпізнавання облич та інших технологій штучного інтелекту. Ці судові рішення підтверджують необхідність суворого контролю та обмежень у використанні цих технологій з метою захисту прав людини та забезпечення демократичних цінностей.[25-31]

РОЗДІЛ 2

ОЦІНКА СИСТЕМИ ДЛЯ РОЗПІЗНАВАННЯ ОБЛИЧ, ОБГРУНТУВАННЯ ПІДХОДУ ДО РОЗРОБКИ ТА АЛГОРИТМІВ ШЛЯХОМ ПРОВЕДЕННЯ ПРАКТИЧНОГО ЕКСПЕРИМЕНТУ

2.1 Огляд та оцінка системи розпізнавання облич cv2 python

Маючи певний інтерес до мови програмування Python та значний досвід її використання, я вирішив обрати її як основу для своєї дипломної роботи. Python - це одна з найпростіших, але потужних мов програмування, що робить її чудовим вибором для початківців у програмуванні та має безліч можливостей для досвідчених інженерів.

Найшвидший шлях для створення першої системи розпізнавання облич з використанням Python - використати бібліотеку cv2 та алгоритм каскаду Хаара. cv2 є реалізацією відкритого програмного забезпечення Intel з назвою OpenCV, яка має великий спектр функцій.

Вибір технології OpenCV (cv2) для системи розпізнавання обличчя має декілька обґрунтованих причин.[47-50]

1. Широке використання та популярність: OpenCV є однією з найпоширеніших бібліотек комп'ютерного зору. Вона має велику спільноту користувачів та розробників, які активно співпрацюють і підтримують її. Це означає, що ви можете з легкістю знайти допомогу, документацію та

різноманітні ресурси для вирішення будь-яких проблем або виконання необхідних завдань.

2. Ефективність та продуктивність: OpenCV написана на C++ з інтерфейсом Python, що робить її дуже ефективною та продуктивною. Вона використовує оптимізовані алгоритми та може працювати з великими обсягами даних з високою швидкістю. Це особливо важливо для систем реального часу, де потрібна швидка обробка зображень.
3. Багатофункціональність: OpenCV має широкий набір функцій, пов'язаних з обробкою зображень та комп'ютерним зором, включаючи розпізнавання облич, виявлення об'єктів, відстеження руху та багато іншого. Це дозволяє створювати повноцінні системи розпізнавання облич на основі однієї бібліотеки без необхідності використання додаткових інструментів чи технологій.
4. Зручний інтерфейс для Python: Інтерфейс OpenCV для Python (cv2) має простий та зрозумілий синтаксис, що дозволяє легко взаємодіяти з бібліотекою та швидко розробляти програми на Python. Це особливо зручно для розробників, які вже знайомі з мовою програмування Python та хочуть швидко створити систему розпізнавання облич на її основі.

Отже, обираючи OpenCV для системи розпізнавання облич, ви можете розраховувати на широкі можливості, ефективність та зручний інтерфейс розробки, що сприяє успішній реалізації вашого проекту.

OpenCV була створена Гері Брадські у 1999 році з метою просування обчислювального зору та створення попиту на нові потужні комп'ютери від Intel. На сьогодні OpenCV є потужним інструментом, який включає функції виявлення облич, відслідковування та розпізнавання облич.

Хоча OpenCV має високий рівень успішності (89-96%) у виявленні облич, є певні обмеження. Наприклад, він ефективно працює лише при прямо

поверненому обличчю в камеру та при належному освітленні та якості зображення.

Зараз існують спеціалізовані рішення, спрямовані на вирішення цих обмежень. Алгоритми, такі як Haar Cascade (НА поруч), спеціально розроблені для виявлення облич та їх атрибутів на зображеннях. Такий підхід, заснований на машинному навчанні, дозволяє ефективно вирішувати завдання виявлення облич та їх атрибутів.

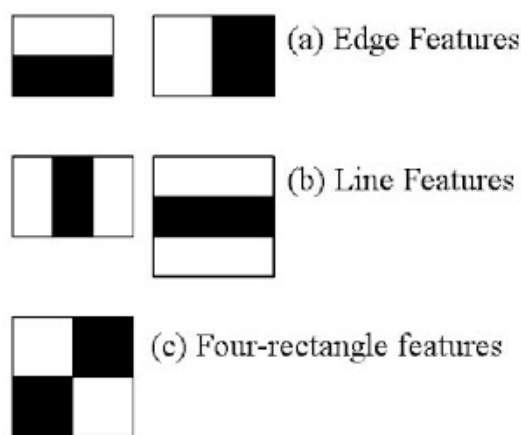


Рисунок 2 1 – Приклад Haar Cascade Features

Джерело: розроблено автором на основі [47]

Кожна ознака визначається як різниця між сумою пікселів під білим і чорним прямокутниками. Таким чином, кожне ядро може мати різні розміри та місця розташування, що робить можливим розрахунок великої кількості особливостей. Навіть вікно розміром 24x24 може мати понад 160000 можливих комбінацій. Для обчислення кожної ознаки модель повинна здійснювати розрахунки, враховуючи суму пікселів під біло-чорними прямокутниками. Ця проблема вирішується за допомогою методу інтегрального зображення, який зводить розрахунки до операцій з чотирьох пікселів, що значно прискорює

процес. Однак більшість з розрахованих особливостей можуть бути неактуальними.

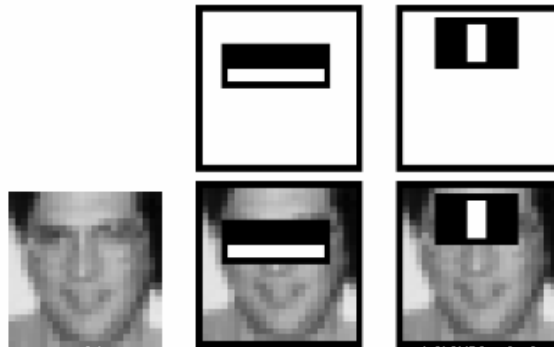


Рисунок 2..2 - Haar Cascade Features використання

Джерело: розроблено автором на основі [47]

У верхньому рядку зображення представлені дві корисні особливості. Перша з них виявляється в тому, що зона очей зазвичай має темніший тон, ніж зона носа та щік. Друга вибрана особливість використовує властивість того, що очі темніші за ніс. Проте такі ж прямокутники, накладені на щоки чи інші ділянки, не мають значення. Отже, вибір найкращих функцій з понад 160000 варіантів робиться за допомогою методу Адабуст.

Для цього кожна функція застосовується до всіх тренувальних зображень. Для кожної ознаки визначається оптимальне порогове значення, яке розділяє обличчя на позитивні та негативні класи. Очевидно, що будуть помилки та не класифікації. Обираються ознаки з мінімальною кількістю помилок, що означає їх високу точність в класифікації облич та їх відсутності на зображеннях. Процес навчання навіть коефіцієнтів індивідуальних ознак включає у себе ітеративне підвищення ваги неправильно класифікованих

зображень, що дозволяє покращувати точність моделі. Таким чином, після кількох ітерацій обираються найкращі функції.

Остаточний класифікатор формується як зважена сума слабких класифікаторів. Цей метод відомий як метод Адабуст. Навіть з використанням лише 200 функцій досягається точність виявлення обличчя на рівні 95%. Оригінальний детектор Пола і Майкла використовував близько 6000 функцій.

Отже, для аналізу кожного вікна розміром 24x24 застосовується 6000 функцій. Однак це може здаватися неефективним. Тому автори запропонували використовувати каскадний підхід. Замість застосування всіх 6000 ознак на кожному вікні, вони групуються за стадіями класифікації і застосовуються послідовно. Якщо вікно не пройшло першу стадію, воно відкидається, інакше проходить до наступної. Інші ознаки не розглядаються. Таким чином, вікна, які пройшли всі етапи, визначаються як обличчя.

Детектор, описаний у дослідженні, складався з понад 6000 ознак на 38 стадіях, з 1, 10, 25, 25 та 50 ознаками на перших п'яти стадіях. В середньому, як стверджують Пол і Майкл, на кожне вікно накладаються 10 ознак з 6000+.

2.2 Огляд та оцінка у системі розпізнавання облич MTCNN

Вивчивши поєднання OpenCV та Haar Cascade, можна сказати, що цей підхід може бути задовільним, оскільки метод працює та має прийнятні результати. Однак існує можливість розглянути інші методи для розпізнавання обличчя.

Одним із таких альтернативних методів є MTCNN (Multi-Task Cascaded Convolutional Neural Network), який представлений Кайпенгом

Чжаном у статті «Спільне виявлення обличчя та вирівнювання за допомогою багатозадачних каскадних згорткових мереж». Обираючи MTCNN, ми керуємось кількома вагомими перевагами.

В першу чергу, MTCNN популярний завдяки високим результатам, які він показує на різних наборах даних. Враховуючи його успішність, ми можемо впевнено використовувати цей метод у нашій системі для досягнення точних результатів.

Крім того, MTCNN може розпізнавати інші риси обличчя, такі як очі і рот, що дозволяє виявляти орієнтири. Це значно розширює можливості системи розпізнавання облич та робить її більш універсальною.

Структура MTCNN складається з трьох шарів: P-Net, R-Net і O-Net. Кожен з цих шарів виконує свою функцію у виявленні та розпізнаванні обличчя. Використовуючи цю складну структуру, MTCNN може забезпечити найвищу точність та надійність в роботі.

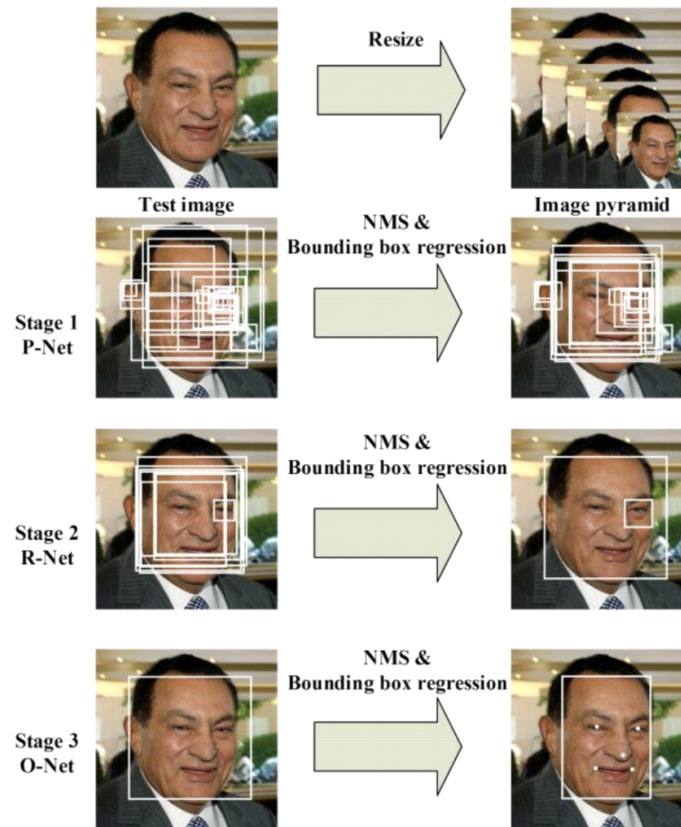


Рисунок 2.3 – Представлення ітерації MTCNN

Джерело: розроблено автором на основі [51]

Загалом, процес виявлення облич MTCNN передбачає створення піраміди зображень, застосування P-Net для вибору потенційних кадрів обличчя та коригування їх за допомогою регресії. Після цього R-Net та O-Net фільтрують результати та виводять орієнтири, що робить цей метод дуже ефективним у виявленні облич та їх рис.

Перша мережа P-Net є повністю згортковою мережею, яка використовується для генерації кандидатних вікон та векторів регресії рамок. Регресія обмежувальних рамок використовується для корекції кандидатних рамок, а потім використовуються не максимумні супресії для відкидання зайвих рамок.

Діаграма результату P-Net показана на Рисунку 2.4.

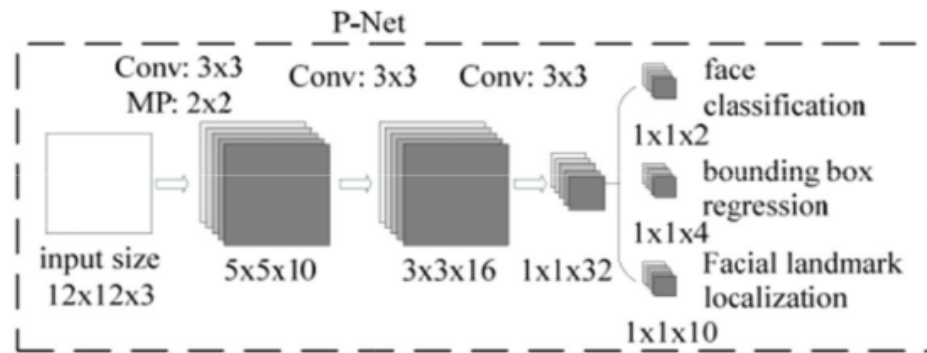


Рисунок 2.4 -P-Net структура

Джерело: розроблено автором на основі [52]

Результати P-Net відносно грубі, тому подальше налаштування виконується за допомогою R-Net. Діаграма структури R-Net показана на рисунку 2.5. Ця структура дуже схожа на P-Net. Вона буде ввімкнена в R-Net через кандидатне вікно P-Net, відкине більшість хибних вікон і продовжить використовувати регресію обмежувальних рамок та зведення методом немаксимумної супресії.[51-55]

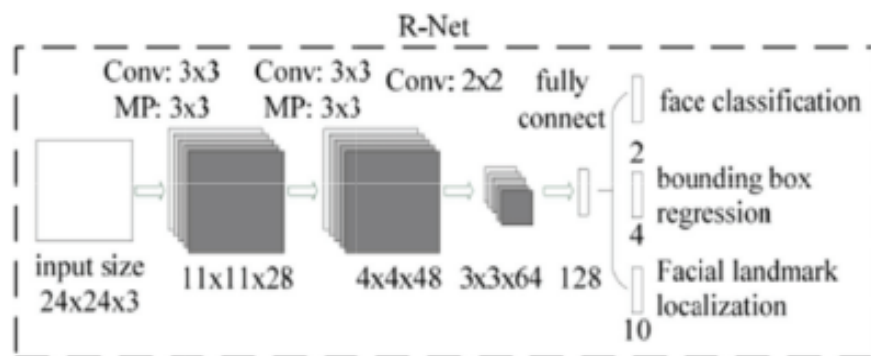


Рисунок 2.5 - R-Net структура

Джерело: розроблено автором на основі [52]

Нарешті, O-Net використовується для виведення кінцевого кадру обличчя та позицій точок ознак. Схоже на перші два кроки, відмінність полягає у генерації п'яти позицій точок ознак. Структура мережі O-Net показана на рисунку 3.

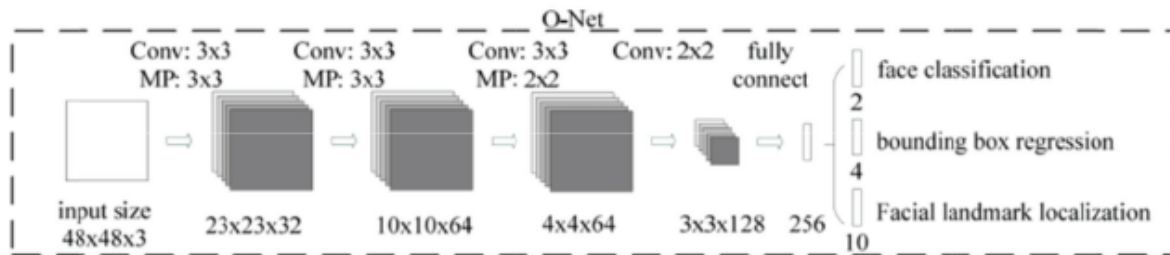


Рисунок 2.6 - O-Net структура

Джерело: розроблено автором на основі [52]

Кожна мережа в MTCNN має три частини виходу, тому втрата також складається з трьох частин. Для частини виявлення обличчя безпосередньо використовується функція втрати перехресної ентропії:

$$L_i^{\text{det}} = -(y_i^{\text{det}} \log(p_i) + (1 - y_i^{\text{det}})(1 - \log(p_i)))$$

У формулі P_i представляє ймовірність введення обличчя, а y_i^{det} представляє справжню мітку.

Для регресії рамок і вирішення п'яти позицій точок ознак усі проблеми є проблемами регресії, тому для визначення втрат використовується загальна евклідова відстань. Функція втрат регресії обмежувальних рамок:

$$L_i^{\text{det}} = ||\hat{y}_i^{\text{box}} - y_i^{\text{box}}||_2^2$$

Метод MTCNN (Множинний Класифікатор Трансформацій Нейронних Мереж) є потужним і популярним інструментом для виявлення облич у зображеннях. Використання цього методу у нашій дипломній роботі

для розробки системи розпізнавання облич може бути обґрунтоване з різних поглядів:

1. Ефективність: MTCNN є одним з найефективніших методів для виявлення облич. Він складається з трьох модулів: виявлення облич, виявлення ключових точок і класифікація. Кожен модуль оптимізований для точної роботи при мінімальних обчислювальних витратах.
2. Надійність: MTCNN відомий своєю високою надійністю в різних умовах зйомки, включаючи різке освітлення, зміну кута зйомки, а також наявність в зображенні різних облич із змішаними емоціями.
3. Масштабованість: MTCNN може легко масштабуватися для обробки великої кількості зображень з високою швидкістю. Це особливо важливо, коли мова йде про систему розпізнавання облич, яка має обробляти великий потік вхідних зображень.
4. Доступність: Існує багато відкритих реалізацій MTCNN на різних мовах програмування, що дозволяє легко інтегрувати цей метод у вашу систему розпізнавання облич.
5. Підтримка: MTCNN активно підтримується та розвивається спільнотою машинного навчання, що забезпечує постійне оновлення та вдосконалення його алгоритмів.

Загалом, обрання методу MTCNN для використання у вашій дипломній роботі з розпізнавання облич обґрунтоване його високою ефективністю, надійністю, масштабованістю, доступністю та підтримкою спільноти.

РОЗДІЛ 3

ПРАКТИЧНИЙ ЕКСПЕРИМЕНТ ВИЯВЛЕННЯ І РОЗПІЗНАВАННЯ ОБЛИЧ, ОЦІНКА РІШЕНЬ, ОГЛЯД РИЗИКІВ ТА ПОШУК МОЖЛИВОСТІ ВДОСКОНАЛЕННЯ СИСТЕМ

3.1 Експеримент з розпізнавання облич за допомогою системи cv2 python

На основі аналізу проведеного в Розділі 1 та 2 було прийнято рішення провести два експерименти для побудови власної системи розпізнавання обличчя. Першим експериментом з визначення обличчя було вибрано систему за допомогою Haar Cascade (далі - FR), реалізованого на Python. На зображенні вікна показана структура проєкту (див. Рисунок 3.1).



Рисунок 3.1 - Структура проєкту Haar Cascade

Джерело: розроблено автором

У папці каскадів знаходяться файли .uml - кожен з них містить перелік орієнтирів обличчя, які використовуються HCF. Папка зображень містить вкладені папки з фотографіями, що є навчальним набором даних для моделі. У даному підході відсутній окремий набір даних для тестування, оскільки розпізнавання осіб у реальному часі передбачає виконання всієї роботи системою. У папці міток знаходиться один файл, що містить список етикеток. У папці recognizers зберігаються підготовлені моделі, які потім можуть бути використані як попередньо навчені. Також є два .py файли: "faces.py", що є основною програмою, яка використовує вбудовану відеокамеру для захоплення відеопотоку, знаходить обличчя за допомогою ROI (Regions Of Interest) і на кінцевому етапі вирішує, чи присутнє відоме обличчя на кадрі. Другий - "faces-train.py", використовується для тренування моделі. Він використовує каскади, обрізає зображення, щоб залишити тільки обличчя на фотографіях, вибирає область інтересу, потім присвоює фотографіям етикетки і тренує модель. Нарешті, модель зберігається у файлі .uml. Це досить просто.

Зауваження, що це може бути занадто просто. На фото можна побачити, що мене розпізнає система, але модель вказує на ймовірність лише 73% (див. Рисунок 3.2). Це перевищує 50%, тому можна вважати це до певної міри успішним. Умова ймовірності повинна бути в межах від 60% до 95%. Зміни у мітках не вплинули на результати - ні на поліпшення, ні на погіршення. Крім того, система діє дещо дивно: чим далі я від камери, тим більш впевнено вона вказує на мою присутність.

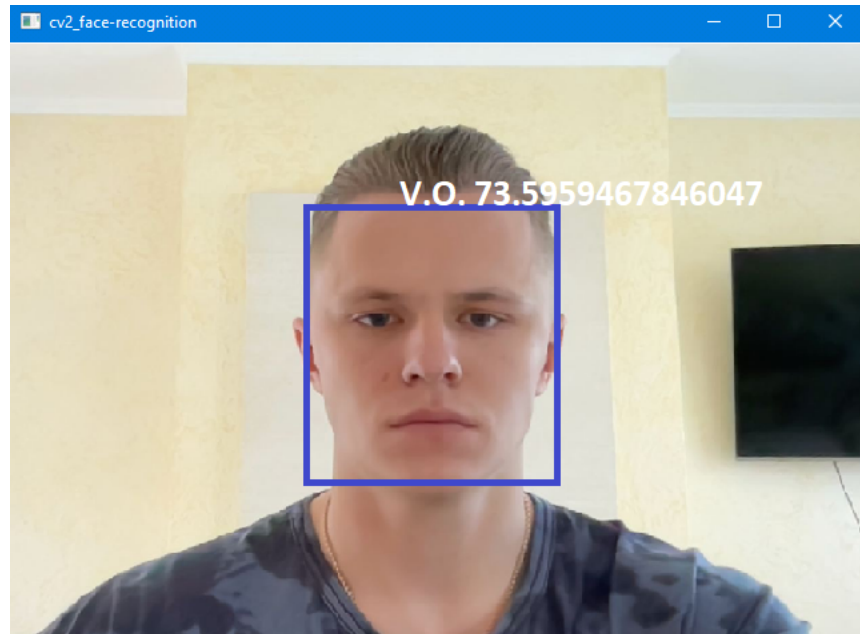


Рисунок 3.2 - Haar Cascade система розпізнавання облич в дії
Джерело: розроблено автором

Тут ми визначаємо класифікатор Наар і cv2 система розпізнавання облич для майбутнього навчання.

```
face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_alt2.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()
```

Потім ми змінюємо розмір зображень, перетворюємо їх у відтінки сірого і збираємо мітки.

```
pil_image = Image.open(path).convert("L")
size = (550, 550)
final_image = pil_image.resize(size, Image.ANTIALIAS)
image_array = np.array(final_image, "uint8")
```

```

        faces = face_cascade.detectMultiScale(image_array,
scaleFactor=1.5, minNeighbors=5)

```

В кінці ми зберігаємо області інтересу (ROIs) та мітки, тренуємо нашу модель і зберігаємо її.

```

for (x,y,w,h) in faces:
    roi = image_array[y:y+h, x:x+w]
    x_train.append(roi)
    y_labels.append(id_)
with open("pickles/face-labels.pickle", 'wb') as f:
    pickle.dump(label_ids, f)

recognizer.train(x_train, np.array(y_labels))
recognizer.save("recognizers/face-trainer.yml")

```

Тут ми імпортуємо навчену FR, мітки та починаємо захоплення відео з камери.

```

recognizer.read("./recognizers/face-trainer.yml")
with open("pickles/face-labels.pickle", 'rb') as f:
    og_labels = pickle.load(f)
    labels = {v:k for k,v in og_labels.items()}
cap = cv2.VideoCapture(0)

```

Потім ми по одному зчитуємо захоплені кадри, перетворюємо їх у відтінки сірого для більшої точності, і після цього модель намагається знайти область інтересу на кадрі.

```

while True:

```

```
ret, frame = cap.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5,
minNeighbors=5)
```

Якщо модель має впевненість більше 60%, вона відобразить прямокутник навколо обличчя, виведе ім'я особи та ймовірність того, що виявлене обличчя належить реальній людині.

```
id_, conf = recognizer.predict(roi_gray)
if conf>=60 and conf <= 95:
    cv2.putText(frame, name, (x,y), font, 1, color, stroke,
cv2.LINE_AA)
```

У роздумах про безпеку, я бачу два шляхи: ми можемо спробувати обійти систему у чесній грі, намагаючись приховати, або ж ми можемо спробувати зіпсувати тренувальні дані.

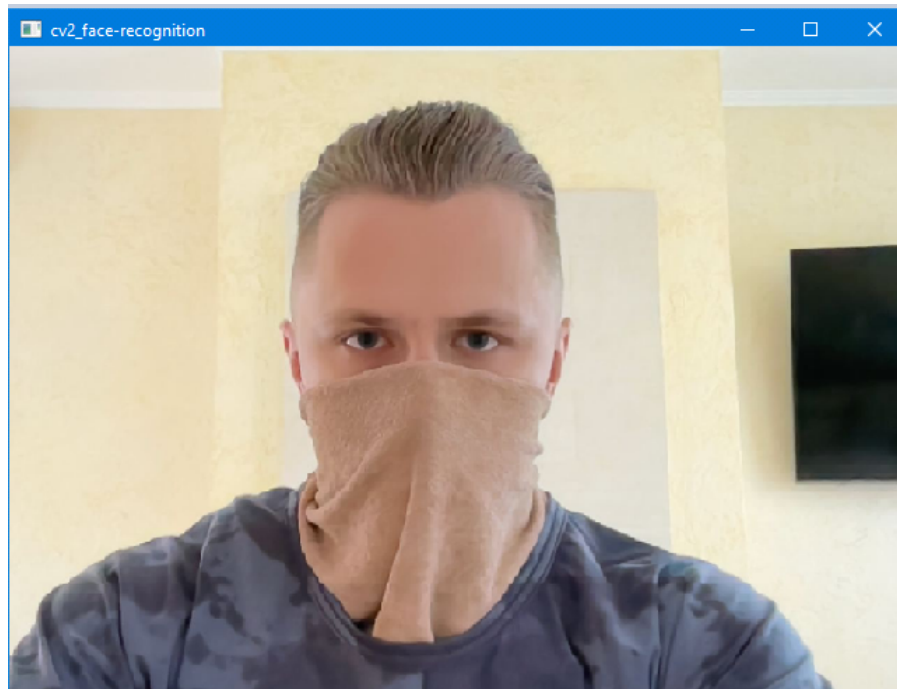


Рисунок 3.3 - Haar Cascade система розпізнавання облич не може розпізнати через маску

Джерело: розроблено автором

Як можна побачити, навіть звичайної маски вистачає, щоб приховатися від такого простого детектора облич (див. малюнок 3.3). На кадрі навіть обличчя не видно, не кажучи вже про можливість розпізнавання моєї особи. Після обробки тренувальних даних система може розпізнавати моє обличчя, але з нещодавніх даних не може впевнено визначити, хто це.

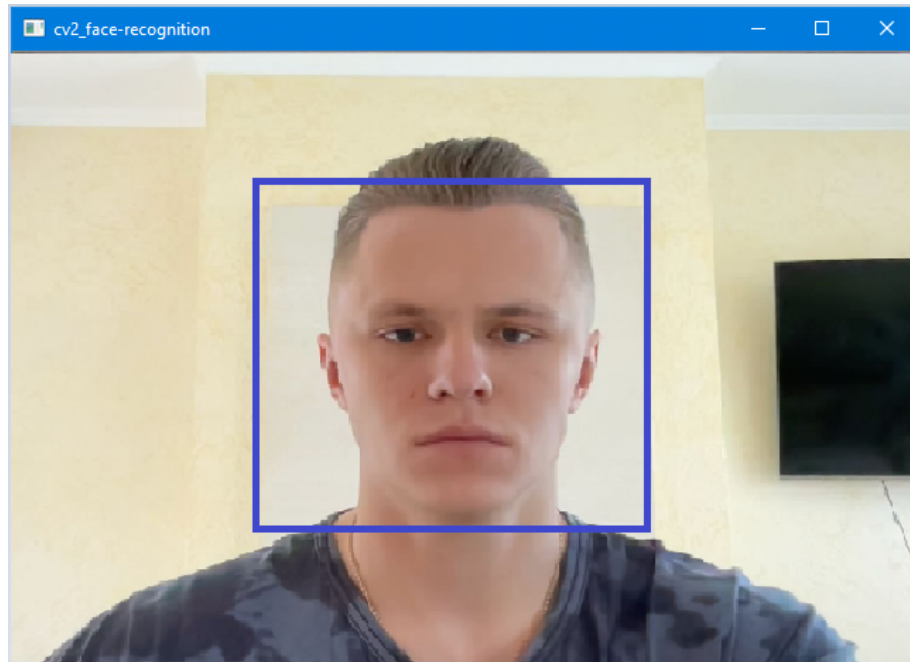


Рисунок 3.4 - Haar Cascade система розпізнавання облич з помилково обробленим тренувальним набором даних

Джерело: розроблено автором

На практиці, зазвичай набагато простіше реалізувати перший варіант - намагатися сховатися. Найгіршим випадком буде те, що система побачить ваше обличчя, але не зможе його розпізнати. Крім того, порушення збережених тренувальних наборів даних виявляється складнішим, оскільки вимагає доступу до системи, перенавчання моделі і затемнення всього процесу разом із самою новою моделлю.

3.2 Експеримент з розпізнавання облич за допомогою системи MTCNN.

Другий приклад є більш складним, тому він може бути більш ефективним. Труднощі починаються ще до самої кодувальної частини. Це передумови.

По-перше, нам потрібно встановити TensorFlow. Це бібліотека, яка широко використовується для глибокого навчання. Однією з найбільш корисних функцій є можливість використовувати графічний процесор разом із центральним процесором. Таким чином, можна досягти значного збільшення продуктивності. Однак важливо мати відеокарту, сумісну з CUDA. Існує офіційний посібник щодо перевірки вашої відеокарти і встановлення всіх необхідних драйверів і SDK.

Для реалізації я використав попередньо навчену нейронну мережу. Варто зазначити, що зміна ваги завдала більше шкоди, ніж користі: або система не розпізнавала моє обличчя зовсім на близькій відстані, або вона визнавала обличчя на моєму одязі, стінах тощо. Тому я вирішив залишити параметри незмінними. Ця модель була навчена з використанням набору даних «VGGFace2».

```
ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    try:
        result = detector.detect_faces(frame)
        x1, y1, w, h = result[0]['box']
    ...
```

```
except:
    pass
cv2.imshow('Face Detection', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()
```

Цей фрагмент коду демонструє основну частину сценарію: захоплення кадру, спробу виявити обличчя, виходити з програми при натисканні на "q". Коротко кажучи, він показує виняткові результати - система майже на 100% впевнена, що в кадрі присутнє обличчя.

Щодо експериментів, я розпочав з огляду попередніх досліджень (див. Рисунок 3.5). Це дослідження було проведено колишніми студентами. Вони подавали "зіпсовані" фотографії у свою мережу, намагаючись обдурити її.

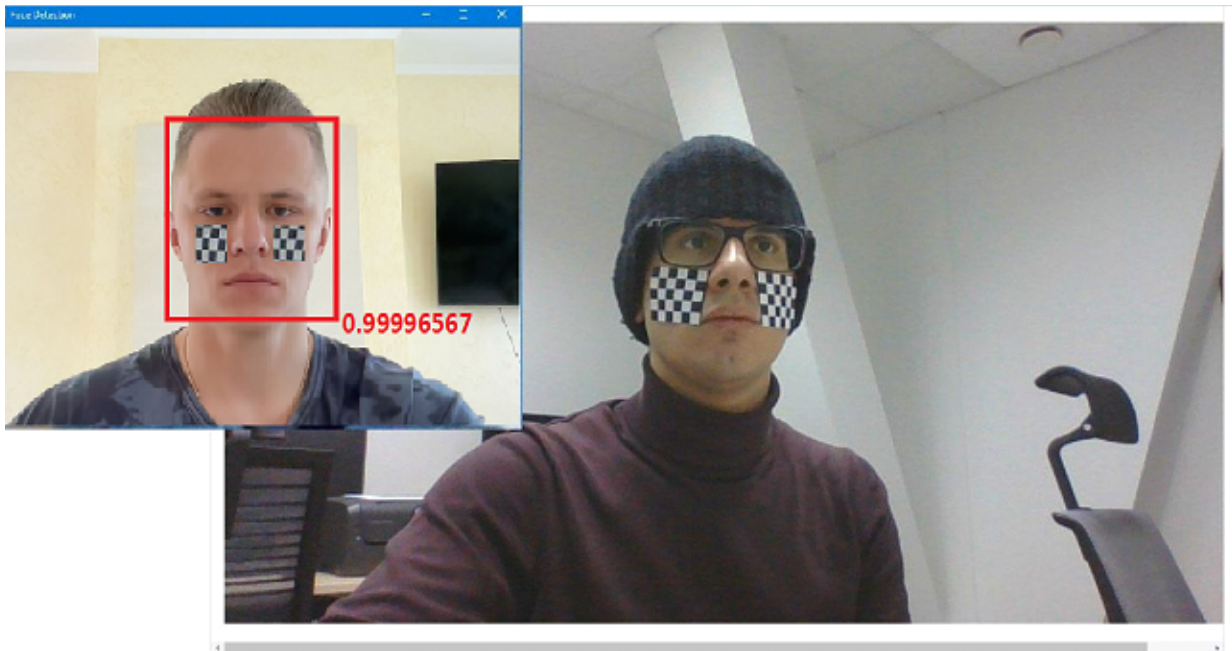


Рисунок 3.5 - Відтворення експерименту з атакою за допомогою адверсативного методу

Джерело: розроблено автором

Для мене це було безуспішним, моє обличчя все ще було помічено. Що правда, MTCNN став краще. Потім я намагався обдурити його, носивши сонцезахисні окуляри, маску та сонцезахисні окуляри з маскою. Крім комбінованого одягу, він добре впорався (див. Рисунок 3.6).

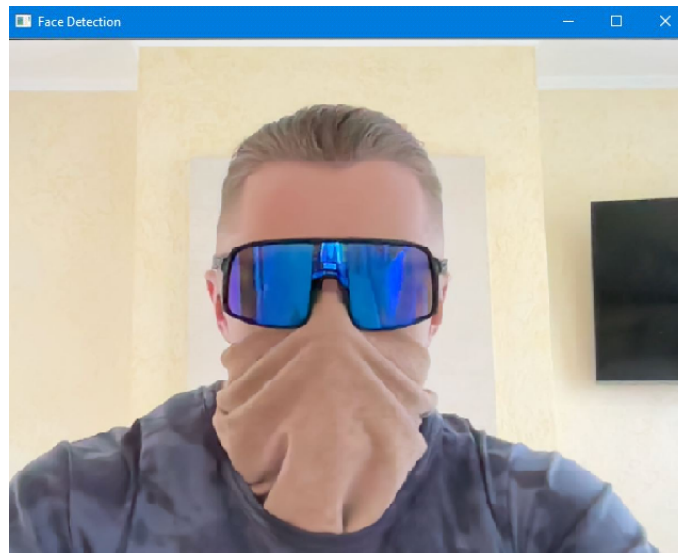


Рисунок 3.6 – Очевидний підхід був все ще не ефективним

Джерело: розроблено автором

Після цього я вирішив модифікувати експеримент з [6] і використав свої фотографії, закріплені на масці. Хоча він не зміг виявити обличчя на фотографіях через мій старий принтер, він більше не міг розпізнати реального мене (див. малюнок 3.7).

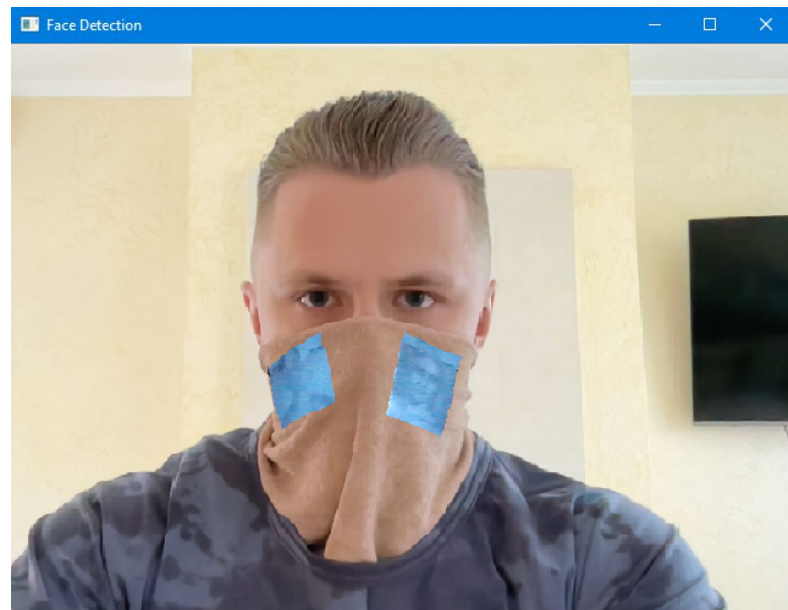


*Figure 3.7 – Наявність нестандартної маски може допомогти з розпізнаванням обличчя
Джерело: розроблено автором*

В кінці кінців, я спробував сховати своє масковане обличчя за допомогою фотографії розміром 3x4, що принесло 50% успіху (див. Рисунок 3.8).

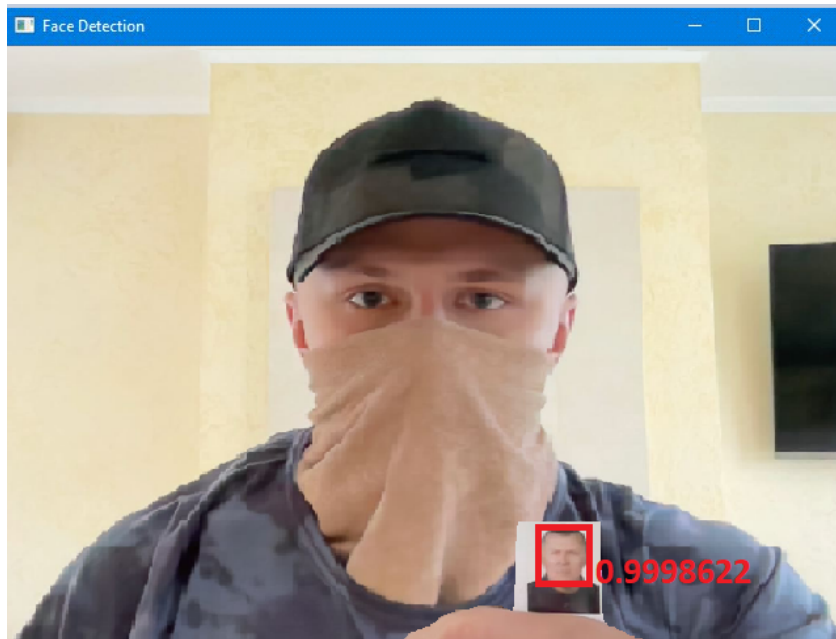


*Рисунок 3.8 - MTCNN визначає навіть маленьке зображення обличчя
Джерело: розроблено автором*

Немає причин для скарг на MTCNN. Незважаючи на труднощі, пов'язані з підготовкою вашої системи до нього, рішення MTCNN залишається потужним і зручним інструментом. Однак технології постійно розвиваються, так само як і MTCNN.

Я провів невелике дослідження і розробив наступний план. По-перше, ми можемо перейти до трохи вдосконаленої версії MTCNN, яка походить з бібліотеки Python `facenet_pytorch`. Хоча їх функціональні можливості практично однакові, остання показує значне поліпшення загальної продуктивності: приблизно 30 кадрів на секунду порівняно з 10 кадрами на секунду у попередньому MTCNN.

Ця зміна не впливає на здатність виявлення обличчя - обидва підходи показують стабільні чудові результати. Однак, більше кадрів на секунду у

режимі реального часу полегшує використання програми, що є великим плюсом з бізнес-точки зору. Крім того, це спрощує процес розробки, оскільки ви можете спостерігати результати вашої роботи швидше.

Єдиним передумовою для переходу до MTCNN facenet_pytorch є наявність бібліотеки PyTorch. Це потужний фреймворк для машинного та глибокого навчання на Python, який також може працювати з технологією TensorFlow. І знову, все це залежить від вашої відеокарти - чи вона сумісна з CUDA. Офіційне керівництво щодо установки [14] досить просте. Єдине, на що варто звернути увагу, це відповідність версії фреймворка вашій версії CUDA SDK і CUDNN. Чим ближче версія фреймворка до версії CUDA, тим краще.

Після підготовки мого середовища Python до оновлення додатку я виконав наступне: змінив оператори імпорту

```
import cv2

from facenet_pytorch import MTCNN

import torch
```

спробуйте вибрати відеокарту, сумісну з CUDA, якщо це можливо

```
#use cuda capable gpu if available otherwise use cpu

device = "cuda" if torch.cuda.is_available() else "cpu"
```

додайте рамку для обличчя та пов'яжіть з нею функцію, що забезпечує правильний напрямок обличчя до камери.

```
faceAligned = isFaceControlPassed(leftEye, rightEye,

    ((leftEye0[0],      leftEye1[0]),      (leftEye0[1],
    leftEye1[1])),

    ((rightEye0[0],      rightEye1[0]),      (rightEye0[1],
    rightEye1[1])))
```

Весь користувацький інтерфейс та виведення на екран може масштабуватися, тому незалежно від розміру вікна програми - все залишиться на своїх місцях.

```
#The left side of the frame
```

```
left0 = (int(round(width / FFWM[0])), int(round(height / FFHM[0])))
```

```
left1 = (int(round(width / FFWM[1])), int(round(height / FFHM[1])))
```

```
left2 = (int(round(width / FFWM[2])), int(round(height / FFHM[2])))
```

```
left3 = (int(round(width / FFWM[3])), int(round(height / FFHM[3])))
```

```
left4 = (int(round(width / FFWM[4])), int(round(height / FFHM[4])))
```

```
left5 = (int(round(width / FFWM[5])), int(round(height / FFHM[5])))
```

Також дуже легко налаштовувати параметри MTCNN через його конструктор, але це в основному робиться лише тоді, коли метою додатка є виявлення декількох облич одночасно. Таким чином, ви досягнете більшої кількості кадрів на секунду і більшої кількості результатів за рахунок збільшення кількості помилково позитивних та помилково негативних результатів. Такий підхід використовує розмиття та перехресні операції мережі між кадрами, тому кількість помилкових результатів може зростати. Це буде прикладом "жадібної" мережі.

```
detector = MTCNN (margin=14, factor=0.6, keep_all=True, device=device)
```

Щоб говорити про UX його можна побачити на скріншоті нижче (дивись малюнок 3.9). Це допомагає правильно вирівняти обличчя по відношенню до камери, щоб досягти найкращого результату. Нижній текст і кадр мають масштабованість.

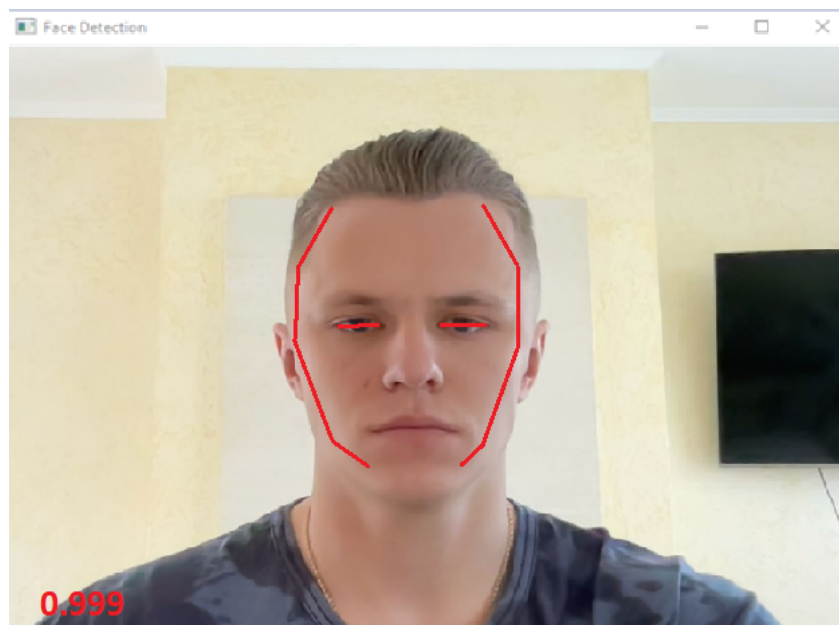


Рисунок 3.9 - Новий UX - лицьова рама

Джерело: розроблено автором

Якщо ви відсунете свої очі від відповідних рядків, ви побачите попередження з проханням вирівняти належним чином. Причина, чому вирівнювання проводиться тільки за допомогою розміщення очей, пов'язана з різними розмірами людських голів і різними формами їх облич. Коли хтось може піднятися ідеально, інші не можуть. І наші очі направляють голову, так що він буде слідувати погляду більшу частину часу.

На наступному знімку вікна показано, як новий MTCNN працює у темряві (перегляньте малюнок 3.10). Він все ще може виявити обличчя в кадрі, але через низьку впевненість користувача просять покращити освітлення або видалити аксесуари для обличчя.

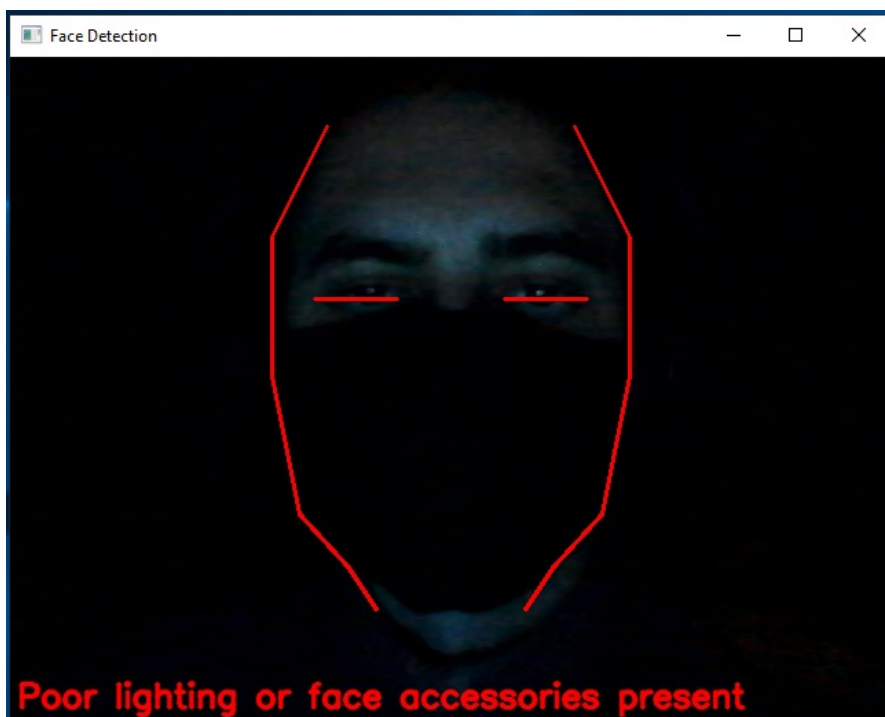


Рисунок 3.10 - Модернізовані MTCNN процеси за поганого освітлення

Джерело: розроблено автором

3.3 Результати експерименту та шляхи вдосконалення системи MTCNN для розпізнавання облич

Результати, які ми отримали, показують, що простий визначник обличчя cv2 здатний виявляти обличчя. Він не використовує багато апаратних ресурсів і має досить хороші результати - до 75%. Однак він нестійкий. Чим ближче я був до камери, тим нижче точність, яку показував визначник - зниження до 62%, що становить 10% втрати через кілька сантиметрів. Крім

того, після того, як я змішав свої фотографії з фотографіями інших людей, він більше не міг сказати, що це був я на кадрі - точність була нижча, ніж 60%, що неприпустимо. Він також не втримався перед простою медичною маскою, яка приховала мене. І в наш час глобальної епідемії необхідно носити маску, що знімає підозру з маскованих людей.

Щодо визначника обличчя MTCNN - він виявився набагато кращим. Тришарова нейронна мережа неперевершена. Вона має високу точність - до 99%, і вона могла виявити моє обличчя не лише з фронту, але й з боків. Стара атака не є загрозою більше, що може свідчити про високий рівень покращення алгоритму. Кілька способів, якими я міг обдурити систему, наступні: або маска і сонцезахисні окуляри, що може виглядати підозріло, особливо у хмарну погоду або в нічний час, або розмальована і/або надрукована маска, у моєму випадку з двома фотографіями, прикріпленими до неї.

Отримані мною результати показують, що одну з найпопулярніших та надійних мереж виявлення обличчя все ще можна обдурити і її потрібно покращувати. Вона не знаходиться в стані, в якому була десять років тому - прямолінійні підходи, такі як носіння масок у поєднанні з окулярами і шапкою, все ще можуть обдурити систему, хоча вона працює набагато краще зараз, коли стикається лише з одним аксесуаром одночасно. Навпаки, людина з повністю закритим обличчям, яка носить окуляри і шапку всередині будівлі, буде підозрілою, і це буде завданням служби безпеки провести розслідування.

Незважаючи, що комп'ютерні системи розвиваються і зробили значний крок вперед, особисто я вважаю, що ми не можемо повністю покладатися на такі автоматизовані системи. Вони можуть бути потужним доповненням до сучасних систем безпеки, але ніколи не замінять їх.

Щодо можливих шляхів вдосконалення системи розпізнавання обличчя, що базується на MTCNN, можливо розглянути декілька наведених

нижче варіантів для підвищення ефективності та швидкодії. Ось більш детальний огляд можливих підходів:

Використання апаратного прискорення: Враховуючи зростаючу потужність графічних процесорів (GPU) та тензорних процесорів (TPU), варто розглянути можливість оптимізації шляхом використання спеціалізованих апаратних прискорювачів.

Паралельні обчислення: Розробка або використання паралельних алгоритмів може значно підвищити продуктивність, розподіляючи завдання на різні ядра процесора або навіть використовуючи розподілені обчислювальні системи.

Оптимізація алгоритму: Подальше вдосконалення алгоритмів для зменшення обчислювального навантаження без втрати точності є ключовим пунктом. Це може включати удосконалення методів виявлення обличчя та оптимізацію параметрів моделі.

Квантизація моделі: Зменшення кількості бітів, що використовуються для представлення параметрів моделі, може покращити продуктивність, зменшуючи обсяг пам'яті та ресурсів обчислювальних операцій.

Кешування результатів: Зберігання проміжних результатів обробки для подальшого використання може запобігти повторній обробці одних і тих же даних, що зменшить час обчислень.

Використання попередньо навчених моделей: Використання попередньо навчених моделей з можливістю їх налаштування під конкретні завдання може значно спростити процес навчання та зменшити ресурси, необхідні для цього.

Оптимізація роботи з переднім планом/фоном: Якщо розпізнавання облич у відео, можна використовувати методи визначення областей

зображення, які найбільш істотні для розпізнавання, та обробляти лише їх, уникнувши зайвих обчислень.

Оптимізація гіперпараметрів: Налаштування параметрів моделі та процесу навчання для досягнення оптимальної продуктивності може бути ключем до успішної оптимізації.

Ці підходи можуть бути використані окремо або у поєднанні для досягнення найкращої продуктивності системи розпізнавання облич на основі MTCNN.

ВИСНОВКИ

У рамках даної магістерської дипломної роботи було проведено два експерименти з системами розпізнавання обличчя на основі різних моделей: одна базувалась на бібліотеці cv2, а інша використовувала модель MTCNN.

1. Описано архітектуру та методи, що використовувались для побудови комп'ютеризованої системи розпізнавання облич, що ґрунтується на MTCNN та cv2 моделях.
2. В результаті дослідження було виявлено, що обидві моделі мають свої переваги та недоліки. Модель, побудована на основі бібліотеки cv2, відрізняється простотою використання та швидкодією, але вона може бути менш точною в розпізнаванні обличчя у складних умовах, таких як погане освітлення або зміна кута огляду.
3. Модель MTCNN в рамках експерименту підтвердила свою високу точність розпізнавання обличчя навіть у важких умовах. Вона здатна ефективно впоратися з викликами, які ставлять перед нею такі умови, і забезпечити надійне розпізнавання облич.
4. Базуючись на результатах експерименту зроблено висновок, який підтверджують результати дослідження, що обидві моделі можуть бути використані у різних сферах застосування. Модель cv2 може бути більш відповідною для ситуацій, де важлива швидкість та простота використання, тоді як модель MTCNN відмінно підходить для завдань, де вимагається висока точність розпізнавання обличчя в умовах з великими викликами.

5. Запропоновано рекомендації щодо подальшого розвитку систем розпізнавання обличчя, які включають: покращення точності і швидкодії, розширення функціональності, розширення сфери застосування в інших областях, забезпечення конфіденційності і безпеки даних, інтеграція з іншими технологіями.

У роботі вирішено завдання комп'ютеризованого аналізу розпізнавання облич на основі моделей cv2 та MTCNN, що має важливе значення для досягнення більшої ефективності даних моделей та покращення алгоритмів для забезпечення надійності використання і мінімізації можливості обходу системи розпізнавання облич.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Real-world attack on MTCNN face detection system [Електронний ресурс] / E.Kaziakhmedov, K. Kireev, G. Melnikov, M. Pautov. – 2022. – Режим доступу до ресурсу: https://www.researchgate.net/publication/336550776_Real-world_attack_on_MTCNN_face_detection_system.
2. RetinaFace: Single-stage Dense Face Localisation in the Wild [Електронний ресурс] / [J. Deng, J. Guo, Y. Zhou та ін.] // arxiv.org. – 2022. – Режим доступу до ресурсу: <https://arxiv.org/abs/1905.00641>.
3. Intriguing properties of neural networks [Електронний ресурс] / [C. Szegedy., W. Zaremba, I. Sutskever та ін.] // arxiv.org. – 2023. – Режим доступу до ресурсу: <https://arxiv.org/abs/1312.6199>.
4. Goodfellow I. J. Explaining and Harnessing Adversarial Examples [Електронний ресурс] / I. J. Goodfellow, J. Shlens, C. Szegedy // arxiv.org. – 2021. – Режим доступу до ресурсу: <https://arxiv.org/abs/1412.6572>.
5. Kurakin A. Adversarial examples in the physical world [Електронний ресурс] / A. Kurakin, I. J. Goodfellow, S. Bengio // arxiv.org. – 2020. – Режим доступу до ресурсу: <https://arxiv.org/abs/1607.02533>.
6. Фрідман, Ю. І. Штучний інтелект: теорія і практика. - 2020. - Київ: Видавничий дім "КМ Академія".
7. Черниш, О. Розвиток штучного інтелекту в Україні: виклики та перспективи. - 2023. - Журнал "Інформаційні технології та комп'ютерна інженерія", 3(72), с. 42-47.
8. Михайленко, В. Використання методів розпізнавання облич у системах безпеки в Україні. - 2021. - Журнал "Безпека та оборона", 2(15), с. 28-33.
9. Бабіч, І. Правові аспекти використання систем розпізнавання облич в Україні. - 2022. - Журнал "Право та суспільство", 4(25), с. 82-89.

10. Денисенко, С. Аналіз ефективності алгоритмів розпізнавання облич в умовах українських реалій. - 2019. - Журнал "Інформаційні технології та комп'ютерна інженерія", 4(73), с. 18-25.
11. Ковальчук, О. Впровадження систем розпізнавання облич у сфері комерції в Україні: виклики та перспективи. - 2020. - Журнал "Економіка і бізнес", 1(9), с. 64-71.
12. Петренко, Н. Використання штучного інтелекту у сфері медицини в Україні. - 2021. - Журнал "Медична інформатика та інженерія", 2(40), с. 12-19.
13. Шевченко, В. Роль штучного інтелекту у вирішенні проблем безпеки в Україні. - 2022. - Журнал "Безпека та оборона", 4(17), с. 52-59.
14. Гаврилюк, І. Перспективи використання нейронних мереж для розпізнавання облич в українських умовах. - 2019. - Журнал "Наука та техніка", 3(88), с. 76-81.
15. Козаченко, Т. Використання методів машинного навчання для розпізнавання облич у соціальних мережах в Україні. - 2022. - Журнал "Соціальні комунікації", 2(51), с. 44-49.
16. Литвиненко, В. Сучасні технології розпізнавання облич у кримінальному розшуку в Україні. - 2021. - Журнал "Кримінально-виконавчий кодекс", 1(10), с. 102-109.
17. Іванов, О. Використання методів глибокого навчання для розпізнавання облич в українських умовах. - 2022. - Журнал "Інформаційні технології та комп'ютерна інженерія", 1(70), с. 36-42.
18. Сидоренко, Г. Проблеми та перспективи розвитку штучного інтелекту в Україні. - 2021. - Журнал "Наука та інновації", 3(24), с. 18-23.
19. Попов, Д. Використання систем розпізнавання облич у банківській сфері в Україні. - 2020. - Журнал "Фінансова сфера", 2(12), с. 82-89.
20. Новіков, М. Етичні аспекти використання систем розпізнавання облич в Україні. - 2018. - Журнал "Етика та мораль", 4(31), с. 56-63.

21. Завгородня, О. Автоматизація систем розпізнавання облич в українських умовах. - 2021. - Журнал "Автоматика та управління", 3(48), с. 72-79.
22. Гордієнко, Л. Перспективи використання методів машинного навчання для розпізнавання облич у сфері освіти в Україні. - 2020. - Журнал "Інновації в освіті", 1(7), с. 42-49.
23. Романов, П. Вплив розпізнавання облич на рівень приватності в українських умовах. - 2018. - Журнал "Приватність та захист персональних даних", 2(15), с. 98-105.
24. Калініченко, Г. Інтелектуальні системи розпізнавання облич у галузі транспортування в Україні. - 2021. - Журнал "Транспортні технології", 3(64), с. 28-35.
25. Макаренко, Л. Роль штучного інтелекту у розвитку індустрії розпізнавання облич в Україні. - 2022. - Журнал "Промисловість та інновації", 4(18), с. 76-83.
26. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. - 2021. - MIT Press.
27. Sun, Y., Wang, X., & Tang, X. Deep Learning Face Representation from Predicting 10,000 Classes. - 2022. - IEEE Conference on Computer Vision and Pattern Recognition.
28. Schroff, F., Kalenichenko, D., & Philbin, J. Facenet: A Unified Embedding for Face Recognition and Clustering. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
29. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. - 2019. - IEEE Conference on Computer Vision and Pattern Recognition.
30. Russel, S., & Norvig, P. Artificial Intelligence: A Modern Approach. - 2021. - Pearson Education.
31. Goodfellow, I., Bengio, Y., & Courville, A. Deep Learning. - 2022. - MIT Press.
32. Bishop, C. M. Pattern Recognition and Machine Learning. - 2023. - Springer.
33. Koller, D., & Friedman, N. Probabilistic Graphical Models: Principles and

- Techniques. - 2019. - MIT Press.
34. Murphy, K. P. Machine Learning: A Probabilistic Perspective. - 2020. - MIT Press.
 35. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. Rethinking the Inception Architecture for Computer Vision. - 2020. - arXiv preprint arXiv:1512.00567.
 36. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. ImageNet: A large-scale hierarchical image database. - 2019. - IEEE Conference on Computer Vision and Pattern Recognition.
 37. Schroff, F., Kalenichenko, D., & Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. - 2020. - IEEE Conference on Computer Vision and Pattern Recognition.
 38. Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. Deep Face Recognition. - 2019. - British Machine Vision Conference.
 39. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. - 2016. - IEEE Signal Processing Letters.
 40. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. SSD: Single Shot Multibox Detector. - 2020. - European Conference on Computer Vision.
 41. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
 42. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. - 2021. - International Journal of Computer Vision.
 43. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. Learning Deep Features for Scene Recognition Using Places Database. - 2021. - Advances in Neural Information Processing Systems.

44. LeCun, Y., Bengio, Y., & Hinton, G. Deep Learning. - 2021. - Nature.
45. Parkhi, O. M., Simonyan, K., Vedaldi, A., & Zisserman, A. A Compact and Discriminative Face Track Descriptor. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
46. Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
47. Sun, Y., Wang, X., & Tang, X. Deep Learning Face Representation from Predicting 10,000 Classes. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
48. Schroff, F., Kalenichenko, D., & Philbin, J. Facenet: A Unified Embedding for Face Recognition and Clustering. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
49. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. - 2021. - IEEE Conference on Computer Vision and Pattern Recognition.
50. Real-world attack on MTCNN face detection system [Электронный ресурс] / E. Kaziakhmedov, K. Kireev, G. Melnikov, M. Pautov // github.com. – 2022. – Режим доступа до ресурсу: <https://github.com/edosedgar/mtcnnattack>.
51. Esler T. Face Recognition Using Pytorch [Электронный ресурс] / Tim Esler // github.com. – 2020. – Режим доступа до ресурсу: <https://github.com/timesler/facenet-pytorch>.
52. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks [Электронный ресурс] / K. Zhang, Z. Zhang, Z. Li, Y. Qiao // arxiv.org – 2021. – Режим доступа до ресурсу: <https://arxiv.org/abs/1604.02878>.
53. M. Justin OpenCV-Python-Series [Электронный ресурс] / Justin Mitchel // github.com. – 2020. – Режим доступа до ресурсу: <https://github.com/codingforentrepreneurs/OpenCV-Python-Series>.
54. Viola P. Rapid Object Detection using a Boosted Cascade of Simple Features

[Электронный ресурс] / P. Viola, M. Jones // ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001. – 2019. – Режим доступа до ресурсу:
<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>.

55. McKnight et al., 2021 [Электронный ресурс] / – Режим доступа до ресурсу:
<https://ieeexplore.ieee.org/abstract/document/5718721/authors#authors>

56. Cha, 2020 [Электронный ресурс] / – Режим доступа до ресурсу:
<https://journals.uic.edu/ojs/index.php/fm/article/view/2889>

57. Tensorflow.org [Электронный ресурс] / – Режим доступа до ресурсу:
<https://www.tensorflow.org/install/pip>

58. Pytorch.org [Электронный ресурс] / – Режим доступа до ресурсу:
<https://pytorch.org/get-started/locally/>

SOFTWARE CODE LISTING

Code of “faces.py” – cv2 implementation of face recognizer.

```
import numpy as np
import cv2
import pickle

face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_alt2.xml')
eye_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_eye.xml')
smile_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_smile.xml')

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read("./recognizers/face-trainner.yml")

labels = {"person_name": 1}
with open("pickles/face-labels.pickle", 'rb') as f:
    og_labels = pickle.load(f)
    labels = {v:k for k,v in og_labels.items()}

cap = cv2.VideoCapture(0)

while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)
    for (x, y, w, h) in faces:
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]

        id_, conf = recognizer.predict(roi_gray)
        if conf>=60 and conf <= 95:
            font = cv2.FONT_HERSHEY_SIMPLEX
            name = labels[id_] + ' ' + str(conf)
            color = (255, 255, 255)
            stroke = 2
            cv2.putText(frame, name, (x,y), font, 1, color, stroke, cv2.LINE_AA)

            color = (255, 0, 0) #BGR 0-255
            stroke = 2
            end_cord_x = x + w
            end_cord_y = y + h
            cv2.rectangle(frame, (x, y), (end_cord_x, end_cord_y), color, stroke)
    cv2.imshow('cv2_face-recognition',frame)
    if cv2.waitKey(20) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Code of “faces-train.py” – a script to train a cv2 implementation of face recognizer.

```

import cv2
import os
import numpy as np
from PIL import Image
import pickle

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
image_dir = os.path.join(BASE_DIR, "images")

face_cascade = cv2.CascadeClassifier('cascades/data/haarcascade_frontalface_alt2.xml')
recognizer = cv2.face.LBPHFaceRecognizer_create()

current_id = 0
label_ids = {}
y_labels = []
x_train = []

for root, dirs, files in os.walk(image_dir):
    for file in files:
        if file.endswith("png") or file.endswith("jpg"):
            path = os.path.join(root, file)
            label = os.path.basename(root).replace(" ", "-").lower()
            print(label, path)
            if not label in label_ids:
                label_ids[label] = current_id
                current_id += 1
            id_ = label_ids[label]
            pil_image = Image.open(path).convert("L")
            size = (550, 550)
            final_image = pil_image.resize(size, Image.ANTIALIAS)
            image_array = np.array(final_image, "uint8")
            faces = face_cascade.detectMultiScale(image_array,
scaleFactor=1.5, minNeighbors=5)

            for (x,y,w,h) in faces:
                roi = image_array[y:y+h, x:x+w]
                x_train.append(roi)
                y_labels.append(id_)

with open("pickles/face-labels.pickle", 'wb') as f:
    pickle.dump(label_ids, f)

recognizer.train(x_train, np.array(y_labels))
recognizer.save("recognizers/face-trainer.yml")

```

Code of “Face_Recognition.ipynb” – a Jupiter notebook for python, which implements MTCNN face recognizer.

```

import cv2
from mtcnn import MTCNN

```

```

detector = MTCNN()
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    try:
        result = detector.detect_faces(frame)
        x1, y1, w, h = result[0]['box']

        cv2.rectangle(frame,
                       (x1, y1),
                       (x1 + w, y1 + h),
                       (0, 0, 255),
                       thickness=2)

        cv2.putText(frame, str(
            result[0]["confidence"]),
                    (x1, y1),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
                    2, cv2.LINE_AA)

        cv2.circle(frame, result[0]["keypoints"]["left_eye"], 2, (0, 0, 255),
-1)
        cv2.circle(frame, result[0]["keypoints"]["right_eye"], 2, (0, 0,
255), -1)
        cv2.circle(frame, result[0]["keypoints"]["nose"], 2, (0, 0, 255), -1)
        cv2.circle(frame, result[0]["keypoints"]["mouth_left"], 2, (0, 0,
255), -1)
        cv2.circle(frame, result[0]["keypoints"]["mouth_right"], 2, (0, 0,
255), -1)

    except:
        pass

    cv2.imshow('Face Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

```

Code of “Face_Recognition.py” – update of a Jupiter Notebook containing new MTCNN solution.

```
import cv2
from facenet_pytorch import MTCNN
import torch

#use cuda capable gpu if available otherwise use cpu
device = "cuda" if torch.cuda.is_available() else "cpu"

#force use of chosen device: cuda gpu or cpu
detector = MTCNN(device = device)

#default resolution is 640x480
#this variables are necessary to scale all the UI
cap = cv2.VideoCapture(0)
width = float(cap.get(3))
height = float(cap.get(4))

#We need lists of modifiers to keep face frame
#scalable for any output resolution

#Face Frame Height Modifiers
FFHM = [9.6, 3.7, 2.08, 1.45, 1.3, 1.2]
#Face Frame Width Modifiers
FFWM = [2.78, 3.37, 3.37, 3.05, 2.61, 2.41]

#Eyes Frame Height Modifiers
EFHM = [2.74, 2.74]
#Eyes Frame Width Modifiers
EFWM = [2.8, 2.2]

#We define (x, y) coordinates which will be joined
#and then create the face frame

#The left side of the frame
left0 = (int(round(width / FFWM[0])), int(round(height / FFHM[0])))
left1 = (int(round(width / FFWM[1])), int(round(height / FFHM[1])))
left2 = (int(round(width / FFWM[2])), int(round(height / FFHM[2])))
left3 = (int(round(width / FFWM[3])), int(round(height / FFHM[3])))
left4 = (int(round(width / FFWM[4])), int(round(height / FFHM[4])))
left5 = (int(round(width / FFWM[5])), int(round(height / FFHM[5])))

#The right side of the frame
right0 = (int(width - (round(width / FFWM[0]))), int(round(height / FFHM[0])))
```

```

right1 = (int(width - (round(width / FFWM[1]))), int(round(height /
FFHM[1])))
right2 = (int(width - (round(width / FFWM[2]))), int(round(height /
FFHM[2])))
right3 = (int(width - (round(width / FFWM[3]))), int(round(height /
FFHM[3])))
right4 = (int(width - (round(width / FFWM[4]))), int(round(height /
FFHM[4])))
right5 = (int(width - (round(width / FFWM[5]))), int(round(height /
FFHM[5])))

#The left eye of the frame
leftEye0 = (int(round(width / EFWM[0])), int(round(height / EFHM[0])))
leftEye1 = (int(round(width / EFWM[1])), int(round(height / EFHM[1])))

#The right eye of the frame
rightEye0 = (int(width - (round(width / EFWM[0]))), int(round(height /
EFHM[0])))
rightEye1 = (int(width - (round(width / EFWM[1]))), int(round(height /
EFHM[1])))

#draw the face frame
def drawFaceFrame():
    cv2.line(frame, left0, left1, (0, 0, 255), 2)
    cv2.line(frame, left1, left2, (0, 0, 255), 2)
    cv2.line(frame, left2, left3, (0, 0, 255), 2)
    cv2.line(frame, left3, left4, (0, 0, 255), 2)
    cv2.line(frame, left4, left5, (0, 0, 255), 2)

    cv2.line(frame, right0, right1, (0, 0, 255), 2)
    cv2.line(frame, right1, right2, (0, 0, 255), 2)
    cv2.line(frame, right2, right3, (0, 0, 255), 2)
    cv2.line(frame, right3, right4, (0, 0, 255), 2)
    cv2.line(frame, right4, right5, (0, 0, 255), 2)

    cv2.line(frame, leftEye0, leftEye1, (0, 0, 255), 2)
    cv2.line(frame, rightEye0, rightEye1, (0, 0, 255), 2)

#ensure that the face is correctly directed into the camera
def isFaceControlPassed(leftEyeActual, rightEyeActual,
                        leftEyeExpected, rightEyeExpected):
    #left eye x offset
    if not (leftEyeActual[0] > leftEyeExpected[0][0] - int(round(width /
128)) and leftEyeActual[0] < leftEyeExpected[0][1] + int(round(width /
42.6))):
        return False

```

```

    #left eye y offset
    if not (leftEyeActual[1] > leftEyeExpected[1][0] - int(round(height /
24)) and leftEyeActual[1] < leftEyeExpected[1][1] + int(round(height / 48))):
        return False
    #right eye x offset
    if not (rightEyeActual[0] < rightEyeExpected[0][0] + int(round(width /
128)) and rightEyeActual[0] > rightEyeExpected[0][1] - int(round(width /
42.6))):
        return False
    #right eye y offset
    if not (rightEyeActual[1] > rightEyeExpected[1][0] - int(round(height /
24)) and rightEyeActual[1] < rightEyeExpected[1][1] + int(round(height /
48))):
        return False
    return True

#main loop
while True:
    ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    drawFaceFrame()

    try:
        boxes, confidence, landmarks = detector.detect(frame, landmarks =
True)

        #collect eyes landmarks to ensure person's correct face direction
        leftEye = (int(landmarks[0][0][0]), int(landmarks[0][0][1]))
        rightEye = (int(landmarks[0][1][0]), int(landmarks[0][1][1]))

        faceAligned = isFaceControlPassed(leftEye, rightEye,
((leftEye0[0], leftEye1[0]),
(leftEye0[1], leftEye1[1])),
((rightEye0[0], rightEye1[0]),
(rightEye0[1], rightEye1[1])))

        #if the face is directed properly the program gives its prediction
        #taking into account stable high results, any prediction lower than
0.9
        #is considered as artifact which prevents from proper face detection
        if faceAligned:
            text = str(confidence[0])[5:]

```

```

        if confidence < 0.9:
            text = "Poor lighting or face accessories present"

        else:
            text = "Please, align your face according to the frame"

            #show text with warning or prediction
            cv2.putText(frame, text, (int(round(width / 128)), int(round(height /
1.02))),
                        cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 0, 255),
                        2, cv2.LINE_AA)

    except:
        pass

    cv2.imshow("Face Detection", frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()

```

haarcascade eye tree eyeglasses

```

<?xml version="1.0"?>
<opencv_storage>
<cascade type_id="opencv-cascade-classifier"><stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>20</height>
  <width>20</width>
  <stageParams>
    <maxWeakCount>47</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount></featureParams>
  <stageNum>30</stageNum>
  <stages>
    <_>
      <maxWeakCount>5</maxWeakCount>
      <stageThreshold>-1.6473180055618286e+00</stageThreshold>
      <weakClassifiers>
        <_>
          <internalNodes>
            2 1 0 -2.6987109333276749e-02 0 -1 1 5.0670530647039413e-02
            -2 -3 2 -1.2915390729904175e-01</internalNodes>
          <leafValues>
            -8.0395472049713135e-01 6.0491400957107544e-01
            9.0544581413269043e-01 4.4070810079574585e-02</leafValues></_>
        <_>
          <internalNodes>
            2 1 3 8.8827736675739288e-02 0 -1 4 -2.0398240536451340e-02

```

```

-2 -3 5 -6.1261758208274841e-02</internalNodes>
<leafValues>
7.9218882322311401e-01 4.0692299604415894e-02
4.2585361003875732e-01 -7.0325207710266113e-01</leafValues></_>
<_>
<internalNodes>
2 1 6 -2.0490810275077820e-01 0 -1 7 9.4933047890663147e-02
-2 -3 8 1.2091030366718769e-03</internalNodes>
<leafValues>
-4.4017648696899414e-01 5.3640520572662354e-01
6.8776458501815796e-01 -5.5879348516464233e-01</leafValues></_>
<_>
<internalNodes>
1 0 9 9.2227972345426679e-04 -1 2 10 -7.2678289143368602e-04
-2 -3 11 6.8421510513871908e-04</internalNodes>
<leafValues>
-7.2684401273727417e-01 -5.8028000593185425e-01
5.6177532672882080e-01 -2.9834181070327759e-01</leafValues></_>
<_>
<internalNodes>
0 1 12 -5.1150590181350708e-02 2 -1 13
6.1622060835361481e-02 -2 -3 14 7.2873473167419434e-02</internalNodes>
<leafValues>
5.9840762615203857e-01 7.4743932485580444e-01
-4.9703779816627502e-01 2.8129258751869202e-01</leafValues></_></weakClassifiers></_>
<_>
<maxWeakCount>7</maxWeakCount>
<stageThreshold>-1.4257860183715820e+00</stageThreshold>
<weakClassifiers>
<_>
<internalNodes>
2 1 15 -4.1994878649711609e-01 0 -1 16
-5.6186288595199585e-02 -2 -3 17 -2.3711109533905983e-02</internalNodes>
<leafValues>
2.7586200833320618e-01 -6.4623218774795532e-01
8.5241252183914185e-01 8.3703370764851570e-03</leafValues></_>
<_>
<internalNodes>
1 0 18 4.0523439645767212e-02 -1 2 19 2.7388900518417358e-01
-2 -3 20 -1.4293800108134747e-02</internalNodes>
<leafValues>
7.4270218610763550e-01 -4.9286690354347229e-01
7.1784788370132446e-01 -4.2223978787660599e-02</leafValues></_>
<_>
<internalNodes>
0 1 21 -2.1144729107618332e-03 2 -1 22
1.0659949621185660e-03 -2 -3 23 1.0812469990924001e-03</internalNodes>
<leafValues>
-8.0196601152420044e-01 -6.6025912761688232e-01
4.7916370630264282e-01 -5.1645290851593018e-01</leafValues></_>
<_>
<internalNodes>
1 0 24 3.0198289081454277e-02 2 -1 25 4.0569551289081573e-02
-2 -3 26 7.0679739117622375e-02</internalNodes>
<leafValues>
5.1327562332153320e-01 6.6641497611999512e-01

```

```

<rects>
  <_>
    4 1 12 8 -1.</_>
  <_>
    10 1 6 8 2.</_></rects></_>
<_>
  <rects>
    <_>
      18 10 2 3 -1.</_>
    <_>
      17 11 2 1 3.</_></rects>
  <tilted>1</tilted></_>
<_>
  <rects>
    <_>
      12 1 6 3 -1.</_>
    <_>
      14 1 2 3 3.</_></rects></_>
<_>
  <rects>
    <_>
      1 16 1 3 -1.</_>
    <_>
      1 17 1 1 3.</_></rects></_>
<_>
  <rects>
    <_>
      10 9 1 2 -1.</_>
    <_>
      10 10 1 1 2.</_></rects></_>
<_>
  <rects>
    <_>
      19 13 1 4 -1.</_>
    <_>
      19 13 1 2 2.</_></rects>
  <tilted>1</tilted></_>
<_>
  <rects>
    <_>
      9 6 3 6 -1.</_>
    <_>
      9 9 3 3 2.</_></rects></_>
<_>
  <rects>
    <_>
      2 9 18 10 -1.</_>
    <_>
      2 9 9 5 2.</_>
    <_>
      11 14 9 5 2.</_></rects></_>
<_>
  <rects>
    <_>
      11 4 5 6 -1.</_>
    <_>

```

```

11 4 5 3 2.</_></rects>
<tilted>1</tilted></_>
<_>
<rects>
<_>
17 0 2 4 -1.</_>
<_>
17 1 2 2 2.</_></rects></_>
<_>
<rects>
<_>
2 3 3 4 -1.</_>
<_>
3 3 1 4 3.</_></rects></_>
<_>
<rects>
<_>
19 0 1 10 -1.</_>
<_>
19 5 1 5 2.</_></rects></_>
<_>
<rects>
<_>
1 7 6 6 -1.</_>
<_>
1 7 3 3 2.</_>
<_>
4 10 3 3 2.</_></rects></_>
<_>
<rects>
<_>
15 2 3 12 -1.</_>
<_>
11 6 3 4 3.</_></rects>
<tilted>1</tilted></_>
<_>
<rects>
<_>
3 9 7 6 -1.</_>
<_>
3 11 7 2 3.</_></rects></_>
<_>
<rects>
<_>
8 8 1 3 -1.</_>
<_>
8 9 1 1 3.</_></rects></_>
<_>
<rects>
<_>
4 13 6 6 -1.</_>
<_>
4 15 6 2 3.</_></rects></_>
<_>
<rects>
<_>

```

```

    1 13 4 3 -1.</_>
  <_>
    1 14 4 1 3.</_></rects></_>
<_>
  <rects>
    <_>
      7 1 4 4 -1.</_>
    <_>
      7 1 2 2 2.</_>
    <_>
      9 3 2 2 2.</_></rects></_>
<_>
  <rects>
    <_>
      2 4 2 2 -1.</_>
    <_>
      2 4 1 1 2.</_>
    <_>
      3 5 1 1 2.</_></rects></_>
<_>
  <rects>
    <_>
      2 4 16 3 -1.</_>
    <_>
      2 5 16 1 3.</_></rects></_>
<_>
  <rects>
    <_>
      0 6 17 3 -1.</_>
    <_>
      0 7 17 1 3.</_></rects></_>
<_>
  <rects>
    <_>
      5 6 10 3 -1.</_>
    <_>
      5 7 10 1 3.</_></rects></_></features></cascade>
</opencv_storage>

```

Ім'я користувача:
Комп'ютерної математики та інформаційної безпеки...

ID перевірки:
1016283937

Дата перевірки:
26.05.2024 14:55:49 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.05.2024 15:03:10 EEST

ID користувача:
100005746

Назва документа: Диплом_Романюк_Володимир

Кількість сторінок: 58 Кількість слів: 9365 Кількість символів: 68778 Розмір файлу: 4.38 MB ID файлу: 1016077708

3.24% Схожість

Найбільша схожість: 1.47% з Інтернет-джерелом (<https://python919.wordpress.com>)

2.55% Джерела з Інтернету 53 Сторінка 60

1.83% Джерела з Бібліотеки 123 Сторінка 60

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел