

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет «Комп'ютерні технології і системи»
(назва факультету)

Кафедра «Електронні обчислювальні машини»
(повна назва кафедри)

Пояснювальна записка
до кваліфікаційної роботи
магістра
(ступінь вищої освіти)

на тему: Дослідження можливостей використання біометрії обличчя в системі обліку робочого часу.

за освітньою програмою Комп'ютерна інженерія
зі спеціальності: 123 Комп'ютерна інженерія
(шифр і назва спеціальності)

Виконав: студент групи: КС2121

OK

(підпис студента)

Олександр КАБИЦЬКИЙ

(Ім'я ПРІЗВИЩЕ)

Керівник:

[Підпис]

(підпис)

доцент, Денис ОСТАПЕЦЬ

(посада, Ім'я ПРІЗВИЩЕ)

Нормоконтролер:

[Підпис]

(підпис)

доцент, Володимир ШАПОВАЛОВ

(посада, Ім'я ПРІЗВИЩЕ)

Засвідчую, що у цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

OK

(підпис)

Дніпро – 2022 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Faculty «Computer technologies and systems»

(faculty)

Department «Electronic computers»

(department)

Explanatory Note
to Master's Thesis

Master's

(higher education degree)

on the topic: Research of the possibilities of using facial biometrics in the worktime tracking system

according to educational curriculum Computer Engineering

in the Speciality: 123 Computer Engineering

(speciality and its code)

Done by the student of the group: KS2121

Oleksandr Kabytskyi

(name, surname)

Scientific Supervisor:

Associate Professor, Denys Ostapets

(position, name, surname)

Normative controller :

Associate Professor, Volodymyr Shapovalov

(position, name, surname)

Dnipro – 2022

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи

Кафедра: ЕОМ

Рівень вищої освіти: другий (магістерський)

Освітня програма: Комп'ютерна інженерія

Спеціальність: 123 Комп'ютерна інженерія

(шифр та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис)

(Ім'я ПРІЗВИЩЕ)

Дата

11.07.2022

ЗАВДАННЯ

на кваліфікаційну роботу

магістра

(ступінь вищої освіти)

студенту Кабицькому Олександру Михайловичу

(Прізвище, Ім'я По батькові)

1. Тема роботи: Дослідження можливостей використання біометрії обличчя в системі обліку робочого часу

Керівник роботи: Остапець Денис Олександрович, к.т.н., доцент

(Прізвище, Ім'я, По батькові, науковий ступінь, вчене звання)

затверджені наказом від

"29" 03 2022 р.

№ 284ст

2. Строк подання студентом роботи:

19.12.2022 р.

3. Вихідні дані до роботи:

- Методи та алгоритми біометричної ідентифікації за обличчям

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати):

4.1 Аналітична частина:

- Аналіз методів біометричної ідентифікації за обличчям

4.2 Основна частина:

- Огляд методів та засобів ідентифікації та аутентифікації в системах обліку робочого часу;

- Структура та функції системи обліку робочого часу з використанням біометрії обличчя;

- Розробка програмного забезпечення ідентифікації особи;

- Дослідження ефективності застосування біометрії обличчя за методом POINT DISTRIBUTION MODEL;

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

- Характеристики біометричних методик;
- Характеристики алгоритмів розпізнавання обличчя;
- Структура та функції системи обліку робочого часу;
- Основні алгоритми програм;
- Результати дослідження ефективності застосування біометрії обличчя;
- Приклади роботи програм.

6. Консультанти розділів роботи:

| Розділ | Прізвище, ініціали та посада консультанта | Завдання видав: (підпис консультанта, дата) | Завдання прийняв: (підпис студента, дата) |
|--------|---|---|---|
| | | | |
| | | | |

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|---|-------------------------------|----------|
| 1 | Огляд методів та засобів ідентифікації та аутентифікації в системах обліку робочого часу | 21.11.2022 | 20% |
| 2 | Структура та функції системи обліку робочого часу з використанням біометрії обличчя | 28.11.2022 | 20% |
| 3 | Розробка програмного забезпечення ідентифікації особи | 08.12.2022 | 30% |
| 4 | Дослідження ефективності застосування біометрії обличчя за методом POINT DISTRIBUTION MODEL | 16.12.2022 | 25% |
| 5 | Реферат, вступ, висновки | 19.12.2022 | 5% |
| 6 | Подання кваліфікаційної роботи до кафедри | 19.12.2022 | |
| 7 | Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії | 27.12.2022 | |

Студент

OKI

(підпис)

Олександр КАБИЦЬКИЙ

(Ім'я ПРІЗВИЩЕ)

Керівник роботи

OS

(підпис)

Денис ОСТАПЕЦЬ

(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кабицький О. М. Дослідження можливостей використання біометрії обличчя в системі обліку робочого часу. – Український державний університет науки і технологій, кафедра електронних обчислювальних машин. – Дипломна робота. – 88 с., 41 рис., 10 табл., 30 джерела, 9 додатків.

Об'єктом дослідження є методи та засоби біометричної ідентифікації осіб за обличчям в системі обліку робочого часу.

Метою роботи є формування рекомендацій щодо можливостей використання біометрії обличчя в системі обліку робочого часу.

Методи дослідження – експериментальне дослідження з використанням спеціально створеного програмного забезпечення.

Здійснено огляд та аналіз методів та засобів розпізнавання облич, дано характеристику методам ідентифікації облич та системам обліку робочого часу. Обрано метод для реалізації підсистеми ідентифікації особи. Описано структуру та функції системи. Наведено алгоритми роботи, написано та налагоджено програмне забезпечення. Розроблено інструкцію з використання. Проведено аналіз ефективності підсистеми ідентифікації в умовах погіршення зовнішніх та внутрішніх чинників. Сформовані рекомендації з використання біометричної ідентифікації в системі. Розроблені програмні засоби можуть використовуватися для реалізації в системах обліку робочого часу.

БИОМЕТРИЯ ОБЛИЧЧЯ, РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, ІДЕНТИФІКАЦІЯ ОСОБИ, ОБЛІК РОБОЧОГО ЧАСУ, ОПОРНІ ТОЧКИ, ДЕСКРИПТОРИ ЗОБРАЖЕНЬ, ЕВКЛІДОВА ВІДСТАНЬ, JAVASCRIPT, VISUAL STUDIO CODE;

ЗМІСТ

| | |
|---|----|
| ВСТУП | 8 |
| 1 ОГЛЯД МЕТОДІВ ТА ЗАСОБІВ ІДЕНТИФІКАЦІЇ ТА АУТЕНТИФІКАЦІЇ В СИСТЕМАХ ОБЛІКУ РОБОЧОГО ЧАСУ | 10 |
| 1.1 Загальні відомості..... | 10 |
| 1.2 Основні відомості про системи обліку робочого часу | 11 |
| 1.3 Огляд відомих систем обліку робочого часу | 12 |
| 1.4 Ідентифікація та аутентифікація в системах обліку робочого часу.... | 15 |
| 1.5 Відомі методи аналізу біометрії обличчя | 16 |
| 1.6 Порівняльний аналіз методів розпізнавання обличчя | 17 |
| 1.7 Висновки за розділом | 21 |
| 2 СТРУКТУРА ТА ФУНКЦІЇ СИСТЕМИ ОБЛІКУ РОБОЧОГО ЧАСУ З ВИКОРИСТАННЯМ БІОМЕТРІЇ ОБЛИЧЧЯ | 22 |
| 2.1 Структура та склад системи..... | 22 |
| 2.2 Методика розпізнавання обличчя..... | 28 |
| 2.3 Висновки за розділом | 37 |
| 3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІДЕНТИФІКАЦІЇ ОСОБИ | 38 |
| 3.1 Вибір середовища та засобів розробки | 38 |
| 3.2 Основні алгоритми програми | 39 |
| 3.3 Основні моделі, що використовуються в програмі | 41 |
| 3.4 Основні методи і параметри програми | 45 |
| 3.5 Інструкція з використання програми | 46 |
| 3.6 Висновки за розділом | 48 |
| 4. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ БІОМЕТРІЇ ОБЛИЧЧЯ ЗА МЕТОДОМ POINT DISTRIBUTION MODEL | 50 |
| 4.1 Чинники, вплив яких досліджується..... | 50 |
| 4.2 Дослідження впливу освітленості приміщення та відстані до камери | 55 |
| 4.3 Дослідження впливу освітленості приміщення та розміру зображення, що оброблюється | 58 |

| | | |
|------------|--|-----------|
| 4.4 | Дослідження впливу розміру зображення, що оброблюється та відстані до камери..... | 61 |
| 4.5 | Дослідження впливу обмеження кількості доступних для бази фото облич певної особи та зміни куту нахилу голови | 64 |
| 4.6 | Дослідження впливу наявності та відсутності опорних точок та частин обличчя | 68 |
| 4.7 | Аналіз результатів | 73 |
| 4.8 | Рекомендації, основані на отриманих результатах | 74 |
| 4.9 | Висновки за розділом | 75 |
| | ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ | 76 |
| | ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 77 |

ВСТУП

Для підвищення продуктивності праці в світі збільшується кількість компаній які починають інтеграцію систем обліку робочого часу. Суть цих систем полягає в аналізі витраченого часу на основі даних, отриманих з підсистеми ідентифікації особи. Деякі засоби ідентифікації можуть надати інформацію про місцезнаходження працівника, але не те, чим він займається, як витрачає робочий час. За умови інтеграції в систему обліку робочого часу відеокамер та аналізу біометрії обличчя, можна отримати набагато ефективніший аналіз продуктивності працівників.

З кожним днем попит на технології, пов'язанні з розпізнаванням особи за її обличчям зростає, як зростають і вимоги до точності розпізнавання.

Наразі розробляється багато інструментів та комплексів, що намагаються вирішити цю задачу. Тому тема роботи є актуальною.

Мета роботи – формування рекомендацій щодо можливостей використання біометрії обличчя в системі обліку робочого часу. Розроблене програмне забезпечення та рекомендації можуть бути використані при створенні системи обліку робочого часу.

Основні положення роботи доповідались та були схвалені на XVI Міжнародній науково-практичній конференції «Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості та освіті» у 2022 році (див. додаток А), опубліковано тези доповіді [1].

Представлена робота складається зі вступу, 4 розділів, висновків та додатків.

У розділі 1 представлений огляд та аналіз методів та засобів ідентифікації в системах обліку робочого часу. Розглянуто загальну структуру систем обліку робочого часу та їх складових. Наведено особливості методів розпізнавання обличчя та їх порівняння.

У розділі 2 наведено структуру системи обліку, параметри приміщення та описано обраний метод ідентифікації осіб. Здійснено опис розрахунку схожості двох облич.

У розділі 3 здійснено вибір середовища та мови програмування, наведено алгоритми роботи, на основі яких розроблено програмне забезпечення.

У розділі 4 наведено результати дослідження впливу окремих чинників на ефективність роботи підсистеми ідентифікації особи, таблиці.

В додатку А наведено копію тез доповіді на науковій конференції за змістом роботи, в додатках Б, В наведено вихідний код програми.

1 ОГЛЯД МЕТОДІВ ТА ЗАСОБІВ ІДЕНТИФІКАЦІЇ ТА АУТЕНТИФІКАЦІЇ В СИСТЕМАХ ОБЛІКУ РОБОЧОГО ЧАСУ

1.1 Загальні відомості

Системи обліку робочого часу - це програмне забезпечення, направлене на планування, аналіз і контроль активності під час робочого процесу. В основі таких систем знаходиться поняття резервів використання робочого часу (див. рис. 1.1) – це різноманітні можливості більш якісно використати робочий час використовуючи потенцій, непомічені можливості які можуть бути доступні підприємству або компанії [2].



Рисунок 1.1 - Резерви росту ефективності використання робочого часу

Поняття ідентифікатора та ідентифікації є основним поняттям для систем обліку робочого часу. Термін ідентифікація означає - «впізнання», «пошук за ознакою». Ідентифікація може здійснюватися за такими основними принципами [3]:

- ідентифікація за кодом, що запам'ятовується, - за кодом, що вводиться вручну за допомогою клавіатури, кодових перемикачів або інших подібних пристроїв;
- ідентифікація за речовим кодом - за кодом, записаним на фізичному носії (ідентифікаторі), як який застосовують різні ключі, картки, брелки, тощо;
- біометрична ідентифікація - ідентифікація, заснована на визначенні індивідуальних фізичних ознак людини.

Завдяки ідентифікації працівника впродовж дня можна отримати інформацію, як він використав свій робочий час впродовж дня. На основі цих даних, ввести зміни, які мають покращити роботу компанії.

1.2 Основні відомості про системи обліку робочого часу

Здебільшого системи обліку робочого часу займаються аналізом витраченого часу на основі програм і даних. Для отримання цієї інформації, в залежності від системи, використовується такі інструменти [3,4]:

- автоматичні хвіртки;
- датчики емоційного стану;
- інформація про структуру робочого об'єкту;
- зчитувачі біометричних даних;
- контролери;
- програмне забезпечення на комп'ютері для запису дій працівника;
- програмне забезпечення на комп'ютері для аналізу використання інших програм;
- бази даних;
- аналітичні системи;
- зчитувачі виконання механічних дій.

Сучасні системи обліку робочого часу можуть обмежуватись тільки ПК на приклад в ІТ компаніях, програма зчитує як працівник використовує свій час, збирає інформацію про те якими додатками, сайтами, програмним забезпеченням він користується і навіть може зробити аналіз продуктивності витраченого часу. В скороченому вигляді структура роботи програм систем обліку робочого часу зображена на рис. 1.2:



Рисунок 1.2 – Функціонування програм обліку робочого часу

Таким чином аналіз і майбутнє покращення ефективності працівника, тим чи іншим чином, є самоціллю систем обліку робочого часу.

1.3 Огляд відомих систем обліку робочого часу

Як вже було сказано системи обліку робочого часу необхідні для покращення ефективності використання робочого часу. Такі системи можуть працювати самостійно, на приклад, Yaware.TimeTracker, чи в складі інших систем, на приклад, Систем Контролю і Управління Доступом (далі СКУД). Вони мають допомагати в підвищенні продуктивності, безпеки, контролю в системі де вони задіяні, рис 1.3.

Yaware.TimeTracker – це онлайн-сервіс для урахування робочого часу співробітників за комп'ютером [6]. Цей додаток надає інформацію про все що відбувалось за комп'ютером включно з роботою в офлайн - облік часу нарад, зустрічей, перерв, також робить скріншоти та фіксує стан працівника через камеру, а велика база додатків і сайтів дає змогу проаналізувати продуктивність працівників [6,7]:

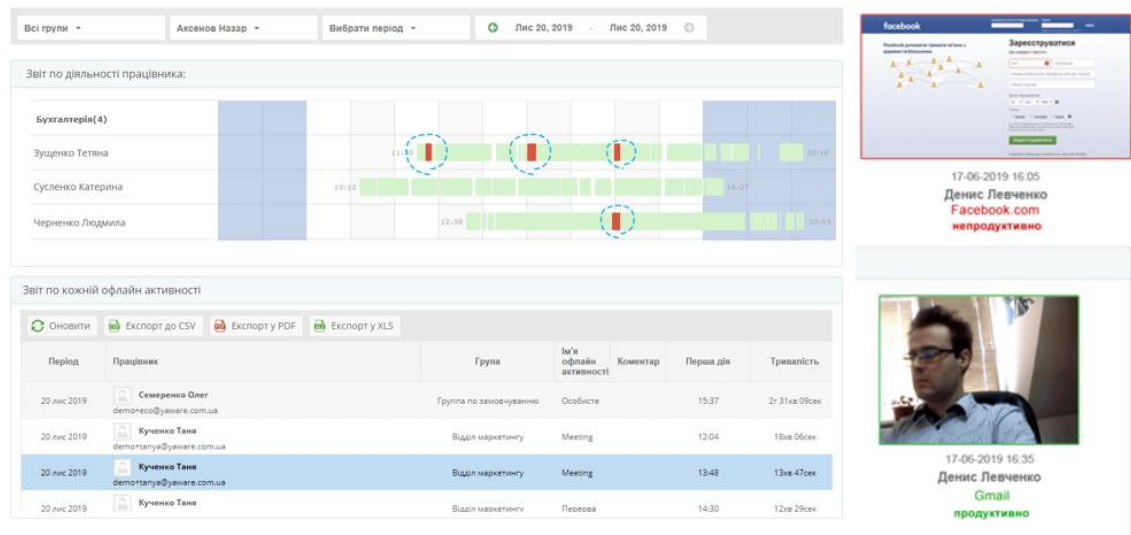


Рисунок 1.3 - Скріншоти роботи додатку Yaware.TimeTracker

Yaware.TimeTracker чудовий варіант для офісу ІТ, але він не може вийти за рамки комп'ютера і не фіксує продуктивність поза комп'ютером.

У свою чергу СКУД щільно пов'язаний з системами обліку робочого часу і фактично не може працювати без використання цих систем. Адже облік робочого часу дає змогу запобігати низці небезпек, насамперед дає змогу виконувати такі функції:

- санкціонування - процедуру присвоєння кожному користувачеві персонального ідентифікатора, коду, реєстрацію його в системі (або реєстрацію його біометричних ознак) та задавання для нього часових інтервалів і рівня доступу (в які приміщення, коли і хто має право заходити);
- ідентифікацію - процедуру впізнавання користувача за пред'явленим ідентифікатором або біометричною ознакою;

- авторизацію - що полягає в перевірці відповідності часу і рівня доступу встановленим у процесі санкціонування;
- автентифікацію - встановлення автентичності користувача за ознаками ідентифікації;
- дозвіл доступу або відмову в доступі - виконується на підставі результатів аналізу попередніх процедур;
- реєстрація - протоколювання всіх дій у системі;
- реагування - реакція системи на несанкціоновані дії (подання попереджувальних і тривожних сигналів, відмова в доступі тощо).

Процедура санкціонування проводиться оператором або адміністратором системи. Вона полягає у введенні необхідних даних у комп'ютер системи або в пам'ять контролера. Усі інші процедури можуть виконуватися системою автоматично. Очевидно, що процедуру автентифікації може бути виконано повноцінно тільки за допомогою біометричних систем [3]. Схема роботи функцій СКУД зображена на рис. 1.4.

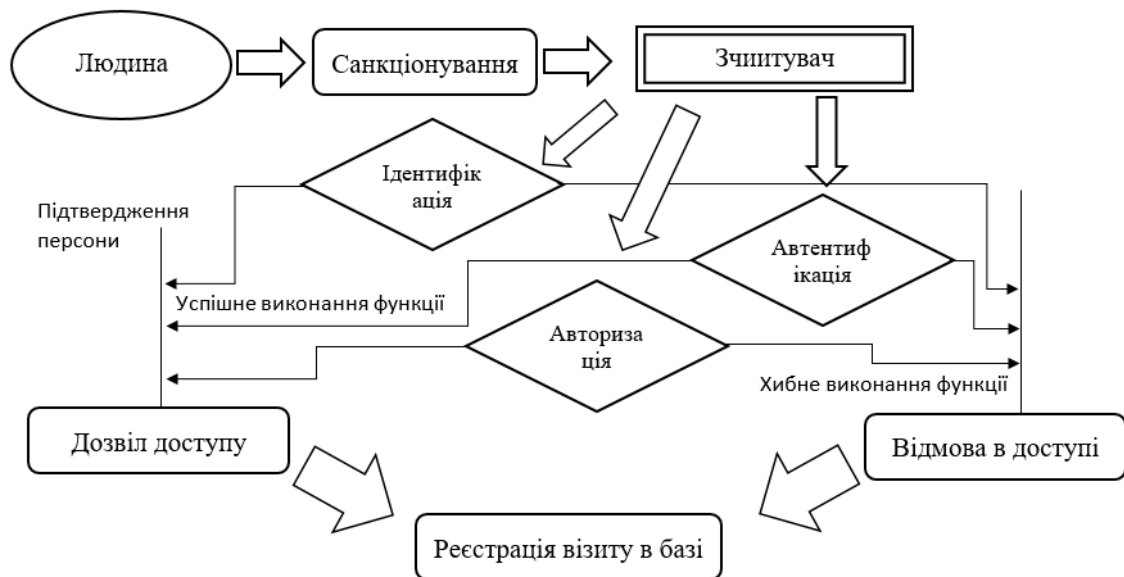


Рисунок 1.4 - Схема роботи та функції СКУД

Після занесення інформації в базу стає відомо - де і який час знаходився працівник під час робочого процесу, скільки в загальному працівник провів в тій

чи іншій аудиторії\кімнаті\будівлі робочого об'єкту з чого будується таблиця витраченого часу рис. 1.5.

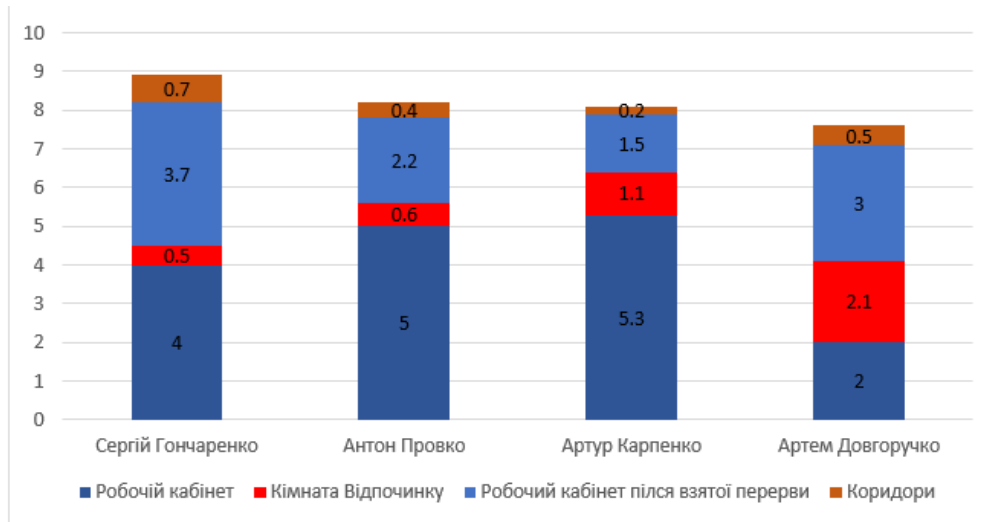


Рисунок 1.5 - Витрачений час працівниками за день (приклад)

Таким чином системи обліку робочого часу можна поділити на внутрішні, в рамках комп'ютера та зовнішні, всього робочого об'єкту.

1.4 Ідентифікація та аутентифікація в системах обліку робочого часу

Зовнішні системи обліку робочого часу можуть використовувати усі види ідентифікації та аутентифікації рис. 1.6:



Рисунок 1.6 - Структура системи обліку робочого часу

В той час як внутрішні системи обліку дуже обмежені в доступних засобах. Серед найбільш поширених використовується пароль, розпізнавання обличчя, розпізнавання відбитку пальця.

Загалом методи ідентифікації можна розділити на надійні, не надійні та потенційно надійні – ті що досі розвиваються.

До не надійних можна віднести ключ-карту, сканування голосу, та пароль, бо їх можна підробити, вкрасти та мімікрувати під іншу особу.

До надійних можна віднести відбиток пальця, через те, що ця технологія вже досягла піку.

До потенційно надійних відноситься сканування обличчя, зіниці ока, температурний аналіз тіла. Усі ці методи можуть бути чудовою заміною чи аналогом відбитку пальця. Через те, що відбиток можна підробити за допомогою сучасних інструментів, на відміну від сканування обличчя. Сканування може стати на заміну відбитку через те, що це спосіб ідентифікації, яким здебільшого користуються люди. Він не потребує фізичної взаємодії зі зчитувачем. Достатньо пройти по коридору де стоїть камера чи сісти за робоче місце.

1.5 Відомі методи аналізу біометрії обличчя

Розпізнавання облич зараз є однією з найпопулярніших задач комп'ютерного зору на зображеннях. Основними проблемами при виявленні облич є оклюзія, освітлення та складний фон. Для боротьби з цими проблемами було запропоновано широкий спектр алгоритмів які зараз можна розділити на дві окремі гілки – функціонально-орієнтовані підходи та зображення-орієнтовані підходи. У той час як функціонально-орієнтовані підходи знаходять ознаки (краї зображення, кути та інші структури, добре локалізовані у двох вимірах), зображення-орієнтовані підходи значною мірою залежать від сканування зображення, яке базується на вікнах або підкадрах [8].

Обидва підходи діляться на методи, а ті в свою чергу на реалізації. З функціонально-орієнтованих підходів необхідно розглянути метод моделі активної форми, а з зображення-орієнтованих підходів: нейронні мережі та лінійні підпросторові методи, що можна побачити на рис. 1.7.

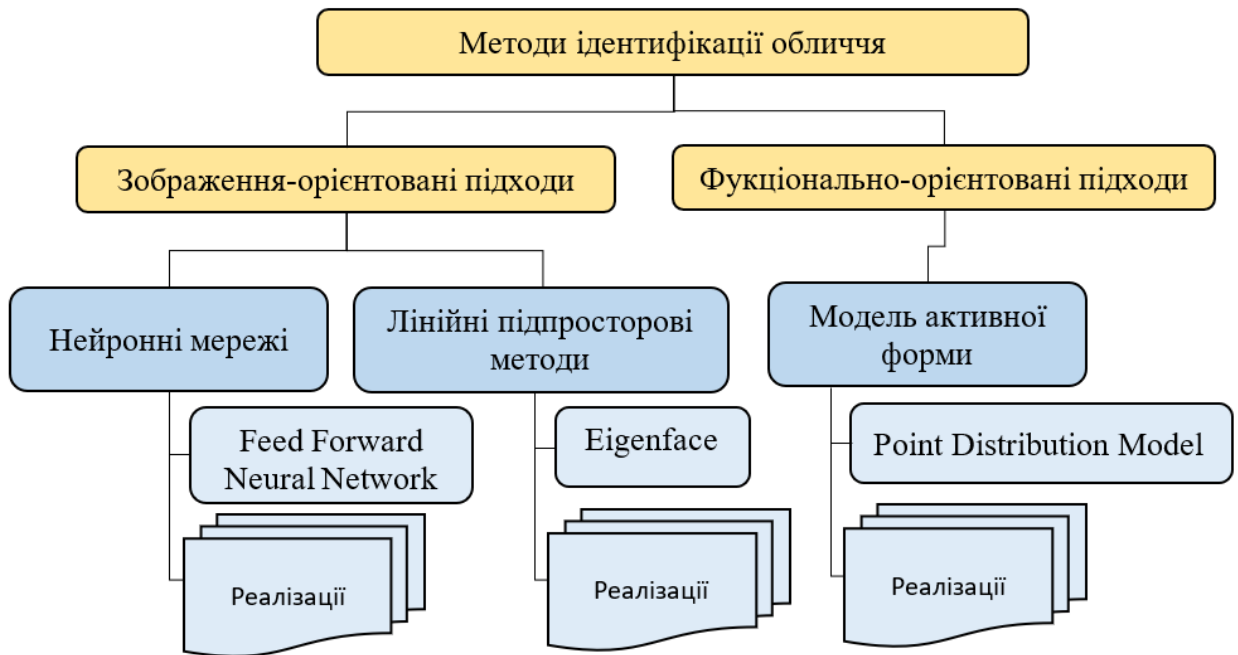


Рисунок 1.7 - Методи ідентифікації обличчя

Моделі активних форм мають справу зі складними та нежорсткими формами шляхом деформації відповідно до заданого прикладу за допомогою ітеративної обробки. Коли система переглядає зображення, вона пов'язує його з рисами обличчя, такими як ніс, рот і т.д., як тільки знаходить близьке сусідство з будь-якою з цих рис, координати цих частин беруться як карта, і звідси генерується маска. Маску можна змінювати вручну. Навіть якщо система визначилася з формою, вона може бути скоригована користувачем. Тренуючись на більшій кількості зображень, можна досягти кращої карти [9].

Сучасні методики виявлення облич, а саме зображення-орієнтовані підходи, базується на віконному скануванні зображення. Вікно сканується піксель за пікселем, щоб класифікувати обличчя і не-обличчя. Як правило, кожен метод в підходах на основі зображень варіюється з точки зору вікна сканування, розміру кроку, кількості ітерацій для отримання більш ефективного підходу.

1.6 Порівняльний аналіз методів розпізнавання обличчя

Point Distribution Model (PDM) - це техніка опису форми, яка спирається на опорні точки. Опорна точка - це анотація будь-якого зображення на будь-яку задану форму на зображеннях навчальної вибірки. Форма обличчя в PDM

формується шляхом нанесення опорних точок на форму обличчя на навчальному наборі зображень [10]. Модель, як правило, будується з глобальною формою обличчя, що має утворення очей, вух, носа та інших елементів обличчя, як показано на рис. 1.8.

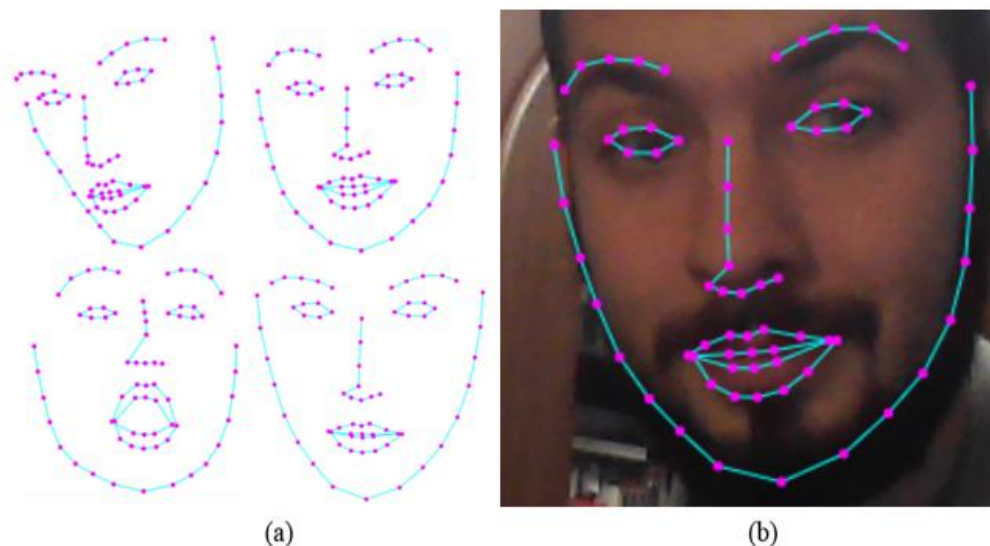


Рисунок 1.8. - Глобальні форми обличчя

На рисунку (a) типові навчальні глобальні форми обличчя, що складаються з рис обличчя, таких як очі, рот, ніс, брови та вуха [10]; (b) точки моделі, спроектовані на навчальне зображення з обличчям, що генерує глобальні форми обличчя [11].

На етапі навчання береться ряд навчальних вибірок, де кожне зображення містить ряд точок для кожної вибірки, будуючи форму для кожної з них. Потім фігури жорстко вирівнюються для обчислення середнього значення та коваріаційної матриці. Під час припасування PDM до обличчя, середнє значення форми розташовується в межах досяжності обличчя. Відповідно, виконується стратегія пошуку, названа стратегією пошуку в сірій шкалі, при деформації фігури до заданого обличчя. При цьому навчальна множина керує деформацією відповідно до того, як в ній змодельована інформація.

Основною проблемою такого підходу є відносна застарілість, вона не була адаптована до сучасних систем, також цей метод підходить тільки коли обличчя

не рухається та при гарному освітленні через те, що затемнене обличчя можна не вірно зчитати в крайніх його частинах.

Feed Forward Neural Network також відома як багатошарове сприйняття, вважається найпростішою формою нейронних мережа [12]. Нейронна мережа походить від персептронів. Персептрони - це методології мозку для зберігання та організації інформації. Інформація про риси обличчя рухається до вихідних вузлів від вхідних вузлів, де рух здійснюється через приховані шари [13, 14]. Приховані шари присвоюють вагу рисам обличчя в процесі навчання, як показано на рисунку 1.9 [14]. На етапі виявлення ваги порівнюються, щоб повідомити результат на заданому зображенні.

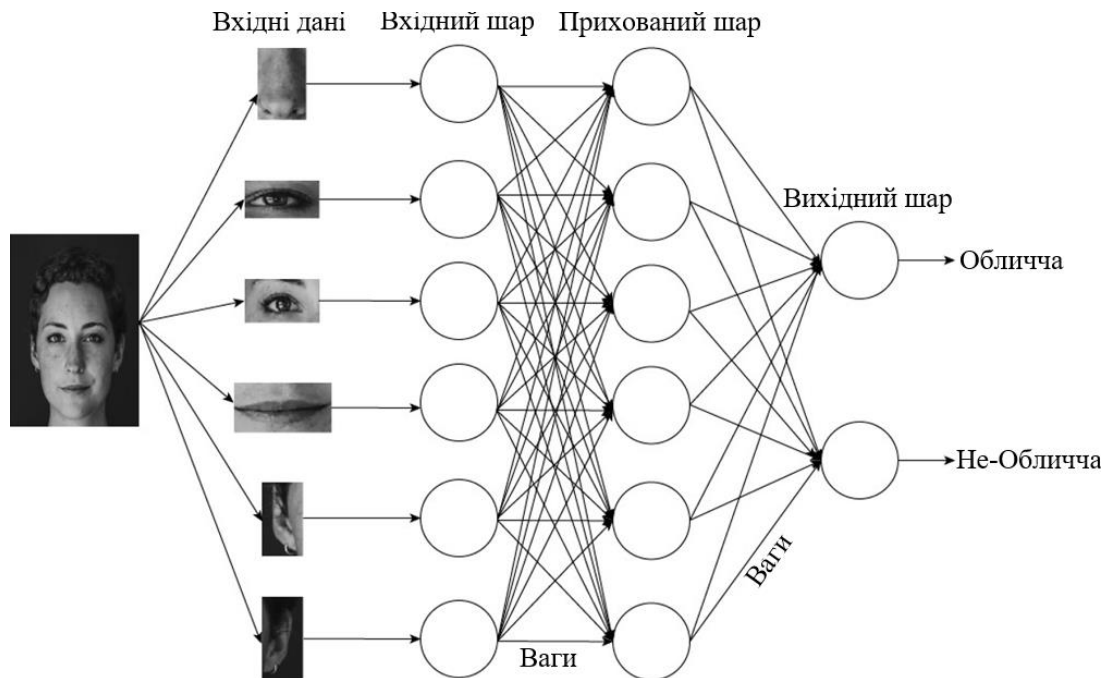


Рисунок 1.9 - Принципова схема нейронної мережі для розпізнавання облич

Будь-які дані зображення надходять в нейронну мережу, і випадкова вага ініціалізується з кожною ознакою обличчя, а потім налаштовується для розпізнавання облич. Проблеми починаються коли обличчя рухається і змазується зображення, також проблему створює мряка, через яку усі частини тіла можуть стати складними для розпізнавання.

Основною метою розпізнавання облич методом Eigenfaces є знаходження облич на заданому вхідному зображеннях, та як правило, є сильно зашумленими

[15, 16]. Шум створюється за рахунок пози, повороту, умов освітлення та інших інваріантів. Незважаючи на шуми, на зображенні існують певні закономірності. Зображення, що містить обличчя, зазвичай складається з патернів, завдяки наявності деяких об'єктів обличчя (очей, носа і т.д.). Ці риси обличчя називаються eigenface-си. Вони зазвичай отримуються із зображення за допомогою PCA. Використовуючи PCA, eigenface відповідних ознак будується з навчальної множини [17].

Поєднуючи всі eigenface-си в правильній пропорції, можна відновити оригінальне обличчя. Кожен eigenface відповідає одній ознаці обличчя.

Завдання алгоритму Eigenfaces - представити зображення як суму базисних компонент [18]:

$$\Phi_i = \sum_{j=1}^K \omega_j \mu_j, \quad (1.1)$$

де Φ_i - центроване (тобто за мінусом середнього) i -те зображення вихідної вибірки, ω_j - являє собою ваги і μ_j - власні вектори (eigenvectors або, в рамках даного алгоритму, eigenfaces).

Усі три методи мають свої переваги, й недоліки: Point Distribution Model дозволяє зекономити ресурси та легкий в надбудові нових функцій, нейронні мережі хоч і вважаються в майбутньому найкращим рішенням, але досі розвивають свою структуру і мають проблеми з хибним спрацюванням, зміною обличчя і освітленням приміщення, Eigenface теж має проблеми з хибним спрацюванням, але те що він аналізує закономірності то може впоратись з шумом на зображенні що підвищує якість, в той час як нейронні мережі не будуть мати з чим порівнювати в таких випадках, однак Point Distribution Model має не гірший результат при дотриманні норм використання.

Тож як окрему реалізацію ідентифікації і аутентифікації обличчя краще обрати або Eigenface або Point Distribution Model, Eigenface потребуватиме

більше витрат, підтримки та часу на реалізацію, Point Distribution Model може мати гірші результати, але реалізується простіше.

1.7 Висновки за розділом

У даному розділі магістерської роботи проаналізована доступна інформація про системи контролю робочого часу, одною з головних задач систем є аутентифікація - встановлення автентичності "один в один" та ідентифікація - пошук відповідності "один з багатьох".

Для виконання завдання ідентифікації та аутентифікації потрібно скористатись допомогою відповідних ідентифікаційних систем, з розділу отримано висновок, що біометричні системи розпізнавання облич є чудовим варіантом для реалізації.

Серед розглянутих в розділі технологій розпізнавання облич найкраще себе показує Point Distribution Model, оскільки для ідентифікації використовуються опорні точки на зображенні та різниця між двома зображеннями людей, така система дозволяє інтегрувати її на будь який об'єкт.

2 СТРУКТУРА ТА ФУНКЦІЇ СИСТЕМИ ОБЛІКУ РОБОЧОГО ЧАСУ З ВИКОРИСТАННЯМ БІОМЕТРІЇ ОБЛИЧЧЯ

2.1 Структура та склад системи

2.1.1 Загальний опис системи обліку робочого

Оскільки просто знаходження на робочому місці недостатньо для того щоб вважати що працівник провела свій час продуктивно, а не, наприклад, в телефоні. В архіві також знаходиться папка з фото обличчя. Найчастіше по обличчю можна зрозуміти чим саме займається працівник, тому ця інформація є істотною для зберігання. Інформація про продуктивність накопичується впродовж місяця, в кінці місяця робиться повний аналіз продуктивності за місяць шляхом об'єднання таблиць за кожен робочий день і формується таблиця продуктивності схожа на таблицю на рисунку 1.5.

На базі розділу 1.2 було розроблено наступну систему контролю робочого часу яка буде використовуватись в цій роботі та для тестування:

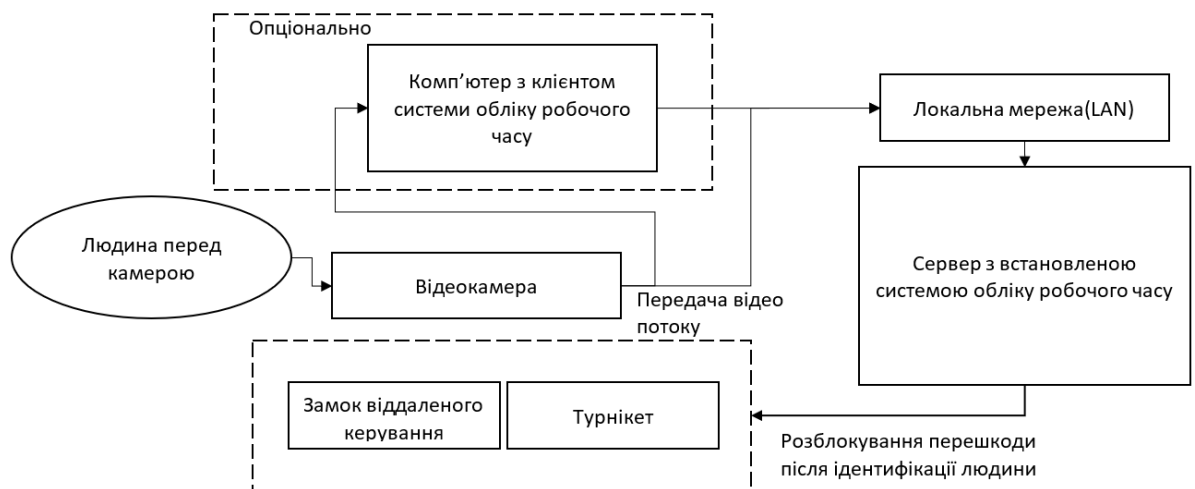


Рисунок 2.1 - Структура системи обліку робочого часу

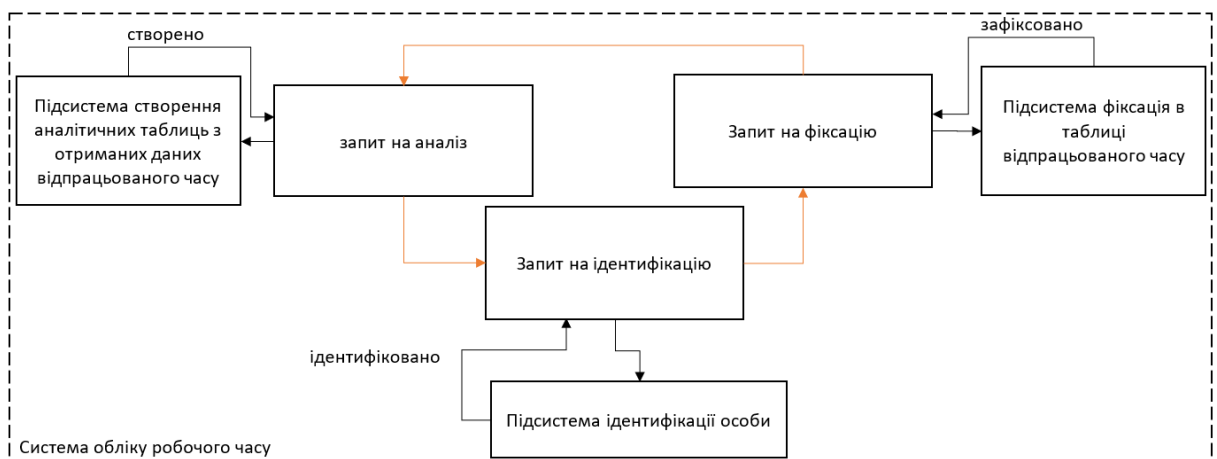
На рисунку 2.1 бачимо, що система обліку робочого часу буде працювати наступним чином:

Людина з'являється в об'єктиві камери в коридорі. Доки людина не буде розпізнана на камері як працівник компанії, вона не зможе отримати доступ до приміщення. Після розпізнання, працівник може потрапити всередину, час

входження працівника в приміщення фіксується для майбутнього аналізу продуктивності, далі працівник сідає за своє робоче місце, кожні 300 мілісекунд система обліку робочого часу фіксує наявність працівника за робочим місцем. Кожні десять хвилин одне зображення зберігається в папці з ПІБ працівника, відсутність і присутність працівника фіксується в окремому текстовому документі, щохвилинно, шляхом додавання одиниці при кожному спрацюванні системи обліку робочого часу в відповідну таблицю присутності та відсутності. В кінці робочого дня система встановлена на комп'ютер відправляє .zip архів на сервер через локальну мережу, в архіві знаходиться інформація про продуктивність проведеного дня, а саме скільки часу працівник знаходилась на робочому місці [7].

Невід'ємною частиною системи розпізнавання облич є база облич, мінімальна рекомендована кількість 10. Але в залежності від реалізації і параметрів заліза комп'ютера і сервера, цю кількість можна і варто збільшувати. Важливо зазначити, що кількість операцій порівняння пропорційно збільшується з збільшенням кількості працівників і бази зображень для кожного з цих працівників [7].

На рисунку 2.2 зображено структурну схему системи обліку робочого часу.



Рисунку 2.2 - Функції системи обліку робочого часу

Систему обліку робочого часу можна представити у вигляді серії запитів до різних підсистем. Кожна підсистема не знає про існування іншої, а лише про свої задачі та те що має бути отримано.

Якщо якась з підсистем отримує невірні данні для опрацювання чи під час роботи підсистеми виникає помилка, то інформація про це приходить у систему обліку робочого часу, де адміністратор системи обліку робочого часу може опрацювати та виправити проблему.

В цій системі кількість працівників дорівнює 20 і для кожного працівника було створено базу зображень, по 14 на кожного працівника.

Для того щоб база зображень давала більший ефект усі зображення мають бути зроблені при різних нахилах, та емоціях обличчя, для того щоб система мала більшу варіативність для порівняння зображення з відеопотоку.

В кінці чи впродовж дня коли працівнику буде потрібно вийти з офісу вона має ідентифікувати свою особу на камері, інакше двері будуть зачинені.

За роботою системи обліку робочого часу має слідкувати адміністратор системи обліку робочого часу. В обов'язки якого входить адміністрування, перевірка і корегування системи обліку робочого часу в випадку несправності, аналіз результатів системи та надання цих результатів начальникам відділів. В можливості адміністратора також має входити можливість покращення системи шляхом додавання нових функцій, наприклад, замість зберігання фотографій робити авто аналіз зображень, на предмет неефективного проведення часу чи аналіз сайтів, якими користувався окремий працівник.

Частиною обліку робочого часу є система ідентифікації людини, яка кожні 300 мілісекунди розпізнає обличчя на кадрі відеопотоку.

Системи обліку робочого часу можуть значно покращити показники працівників шляхом підрахунку продуктивності протягом місяця, таким чином власникам компанії, керуючому персоналу стає відомо хто з працівників незадовільно витрачає час і потребує або заміни, або покращення умов.

2.1.2 Підсистема ідентифікації особи

Підсистема ідентифікації з використання біометрії обличчя є основною темою цієї роботи, в загальному підсистема ідентифікації особи – це взаємодія системи ідентифікації обличчя та наступного аналізу цього обличчя за допомогою бази даних облич, в якій знаходяться обличчя кожного працівника в кількості 14 штук рис. 2.3.

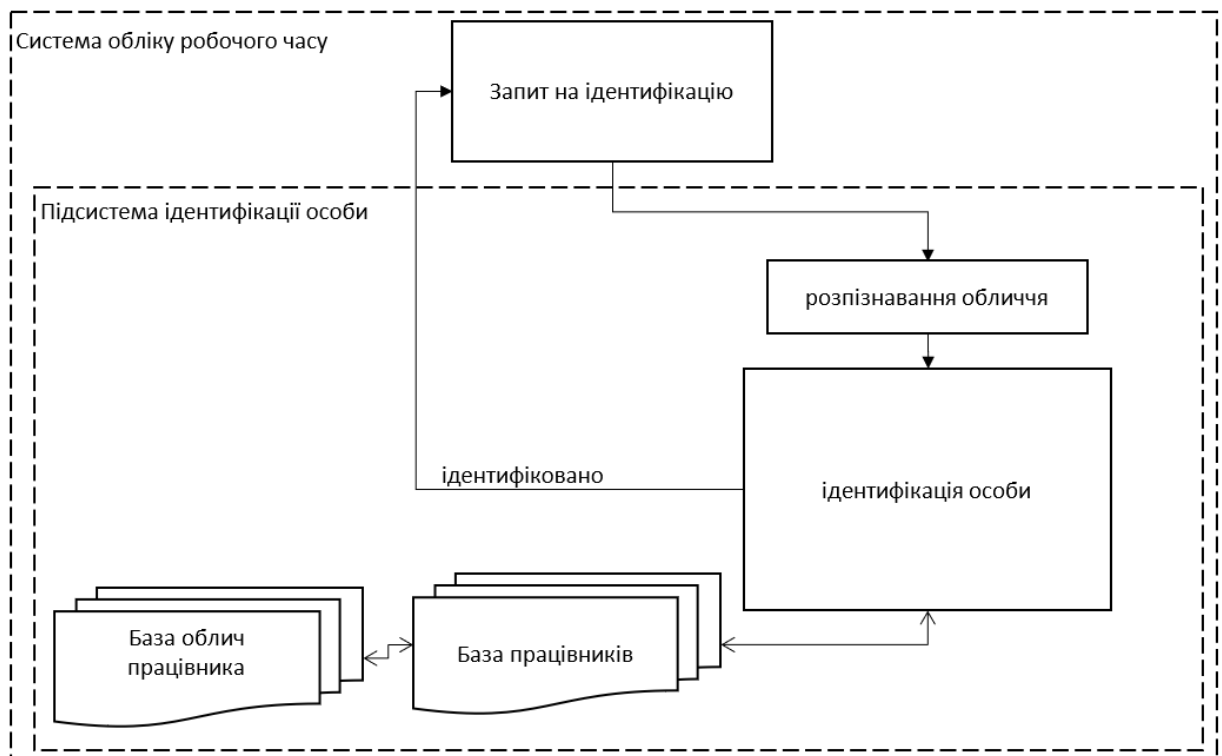


Рисунок 2.3 - Підсистема ідентифікації особи в складі системи обліку робочого часу

Для розпізнання обличчя з відеопотоку було використано метод Point Distribution Model [10, 11], а саме реалізацію на JavaScript під назвою face-api.js, більш детально про роботу бібліотеку face-api.js в наступних розділах.

Для ідентифікації особи було використано евклідову відстань між зображенням в базі облич та зображення з відеопотоку, принцип розрахунку евклідової відстані буде описаний в наступних розділах.

При роботі підсистеми ідентифікації особи доступні 3 основні відповіді:

- Особу ідентифіковано – ім'я та евклідова відстань;
- Особу не ідентифіковано – “unknown” та евклідова відстань;

- Обличчя не розпізнано – “-”.

Основною перевагою систем ідентифікації обличчя над іншими системами ідентифікації в тому, що людині не треба взаємодіяти з контролером фізично, в більшості випадків достатньо пройти повз камеру, що зменшує час взаємодії, не впливаючи на захищеність об'єкта [10, 11].

Головне завдання підсистеми ідентифікації – розпізнати працівника та надіслати системі обліку робочого часу інформацію про те, хто перед камерою.

Важливо розуміти, що підсистема ідентифікації, як і інші програми, використовує пам'ять процесора та оперативну пам'ять, тому необхідно оптимізувати систему, а саме:

- Підсистема ідентифікації не має споживати багато пам'яті;
- Підсистема ідентифікації має розпізнавати працівників в різних умовах.

При збільшенні кількості працівників - кратно збільшиться кількість операцій по розпізнаванню, для оптимізації варто розділити персонал по відділах, тим самим розділивши базу облич на відділи, зменшивши навантаження на систему.

2.1.3 Параметри приміщення

В якості приміщення для реалізації системи обліку робочого часу була обрана кімната з довжиною 5м та шириною кімнати 3м, висота стелі 2,7м, ці значення будуть необхідні в майбутньому для розрахунків і тестування. Оскільки завданням роботи є дослідження можливостей використання біометрії обличчя в системі обліку робочого часу, було вирішено вважати в усіх розрахунках офісом для тестування. Отже маємо наступну схему:

На рисунку 2.4 зображена схема офісу, де:

- 1 – довжина кімнати 5м,
- 2 – ширина кімнати 3м,
- 3 – робоче місце працівника,
- 4 – сервер,
- 5 – відеокамера,
- 6 – кабель передачі відеопотоку з камери на сервер,

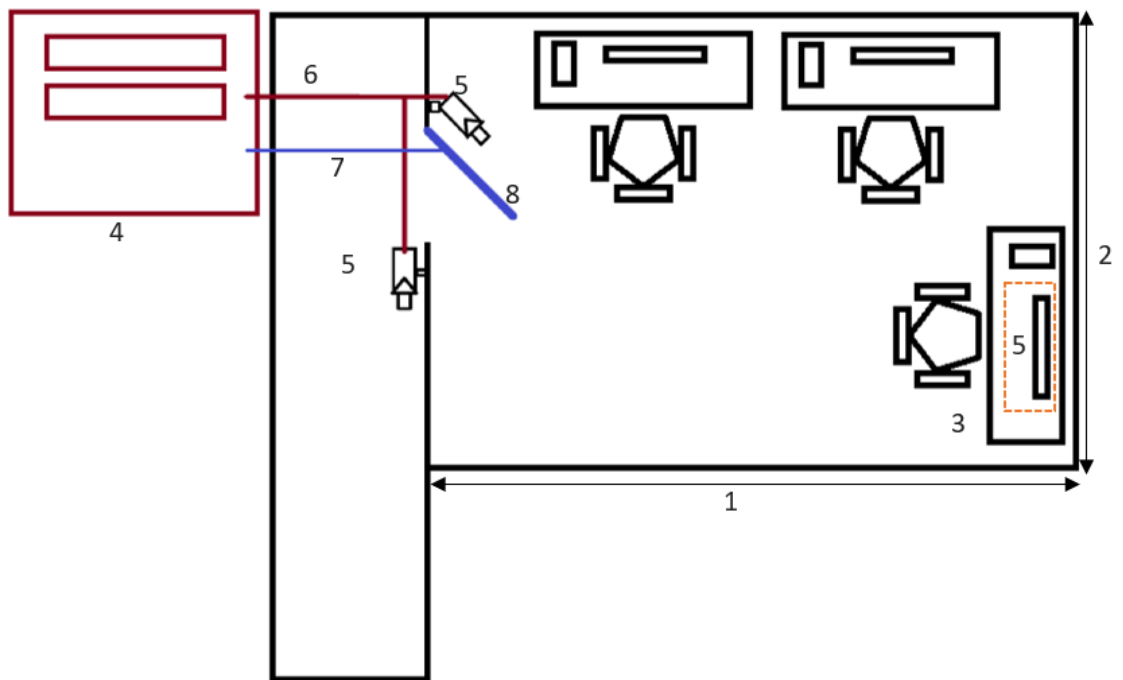


Рисунок 2.4 - Приклад схеми офісу

7 – кабель передачі даних,

8 – двері з замком віддаленого керування.

В офісі є три основні частини:

- сервер, на якому зберігається та аналізується інформація про ідентифікованих, та неідентифікованих людей протягом певного часу;
- комп'ютер працівника;
- камера, яка передає відеопотік на сервер або в комп'ютер працівника.

Система ідентифікації встановлена як на комп'ютер, так і на сервер, таким чином сервер може аналізувати інформацію про людей які пройшли по коридору чи в офісі, а з комп'ютерів працівників передається вже проаналізована інформація для зберігання на сервері.

Як видно з рисунку 2.4 камери встановлені не тільки в коридорі, а й в монітори комп'ютерів працівників, завдяки комбінації камери в моніторі та системи ідентифікації в комп'ютері працівника можна в реальному часі аналізувати продуктивність працівника, наприклад проаналізувати присутність працівника на робочому місці впродовж робочого дня після чого створити

таблицю аналізу місцезнаходження працівника впродовж робочого дня, як на рис. 1.5.

2.2 Методика розпізнавання обличчя

2.2.1 Описання метода розпізнавання обличчя

Як вже було зазначено існує багато методів розпізнавання обличчя. Був обраний метод моделі активних форм через те, що вони можуть працювати зі складними формами шляхом деформації відповідно до заданого прикладу за допомогою ітеративної обробки. Оскільки коли система переглядає зображення, вона розшукує на ньому опорні риси обличчя, такі як рот, ніс, очі і т.д., як тільки знаходить близьке сусідство з будь-якою з цих рис. координати цих частин беруться як карта, і звідси генерується маска. Тренуючись на більшій кількості зображень, можна досягти кращої карти.

З усіх методів моделі активних форм була обрана Point Distribution Model. Це техніка опису форми [10, 11], яка спирається на опорні точки, що є дуже зрозуміло для користувача і має високу точність розпізнавання облич. Опорна точка - це анотація будь-якого зображення на будь-яку задану форму на зображеннях навчальної вибірки. Форма обличчя в Point Distribution Model формується шляхом нанесення опорних точок на форму обличчя на навчальному наборі зображень які були занесені в базу даних облич. Модель, як правило, будується з глобальною формою обличчя, що має утворення очей, вух, носа та інших елементів обличчя рис. 2.5.

Для розрахунку положення опорних точок необхідно розрахувати дескриптори для кожної опорної точки зображення, в деяких методах розрахунку кількість дескрипторів переважає 128 на кожну точку.

Дескриптор - це опис конкретної точки на зображенні, що визначає характеристики її оточення, являє собою числовий або двійковий вектор певних параметрів. Довжина вектора і тип параметрів визначаються використовуваним алгоритмом. Дескриптор дає можливість виділити конкретну точку із сукупності точок на зображенні. За допомогою дескрипторів дуже зручно виконувати

порівняння різних зображень, для цього повинні бути складені опорні пари ознак, що належать одному і тому ж обличчю.

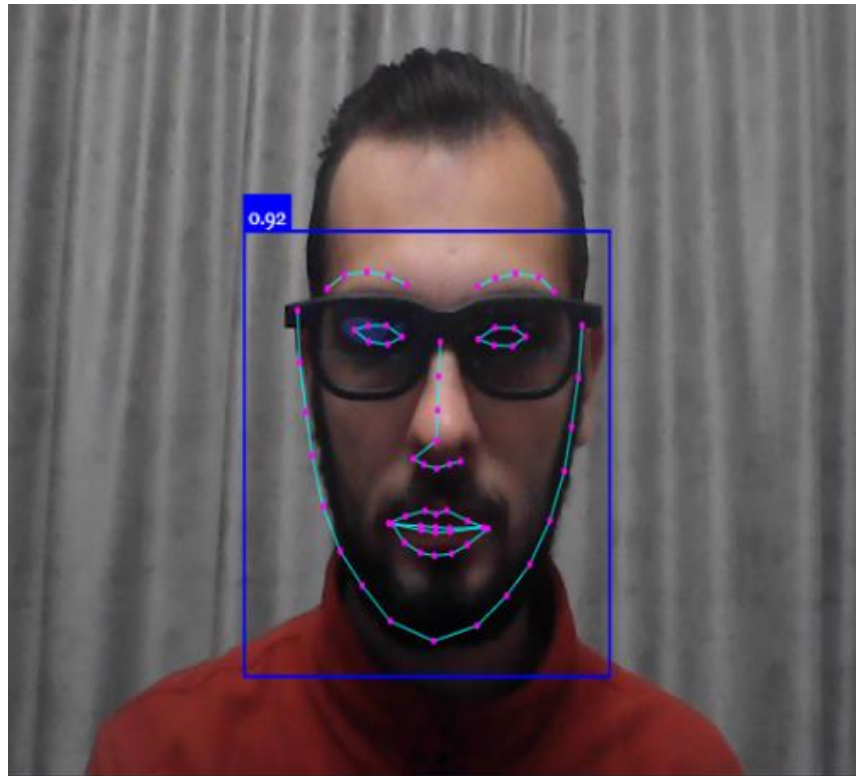


Рисунок 2.5 - Демонстрація опорних точок на зображенні з відеопотоку методом Point Distribution Model

Для того щоб порівняти дескриптори різних фотографій на предмет ідентичності облич зображених на фотографії використовується метод розрахунку Евклідової відстані між дескрипторами зображень.

Завдяки тому, що Point Distribution Model [10, 11] аналізує основні риси обличчя, без урахування верхнього контура обличчя, як показано на рис. 2.5.

Отримано висновок що будь які головні убори, за виключенням тих, які приховують очі, брова, ніс не є перепоною для роботи системи розпізнавання.

Вплив відсутності частини обличчя на якість розпізнавання облич та розпізнавання особи буде розглянуто в тестовій частині роботи.

2.2.2 Пошук опорних точок

SURF являється найбільш відомим алгоритмом, принцип його роботи полягає в наступному. Для зображення сцени та зображення еталона за допомогою

методу SURF знаходяться опорні точки та унікальні дескриптори для них. Порівнюючи ці набори дескрипторів, буде можливо виділити еталонний об'єкт на сцені [19].

SURF вирішує два завдання - пошук опорних точок зображення і створення їхніх дескрипторів, інваріантних до масштабу і обертання. Метод шукає опорні точки за допомогою матриці Гессе. Детермінант матриці Гессе - гессіан, досягає екстремуму в точках максимальної зміни градієнта яскравості. Він добре розпізнає плями, кути та краї ліній. Гессіан інваріантний щодо обертання. Але не інваріантний до масштабу. Тому SURF використовує різномасштабні фільтри для знаходження гессіанів.

Для кожної опорної точки рахується напрямок максимальної зміни яскравості - градієнт, і масштаб, взятий із масштабного коефіцієнта матриці Гессе.

Градієнт у точці обчислюється за допомогою фільтрів Хаара. Після знаходження опорних точок, SURF формує їхні дескриптори. Дескриптор являє собою набір із 64 (або 128) чисел для кожної опорної точки. Ці числа відображають флуктуації градієнта навколо опорної точки [20].

Оскільки опорна точка є максимумом гессіана, то це гарантує, що в околиці точки мають бути ділянки з різними градієнтами. Таким чином, забезпечується дисперсія (відмінність) дескрипторів для різних опорних точок. Флуктуації градієнта околиць опорної точки вважаються відносно напрямку градієнта по всій околиці опорної точки. Таким чином, досягається інваріантність дескриптора щодо обертання. Розмір же області, на якій рахується дескриптор, визначається масштабом матриці Гессе, що забезпечує інваріантність щодо масштабу.

Флуктуації градієнта також рахуються за допомогою фільтра Хаара.

Для ефективного обчислення фільтрів Гессе і Хаара - використовується інтегральне представлення зображень.

В скороченому вигляді, інтегральне представлення є матрицею, розмірність якої збігається з розмірністю вихідного зображення, а елементи рахуються за формулою [21]:

$$I(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j), \quad (2.1)$$

де $I(i, j)$ - яскравість пікселів вихідного зображення.

Маючи інтегральну матрицю, можна дуже швидко обчислювати суму яскравостей пікселів довільних прямокутних областей зображення, за формулою:

$$SumOfRect(ABCD) = I(A) + I(C) - I(B) - I(D) \quad (2.2).$$

Обчислення матриці Гессе. Виявлення опорних точок у SURF ґрунтується на обчисленні детермінанта матриці Гессе (гессіана).

Матриця Гессе для двовимірної функції та її детермінант визначається таким чином:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

$$\det(H) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left(\frac{\partial^2 f}{\partial x \partial y} \right)^2. \quad (2.3)$$

Значення гессіана використовується для знаходження локального мінімуму або максимуму яскравості зображення. У цих точках значення гессіана досягає екстремуму. Теоретично, обчислення матриці Гессе зводиться до знаходження Лапласіана Гауссіана.

По суті, елементи матриці Гессе обчислюються як сума добутків пікселів зображення на фільтри, зображені на рис. 2.6.

На рисунку 2.6 зображено дискретизовані фільтри для знаходження чотирьох елементів матриці Гессе (четвертий - збігається з третім, оскільки матриця Гессе

симетрична). Фільтри мають просторовий масштаб 9×9 пікселів, де темні ділянки відповідають від'ємним значенням фільтра, світлі - позитивним.

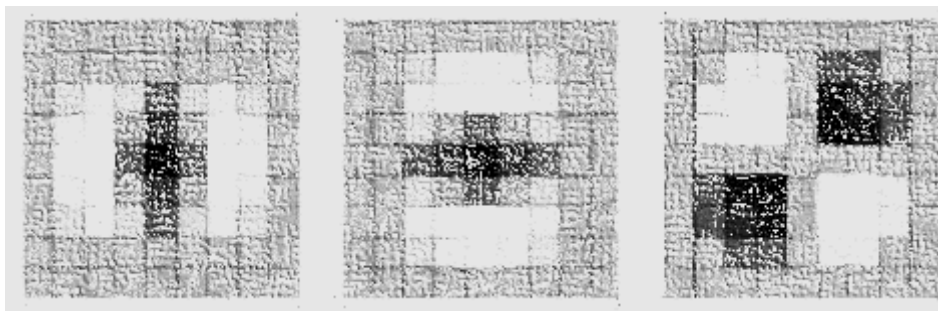


Рисунок 2.6 - Дискретизовані фільтри

Однак, SURF не використовує лапласіан гауссіани в тому вигляді, який зображений на малюнку. По-перше, за твердженням авторів, дискретизований лапласіан гауссіани має доволі великий розкид значення детермінанту, при обертанні зразка, що важливо пам'ятати, що в ідеалі гессіан має бути інваріантний до обертання. Особливо детермінант "просідає" в районі повороту на 45 градусів. А по-друге, і це головне, фільтр для лапласіана гауссіани має безперервний характер. Майже всі пікселі фільтра мають різні величини яскравості. А це не дає змоги ефективно використовувати такий потужний механізм розрахунку, як інтегральну матрицю зображення. Тому SURF використовує бінаризовану апроксимацію лапласіана гауссіан, під назвою Fast-Hessian:

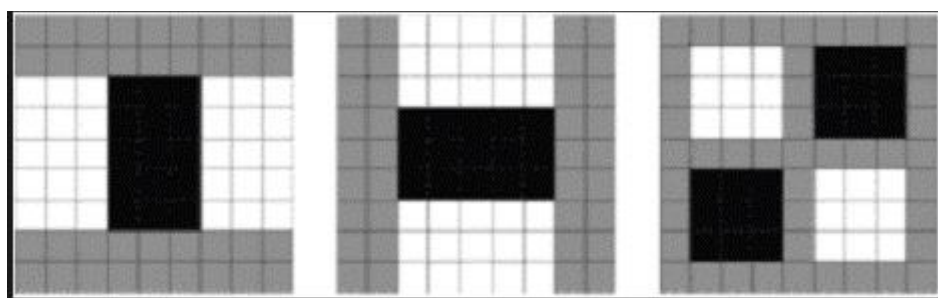


Рисунок 2.7 - фільтри матриці Гессе

На рис. 2.7 зображено фільтри, що використовуються для знаходження матриці Гессе в SURF. Білі області відповідають значенню $+1$, чорні -2 (на третьому фільтрі -1), сірі - нульові. Просторовий масштаб - 9×9 пікселів.

Цей фільтр більш стійкий до обертання, і його можна ефективно обчислити за допомогою інтегральної матриці [21].

Таким чином, у SURF, гессіан обчислюється так:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2, \quad (2.4)$$

де D_{xx} , D_{yy} , D_{xy} - згортки за фільтрами, зображеними на малюнку вгорі. Коефіцієнт 0.9 має теоретичне обґрунтування, і коригує наближений характер обчислень.

Отже, для знаходження опорних точок, SURF пробігається по пікселях зображення і шукає максимум гессіана. Спосіб знаходження локального максимуму гессіана буде розглянуто пізніше. У методі задається порогове значення гессіана. Якщо обчислене значення для пікселя вище за поріг - піксель розглядається як кандидат на опорну точку.

Тут ще корисно зауважити таке. Оскільки гессіан є похідною, і залежить тільки від перепаду яскравості, але не від абсолютного її рівня, то він інваріантний щодо зсуву яскравості зображення. Таким чином, зміна рівня освітлення зразка не впливає на виявлення опорних точок.

Як уже зазначалося, гессіан не інваріантний щодо масштабу. Це означає, що для одного і того ж пікселя, гессіан може змінюватися при зміні масштабу фільтра.

Розв'язання цієї проблеми тільки одне - перебирати різні масштаби фільтрів і по черзі їх застосовувати до даного пікселя.

З міркувань симетрії і дискретизації, розмір фільтра Fast-Hessian не може приймати довільні значення. Допустимі розміри цього фільтра такі, починаючи з мінімального: 9, 15, 21, 27 і так далі, з кроком 6. Однак, на практиці, поступово збільшувати розмір фільтра на 6 - не вигідно, бо для великих масштабів крок 6 виявляється занадто дрібним, а фільтри - надлишковими. Тому, SURF розбиває всю безліч масштабів на так звані октави. Кожна октава покриває певний інтервал масштабів, і має свій характерний розмір фільтра.

При цьому якби на октаву припадав тільки один фільтр, це було б занадто грубим наближенням. Крім того, було б не могли знайти локальний максимум гессіана, серед різних масштабів, у різних октавах. Адже одна й та сама точка може мати кілька локальних максимумів гессіана, у різних масштабах.

При пошуку максимум серед усіх гессіанів, за всіма масштабами, то знайшли б тільки один із максимумів, тоді як їх може бути кілька. Один - в одному масштабі, інший - в іншому.

Виходячи з перерахованого, октава містить не один фільтр, а чотири фільтри, які добре покривають характерний масштаб октави.

2.2.3 Обчислення дескриптора опорних точки

Дескриптор являє собою масив із 64, у розширеній версії, що використовується в цій роботі, - 128 чисел, які дають змогу ідентифікувати опорну точку. Дескриптори однієї й тієї самої опорної точки на зразку і на сцені мають приблизно збігатися. Метод розрахунку дескриптора такий, що він не залежить від обертання і масштабу. Для обчислення дескриптора навколо опорної точки формується прямокутна область, що має розмір $20s$, де s - масштаб, у якому було знайдено опорну точку. Для першої октави, область має розмір 40×40 пікселів. Квадрат орієнтується вздовж пріоритетного напрямку, обчисленого для опорної точки.

Дескриптор вважається як опис градієнта для 16 квадрантів навколо опорної точки.

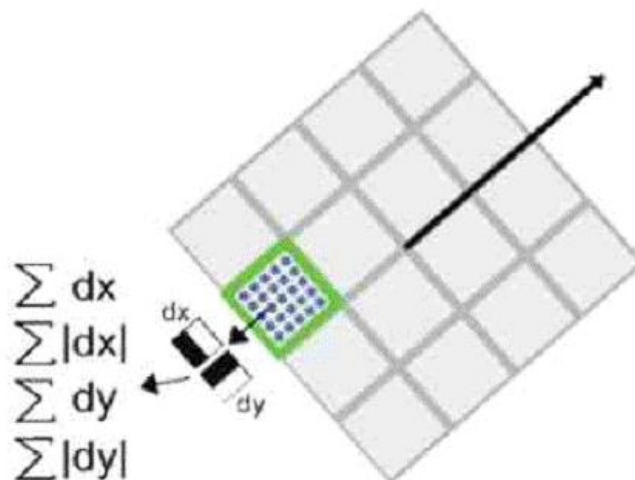


Рисунок 2.8 Розрахунок дескриптора

Далі, квадрат розбивається на 16 дрібніших квадрантів, як показано на малюнку. У кожному квадранті береться регулярна сітка 5×5 і для точки сітки шукається градієнт, за допомогою фільтра Хаара. Розмір фільтра Хаара береться рівним $2s$, і для першої октави становить 4×4 .

Слід зазначити, що під час розрахунку фільтра Хаара зображення не повертається, фільтр рахується у звичайних координатах зображення. А ось отримані координати градієнта (dX, dY) повертаються на кут, що відповідає орієнтації квадрата. Отже, для обчислення дескриптора опорної точки потрібно обчислити 25 фільтрів Хаара, у кожному з 16 квадрантів. Разом, 400 фільтрів Хаара. З огляду на те, що на фільтр потрібно 6 операцій, виходить, що дескриптор обійдеться мінімум у 2400 операцій.

Щоб знайти 25 точкових градієнтів квадранта, обчислюються чотири величини, які власне і є компонентами дескриптора: $\sum dX$, $\sum |dX|$, $\sum dY$, $\sum |dY|$.

Дві з них є просто сумарний градієнт по квадранту, а дві інших - сума модулів точкових градієнтів.

Чотири компоненти на кожен квадрант, і 16 квадрантів, дають 64 компоненти дескриптора для всієї області опорної точки. Під час занесення в масив, значення дескрипторів зважуються на гауссіану, з центром в опорній точці і з сигмою $3.3s$. Це потрібно для більшої стійкості дескриптора до шумів у віддалених від опорної точки областях.

Плюс до дескриптора, для опису точки використовується знак сліду матриці Гессе, тобто величина $\text{sign}(D_{xx} + D_{yy})$. Для світлих точок на темному тлі, слід негативний, для темних точок на світлому тлі - позитивний. Таким чином, SURF розрізняє світлі й темні плями.

Таким чином застосувавши SURF до зображення, отримуємо набір дескрипторів, який буде унікально ідентифікувати еталон на сцені.

Переваги методу:

- Інваріантний до поворотів і масштабування
- Інваріантний до різниці загальної яскравості зображень
- Може детектувати більше 1 об'єкта на сцені

Недоліки методу:

- Досить складний у реалізації
- Відносно повільна робота алгоритму

2.2.4 Розрахунок евклідової відстані по значенням дескрипторів

Як вже було зазначено – евклідова відстань це метрика, яка використовується для визначення відстані між точками або векторами в багатовимірному просторі.

Загальна формула розрахунку евклідової відстані має наступний вигляд [22]:

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}, \quad (2.4)$$

де a, b – дві точки або вектори між яким треба розрахувати відстань, n – мірність простору, в якому проводиться розрахунки.

Оскільки при розрахунку дескрипторів опорних точок створюється по 128 значень дескриптора для кожної точки повинні використати формулу евклідової відстані для кожної з 68 опорних точок, де $n = 128$ значень дескрипторів, a, b – i -те значення дескриптора на фото і в базі обличч рис. 2.9.

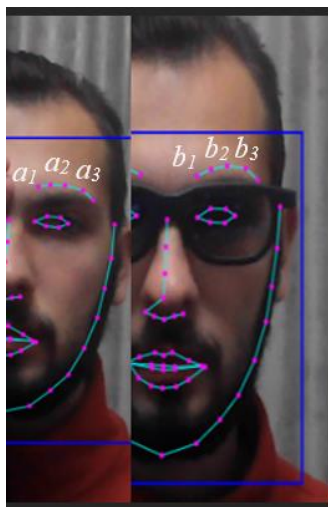


Рисунок 2.9 - Зображення опорних точок порівняння на обличчі

Після розрахунку евклідової відстані для усіх 68 пар дескрипторів ці значення додаються та сума ділиться на 68, таким чином отримуємо значення загальної евклідової відстані для зображення, яке буде в майбутньому відображатися на екрані монітора під час роботи системи.

Важливо зазначити, що такий метод є дуже надійним. Вважається, що загальна евклідова відстань, яка менша 0.6 дає високу вірогідність того, що на порівняних фото - одна й та сама людина. При збільшенні опорних точок та більш глибокого навчання системи на розпізнавання облич, можна отримати результат евклідової відстані меншій 0.3. Те як впливають зовнішні і внутрішньо-системні чинники на розрахунки евклідової відстані буде визначено під час тестування системи.

Point Distribution Model хоч і старий, але досі ефективний метод ідентифікації та аутентифікації людини, яка є:

1. дешевою в розробці,
2. простою в розумінні,
3. надійною в використанні,

через що цей метод досі є популярним в розробці систем обліку робочого часу.

Розрахунок евклідової відстані через дескриптори опорних точок є сучасним методом порівняння зображень, достатньо надійним і швидким, щоб його використовували в різних проектах обліку робочого часу.

2.3 Висновки за розділом

В цьому розділі було розглянуто структуру системи обліку робочого часу та структуру підсистеми ідентифікації особи.

Було зазначено параметри офісу.

Було описано методику ідентифікації обличчя яка була використана в цій роботі.

Також було описано розрахунок дескрипторів для кожної опорної точки та як розраховується евклідова відстань між дескрипторами.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІДЕНТИФІКАЦІЇ ОСОБИ

3.1 Вибір середовища та засобів розробки

Для створення програми ідентифікації було використано мову програмування JavaScript [23] в середовищі Visual Studio Code [24].

Таке рішення було пов'язано з тим, що в JavaScript доступне використання системи модулів – бібліотек з кодом які запрограмовані виконувати певні дії і методи, таким чином можна створити глобальну бібліотеку, яка буде реалізовувати модулі.

Visual Studio Code – зручне і спеціально оптимізоване середовище для JavaScript, в нього автоматично встановлені усі необхідні налаштування для JavaScript і тому з ним можна працювати одразу після встановлення Visual Studio Code.

Додатково в Visual Studio Code було встановлено Live Server - це живий сервер швидкої розробки з перезавантаженням браузера в реальному часі. З його допомогою можна запускати створену програму в локальній мережі.

Для реалізації ідентифікації обличчя в JavaScript достатньо моделей, які реалізують пошук облич і бази облич, на відміну від інших мов. Де додатково треба встановити окремі системи пошуку облич.

Для побудови програми ідентифікації облич було використано face-api.js [25], модуль JavaScript, побудований поверх ядра tensorflow.js [26] який реалізує декілька інших модулів. Які взаємодіють між собою для розв'язання задач виявлення облич, розпізнавання облич і виявлення орієнтирів обличчя, оптимізованих для мережі та для мобільних пристроїв.

В цій роботі було використано 2 основні модулі для розпізнавання облич - ssdMobilenetv1 та Tiny Face Detector, ssdMobilenetv1 була необхідна для більш точного розпізнавання облич в базі даних, перед тим як програма почне розпізнавати людей з відеопотоку, однак для розпізнавання людей з відеопотоку було вирішено використати Tiny Face Detector для більш швидкого отримання результатів, з меншим навантаженням на комп'ютер.

3.2 Основні алгоритми програми

Програма ідентифікації має спочатку розрахувати усі опорні точки і дескриптори для облич в базі даних. Після цього шукає обличчя на зображенні з відеокамери. На рис. 3.1 зображено узагальнений алгоритм роботи програми ідентифікації. Принцип роботи підпрограми виконання передумов зображено на рис. 3.2.

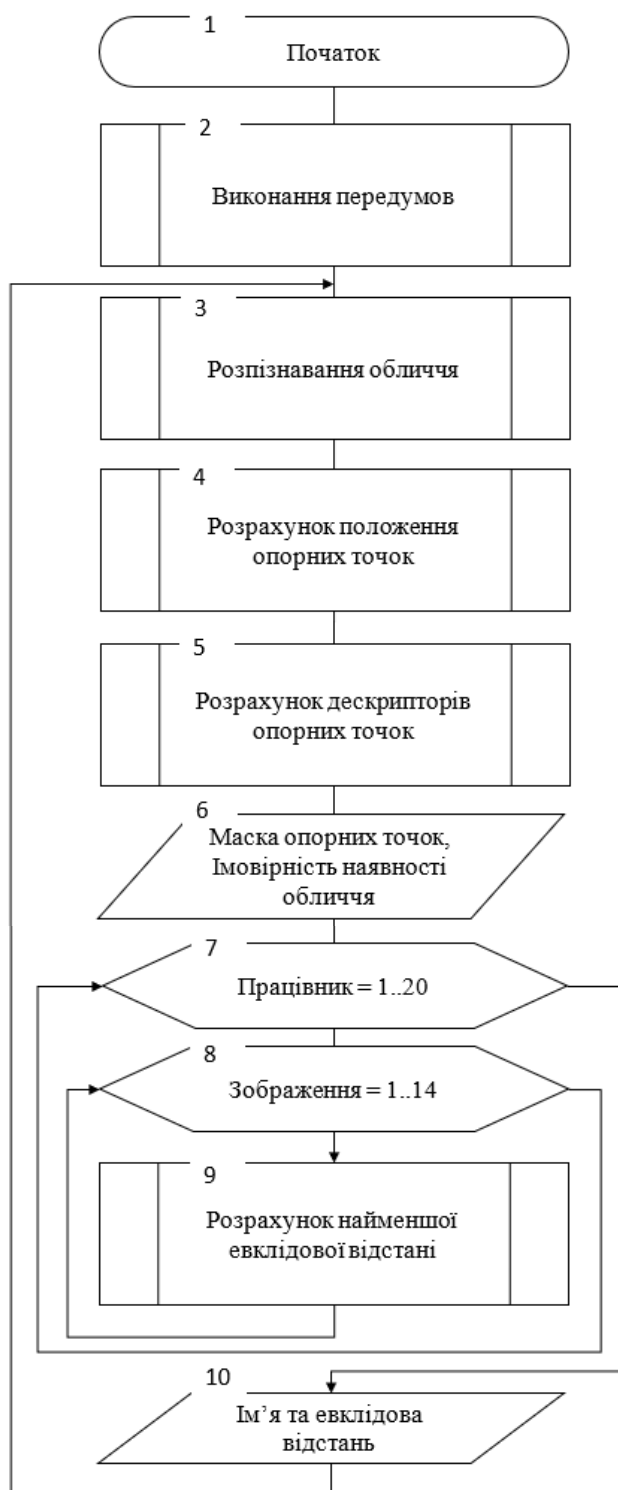


Рисунок 3.1 - Узагальнений алгоритм роботи програми ідентифікації

Блок 1 – початок роботи програми ідентифікації починається разом з стартом системи обліку робочого часу;

Блок 2 – виконання передумов – розрахунок дескрипторів для зображень в базі даних, дивись рис. 3.2;

Блок 3 – за допомогою модуля Tiny Face Detector знаходить на зображенні область з обличчям;

Блок 4 – за допомогою модуля faceLandmark68Net, Tiny Face Detector розраховує положення для 68 опорних точок на обличчі;

Блок 5 – за допомогою модуля Face Recognition Model, Tiny Face Detector розраховує 128 значень дескрипторів для кожної опорної точки;

Блок 6 – виведення на екран форми, з імовірністю наявності обличчя;

Блок 7 – цикл по усіх працівниках в базі, в цій роботі – 20;

Блок 8 – цикл по усіх зображеннях працівників в базі, в цій роботі – 14;

Блок 9 – розрахунок найменшого евклідового числа серед усіх зображень;

Блок 10 – виведення на екран форми, з ім'ям працівника та евклідовою відстанню.

На рис. 3.2 зображені наступні блоки:

Блок 1 – початок алгоритму виконання передумов;

Блок 2 – цикл по усіх працівниках в базі, в цій роботі – 20;

Блок 3 – цикл по усіх зображеннях працівників в базі, в цій роботі – 14;

Блок 4 – за допомогою модуля ssdMobilenetv1 знаходить на зображенні область з обличчям;

Блок 5 – за допомогою модуля faceLandmark68Net, ssdMobilenetv1 розраховує положення для 68 опорних точок на обличчі;

Блок 6 – за допомогою модуля ssdMobilenetv1, Tiny Face Detector розраховує 128 значень дескрипторів для кожної опорної точки [25];

Блок 7 – кінець алгоритму виконання передумов.

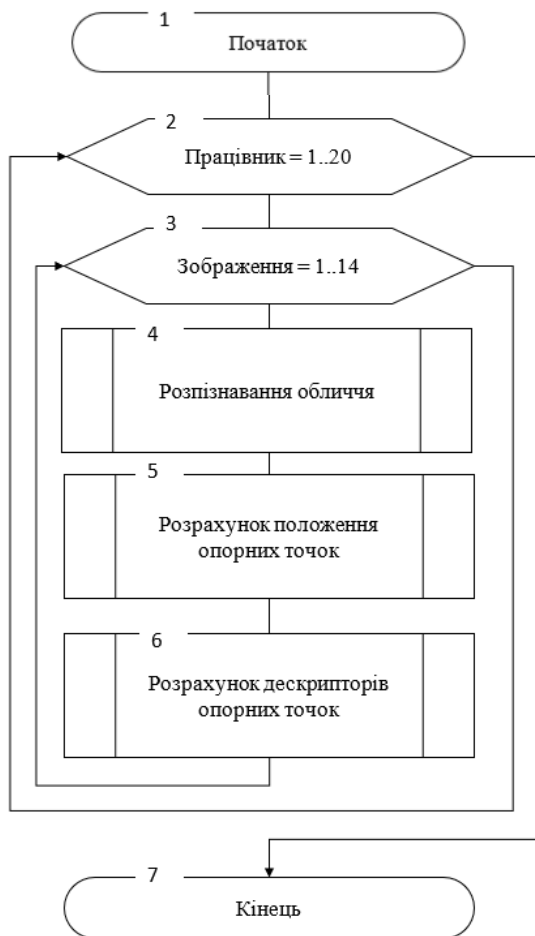


Рисунок 3.2 - Алгоритм роботи підпрограми «виконання передумов»

3.3 Основні моделі, що використовуються в програмі

3.3.1 SSDMobilenetv1

Single Shot Multibox Detector вважається найточнішим детектором обличчя, який по суті є Point Distribution Model на основі MobileNet V1, з деякими додатковими шарами прогнозування блоків, що накладаються поверх мережі. У цій роботі ця модель використовується для аналізу зображень, які вже знаходяться в базі облич, за рахунок цього програма запускається довше, але місцезнаходження опорних точок визначається точніше [25].

Модель буде обчислювати розташування кожного обличчя на зображенні і повертатиме обмежувальні рамки разом з ймовірністю для кожного обличчя. Цей детектор облич спрямований на отримання високої точності у виявленні обмежувальних рамок, що коштувало йому низького часу виведення. Розмір квантованої моделі становить близько 5,4 МБ (ssd_mobilenetv1_model).

Для роботи Single Shot Multibox Detector на основі MobileNet V1 використовує faceLandmark68Net та faceRecognitionModel де зберігається інформація про кількість, орієнтовне положення опорних точок.

Однак найважливішою частиною цієї моделі є те, що модель ідентифікації облич була навчена на наборі даних WIDERFACE, а ваги надані yeephycho [27]. Комбінація цих факторів робить цю модель дуже точною в своїх розрахунках в порівнянні облич. Через що цю модель було використано як підтвердження розпізнавання обличчя на зображенні після того як Tiny Face Detector знайшов на зображенні людину [25].

Метод `faceapi.detectSingleFace()` за замовчуванням використовує саме Single Shot Multibox Detector на основі MobileNet V1. Завдяки цьому методу модель розпізнає тільки одне обличчя на зображенні і програма ідентифікації збереже данні положенні опорних точок в кеші.

3.3.2 Tiny Face Detector

Tiny Face Detector - це дуже продуктивний детектор облич в реальному часі, який набагато швидше, компактніший і менш ресурсоємний в порівнянні з Single Shot Multibox Detector детектором облич Mobilenet V1, натомість він працює трохи гірше при виявленні невеликих облич. Розмір квантованої моделі складає всього 190 КБ (`tiny_face_detector_model`).

В цій роботі ці модель буде використана для розпізнавання облич у відеопотоці з відеокамери, тим самим збільшивши швидкість розпізнавання облич.

Детектор облич був навчений на спеціальному наборі даних, що складається з ~14К зображень, позначених обмежувальними рамками. Крім того, модель була навчена передбачати обмежувальні рамки, які повністю покривають риси обличчя, таким чином, вона в цілому дає кращі результати в поєднанні з подальшим виявленням орієнтирів обличчя, ніж SSD Mobilenet V1. Через що цю модель було використано для розпізнавання облич на зображенні.

Метод `faceapi.detectAllFaces()` отримує як значення налаштування моделі, яку треба використати для розпізнавання облич з відеопотоку. В цьомій роботі для

розпізнання облич з відеопотоку використовується налаштування моделі `Tiny Face Detector` тому передається `faceapi.TinyFaceDetectorOptions()`.

`faceapi.TinyFaceDetectorOptions()` може отримати значення:

- Мінімальний поріг довіри
- Розмір зображення, що оброблюються

Розмір зображення, що оброблюються вимірюється в пікселях, та має бути кратним 32, значенням за замовчуванням є 416, але також поширені розміри 128, 160, 224, 320, 416, 512, 608, для стеження за обличчям через веб-камеру рекомендується використовувати менші розміри, наприклад, 128, 160, для виявлення більш дрібних облич рекомендуються більші розміри, наприклад, 512, 608.

3.3.3 FaceLandmark68Net

Цей пакет реалізує дуже легкий і швидкий, але точний 68-точковий детектор міток обличчя. Стандартна модель має розмір лише 350kb (`face_landmark_68_model`), а мініатюрна модель - лише 80kb (`face_landmark_68_tiny_model`). Обидві моделі використовують ідеї згорток, що розділяються за глибиною, а також щільно з'єднаних блоків. Моделі були навчені на наборі даних з ~35 тис. зображень облич, позначених 68 точками-орієнтирами.

Функція `extract_face_landmarks` виявляє обличчя на заданому зображенні, а потім повертає опорні точки обличчя (також відомі як форма обличчя) для першого знайденого обличчя на зображенні на основі коду виявлення опорних точок обличчя `dlib` [25].

Цей детектор облич побудований з використанням класичної гістограми орієнтованих градієнтів у поєднанні з лінійним класифікатором, пірамідою зображень та схемою виявлення ковзного вікна. Оцінювач пози був створений за допомогою реалізації `dlib` був навчений на наборі даних орієнтирів обличчя `iBUG 300-W` [28].

Після виявлення обличчя `faceapi.detectAllFaces()` необхідно передбачити орієнтири обличчя для кожного виявленого обличчя викликавши після цього метод `withFaceLandmarks()` розробить маску опорних точок.

3.3.4 FaceRecognitionModel

Для більш якісної ідентифікації обличчя використана архітектура, подібна до ResNet-34, для обчислення дескриптора обличчя (вектор ознак зі 128 значеннями) з будь-якого зображення обличчя, який використовується для опису характеристик опорних точок обличчя людини. Особливість моделі є те, що вона не обмежується набором обличчя, що використовуються для навчання, тобто можете використовувати її для розпізнавання обличчя будь-якої людини, наприклад, себе. Визначити схожість двох довільних обличчя можна шляхом порівняння їх дескрипторів, наприклад, шляхом обчислення евклідової відстані.

Ця модель еквівалентна FaceRecognizerNet, що використовується в face-recognition.js, і мережі, що використовується в прикладі розпізнавання обличчя в dlib. Ваги були навчені методом давіскінгу, і модель досягає точності передбачення 99,38% на тесті розпізнавання обличчя LFW (Labeled Faces in the Wild).

Розмір квантованої моделі становить приблизно 6,2 МБ (face_recognition_model).

Після розробки маски опорних точок методами faceapi.detectAllFaces().withFaceLandmarks() необхідно розрахувати дескриптори для усіх опорних точок, викликавши метод withFaceDescriptors. Він розрахує дескриптори для кожної опорної точки маски опорних точок.

Розрахунок евклідової відстані реалізується комбінацією методів faceapi.FaceMatcher() та findBestMatch(), в FaceMatcher заноситься значення дескрипторів з зображення відеопотоку, в findBestMatch заноситься список значень по імені особи з дескрипторами 14-ти зображень цієї особи. Після чого порівнюється зі значеннями дескрипторів з зображення відеопотоку. У відповіді маємо найменшу евклідову відстань між зображеннями обличчя, що є найбільшою схожістю. Варто зазначити, що в метод FaceMatcher() можна занести максимальну евклідову відстань, стандартне значення 0,6 і чим менше евклідова відстань, тим точніше можна стверджувати, що обличчя з відеопотоку належить людині з бази обличчя.

3.4 Основні методи і параметри програми

До важливих параметрів програми можна віднести наступні:

- частота обробки зображення на предмет обличчя
- максимальна евклідова відстань

Частота обробки зображення впливає в першу чергу на продуктивність програми та вимірюється в мілісекундах, і відповідає за частоту пошуку на зображенні облич, програма яка використовується в офісі робить скріншоти кожні 3 секунди, або кожні 2000 мілісекунд, якщо казати про програмні значення, однак для тестування програми було обрано швидшу частоту обробки зображення, для швидшого отримання результату по аналізу можливостей програми, а саме 200, (див. додаток Б, додаток В).

```

        faceapi.draw.drawDetections(canvasTwo, resizedDetectionsTwo)
        faceapi.draw.drawFaceLandmarks(canvasTwo, resizedDetectionsTwo)
    }, 200)
})

```

Частота обробки зображення вказується в кінці виконавчої функції

```
setInterval(async () => {
```

В якій прописується команди для роботи програми розпізнавання облич, таких як виконання пошуку на зображенні опорних точок обличчя, побудова лицевої сітки, тощо.

Параметр максимальної евклідової відстані є частиною метода `faceapi.FaceMatcher`. Цей метод відповідає за порівняння двох зображень і підрахування максимальної евклідової відстані між зображенням з відеопотоку, та зображеннями в базі зображень, та виглядає наступним чином:

```
const maxDescriptorDistance = 0.6
```

```
faceapi.FaceMatcher(labeledFaceDescriptors, maxDescriptorDistance), де
```

`labeledFaceDescriptors` – метод бібліотеки в якому зберігаються формули розрахунку дескрипторів для кожної з опорних точок, по 128 на кожному (див. додаток Б, додаток В).

maxDescriptorDistance - максимальна евклідова відстань, чим менше відстань тим точніше можна казати, що на конкретному зображенні є людина, яка є тією самою людиною, що і в базі зображень.

3.5 Інструкція з використання програми

Результат роботи програми можна побачити, запустивши index.html в режимі Live Server. Як зображено на рис. 3.3.

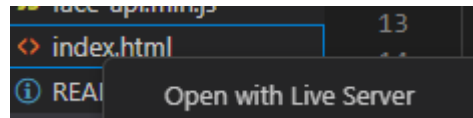


Рисунок 3.3 Запуск index.html в режимі Live Server

В файлі index.html (див. додаток Г), є 3 основні частини, які пов'язують між собою script.js, відеопотік з відеокамери та глобальну бібліотеку методів для розпізнавання та ідентифікації face-api.min.js в якій знаходиться мінімально-необхідна кількість методів для роботи з моделями.

```
<script defer src="face-api.min.js"></script>
```

```
<script defer src="script.js"></script>
```

Також тут необхідно зазначити відображення камери та місцезнаходження отриманого відеопотоку на екрані, налаштування місцезнаходження отриманого відеопотоку знаходиться в body:

```
body {
  margin: 0;
  padding: 0;
  width: 100vw;
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
}
```

Canvas – невидиме полотно того ж розміру, що і зображення відеопотоку.

script.js відображає на полотно коробку:

```
canvas {
    position: absolute;
}
```

На рис. 3.4 видно два вікна. Для кращого розуміння програми в одному вікні браузера було запущено ідентифікацію облич – номери 1, 2, а в іншому тільки розпізнавання облич – номери 3, 4. Варто зазначити, що ідентифікація облич – етап який відбувається після розпізнавання обличчя, розпізнавання обличчя використовує моделі `tinyFaceDetector`, `faceLandmark68Net` для швидкої і достатньо точної побудови маски опорних точок на зображенні. Для ідентифікації обличчя використовується модель `faceRecognitionModel` яка розраховує 128 дескрипторів для кожної опорної точки і після цього `faceapi.FaceMatcher` розраховує евклідову відстань для кожної пари дескрипторів опорних точок кожного зображення в базі з дескрипторами опорних точок для зображення з відеопотоку.

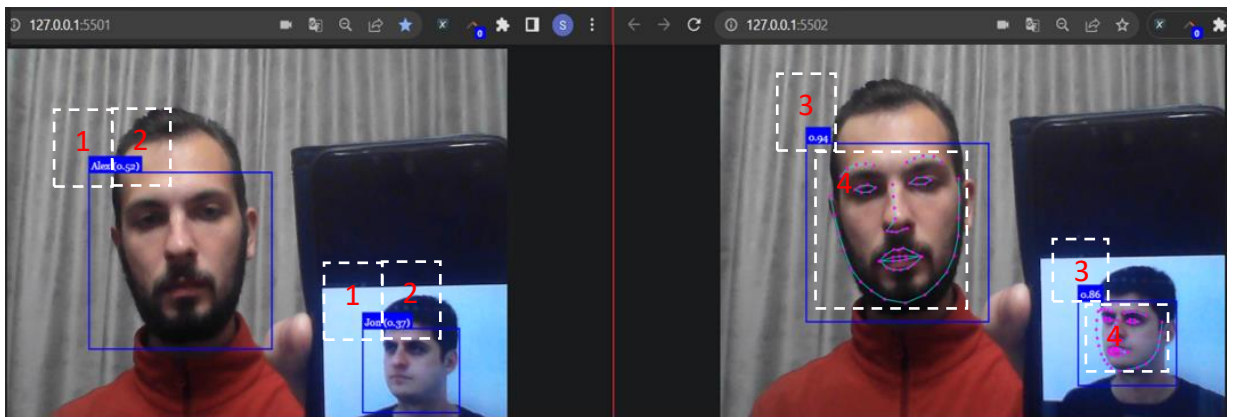


Рисунок 3.4 - Відображення відеопотоку з роботою програми ідентифікації працівників та розпізнавання облич

Варто зазначити, що як видно на рис. 3.4 для програми нема різниці між обличчям живої людини та обличчям на телефоні чи іншому предметі, цієї проблеми можна позбутися, якщо інтегрувати додаткову програму `Liveness Detector` [29] яка аналізує серію зображень, які йдуть одне за одним і тим самим може відрізнити живе обличчя, від фото на телефоні чи іншому носії.

Програми ідентифікації людини за обличчям чудово підходить для систем обліку робочого часу, в основі через те, що усі інші методи не можуть розпізнати людину поки вона сидить і працює за комп'ютером через те, що усі інші методи потребують взаємодії з контролером, в цьому ж методі в цьому нема потреби, але існує необхідність в відкритому обличчі. Відкрите обличчя – загальна норма, за винятком пандемії, коли в офісі працівники сидять в масках.

На рис. 3.5 зображений розрахунок евклідової відстані між опорними точками потенційно однакових людей, де:

array – лист значень для опорної точки на зображенні 1 (див. додаток Д),

arraytwo – лист значень для опорної точки на зображенні 2 (див. додаток Е),

arraytree – лист значень, який зберігає в собі результат виконання $(a_i - b_i)^2$,

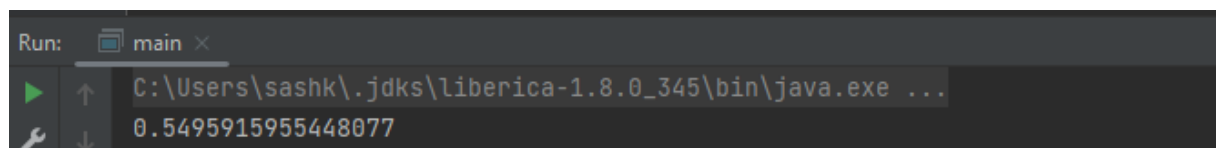
result – сумує усі значення arraytree і враховує корінь числа.

На приклад розрахунок евклідової відстані для однієї опорної точки в Java можна побачити в додатку Ж.

Додаткову інформацію про структуру програми і її файлів можна побачити в додаток И.

Висновок - можна бути впевненим в результаті тільки якщо значення дескрипторів для обох фото робила одна й та сама програма, інакше є ризик порівняння різних опорних точок.

В результаті розрахунку евклідової відстані між двома дескрипторами різних зображень наступний результат (див. додаток Ж):



```
Run: main x
C:\Users\sashk\.jdk\liberica-1.8.0_345\bin\java.exe ...
0.5495915955448077
```

Рисунок 3.5 - Результат розрахунку евклідової відстані між двома дескрипторами різних зображень

3.6 Висновки за розділом

В цьому розділі було наведено можливості face-api.js [25], його структуру і взаємодію між бібліотеками.

На базі face-api.js було розроблено програму ідентифікації людини з використанням зображень з відеокамери.

Також було наведено алгоритм роботи програми, інструкцію по використанню та описані методи.

Було розглянуто приклад вихідного зображення після роботи програми ідентифікації.

4. ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ БІОМЕТРІЇ ОБЛИЧЧЯ ЗА МЕТОДОМ POINT DISTRIBUTION MODEL

4.1 Чинники, вплив яких досліджується

4.1.1 Освітленість

Сучасна формула розрахунку освітленості передбачає використання даних про величину світлового потоку лампочки, що вимірюється люменами (Лм).

$$\text{Норма освітлення} * \text{Площа} * \text{Коефіцієнт висоти стелі} = \text{Світловий потік.} \quad (4.1)$$

З 2019 року діють державні правила природного та штучного освітлення які прописані в таблиці 4.1, де зазначені вимоги до використання всіх освітлювальних приладів, в т.ч. світлодіодних в різних типах комерційних приміщень. У ДБН [30] вказані конкретні показники освітленості зазначені в таблиці 4.1:

Таблиця 4.1 - Норма освітленості в люменах для різних приміщеннях

| Приміщення | Норма, Лм/м ² |
|---|--------------------------|
| Вітальня, спальня (підлога); кухня (стіл) | 150 |
| Дитяча кімната (підлога) | 200 |
| Кабінет, бібліотека (стіл) | 300 |
| Коридор, ванна кімната (підлога) | 50 |
| Офіс (стіл) | 300 |
| Класна кімната школи, аудиторія (стіл) | 500 |

Оскільки в даній роботі розглядається система обліку робочого часу в офісі, то значення норми освітленості для тестування програми розпізнавання облич відповідає значенню норми в таблиці 4.1 як для Офіс (стіл) і відповідає 300 Лм/м²

Далі для розрахунку світлового потоку необхідно розрахувати площу офісу в якому проводяться тестові запуски програми: довжина кімнати 5м, ширина 3м, звідки отримуємо площу офісу для тестувань:

$$3 * 5 = 15\text{м}^2$$

Останнім необхідним значенням є коефіцієнт висоти стелі, з ДБН В.2.5-28:2018 [30] отримуємо наступну таблицю:

Таблиця 4.2 - Коефіцієнт висоти стелі становить

| | |
|-------------------------------|-----|
| при висоті від 2,5 м до 2,7 м | 1 |
| від 2,7 до 3 м | 1,2 |
| від 3 до 3,5 м | 1,5 |
| від 3,5 до 4,5 м | 2 |

Оскільки в офісі для тестувань висота стелі 2,7, тоді коефіцієнт висоти стелі відповідно до таблиці 4.2 має бути 1,2.

Отримавши необхідні значення можна визначити світловий потік використавши формулу (4.1):

$$\frac{\text{Лм}}{\text{м}^2} * \text{м}^2 * \text{конст} = \text{Лм}$$

$$300 * 15 * 1,2 = 5400\text{Лм}$$

Отримавши значення світлового потоку в офісі для тестувань нам необхідно розрахувати необхідну кількість ламп, які будуть відповідати отриманій нормі світового потоку.

Інформація про світловий потік кожної окремої лампочки можна отримати в документації, або інструкції до кожної лампочки, яка йде разом з нею в коробці, або на сайті виробника. В офісі для тестувань використовуються лампочки марки Gauss, моделі 102202210, документація цієї лампочки наведена нижче:

Gauss 102202210

- Лампа світлодіодна
- Розміри: 10,6 / 6 см.
- Цоколь: E27

- Потужність: 10 Вт
- Спосіб живлення: від мережі 220В
- Світловий потік: 860 Лм
- Температура кольору: 4100К
- Кут розсіювання: 360 градусів
- Термін служби: 35000 годин

З документації маємо, що світловий потік однієї лампочки 860 Лм, для того щоб отримати значення необхідної кількості лампочок цієї моделі потрібно розділити значення загального світлового потоку для приміщення на значення світлового потоку однієї лампочки:

$$5400/860=6,2 \text{ лампи}$$

В майбутньому значення загального світлового потоку в приміщенні і кількість лампочок, яка для цього потрібна, буде сприйматись за 100% Освітленості приміщення.

Для того, аби протестувати, як освітленість приміщення впливає на імовірність наявності облич та Евклідову відстань було додано додаткове джерело світла:

RT-908

- Світловий потік: 1080 Лм

З допомогою RT-908 можна збільшити значення освітленості приміщення за межі норми, для розрахунку на скільки відсотків збільшиться освітленість приміщення порівняно з нормою використаємо формулу (4.2):

$$\left(\frac{(b-a)}{a}\right) * 100 = p \quad , \quad (4.2)$$

де, p – відсотки, a – первинне значення, b = кінцеве значення.

Отже необхідно підставити власні значення та провести розрахунки:

$$\left(\frac{((5400 + 1080) - 5400)}{5400}\right) * 100 = 20\%$$

Отже, при використанні RT-908, як доповнення до стандартної норми освітлення отримуємо 120% освітленості приміщення, отримані значення будуть використані в тестуванні програми.

Тепер створюємо таблицю тестових значень на базі отриманих:

Таблиця 4.3 – Значення освітленості для тестувань

| Освітлення (у відсотках до ДБН) | 120% | 100% | 50% | 16,6% |
|------------------------------------|------|------|-----|-------|
| | | | | |

16,6% це відповідно одна працююча лампочка освітлення, без жодних додаткових джерел світла.

4.1.2 Розмір зображення, що оброблюються

Як вже було згадано розмір зображення, що оброблюються внутрішній параметр моделі Tiny Face Detector, за його допомогою можна налаштувати обробку зображення, умовно кажучи – для моделі важлива відстань до орієнтовного обличчя. Це працює наступним чином:

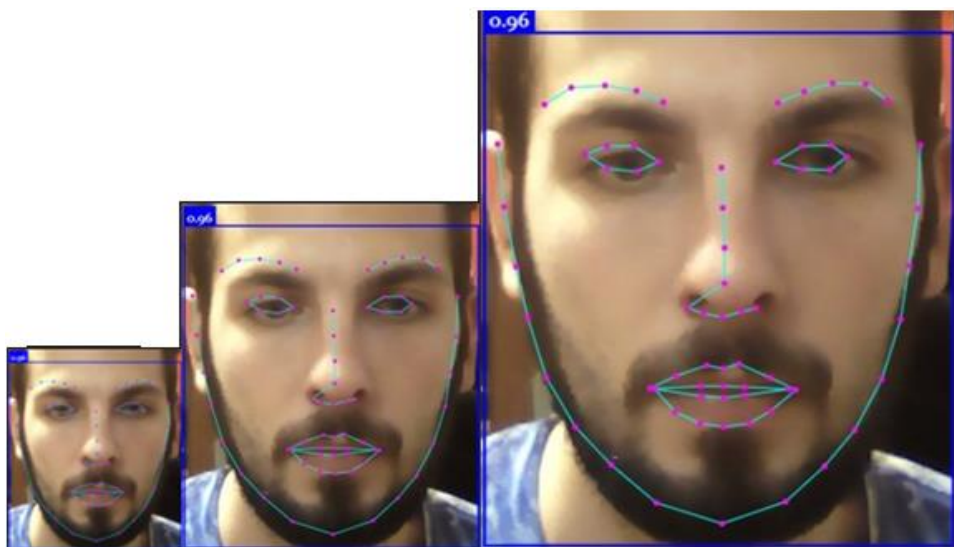


Рисунок 4.1 - Опрацьоване зображення обличчя яке завантажується в кеш програми ідентифікації

На рисунку 4.1 зображено 3 різні розмірності зображення з точки зору програми, відповідно 128, 224, 416 піксельні зображення, цю розмірність можна

вказати на етапі розпізнавання обличчя з відеопотоку камери, у методі `faceapi.TinyFaceDetectorOptions()`.

Важливо розуміти, що кожна розмірність краще підходить для одних задач і погано для інших наприклад чим менше тим швидше опрацьовується зображення, адже розмір зображення менший, але менш точно при виявленні дрібних облич, для стеження за обличчям через веб-камеру краще використовувати менші розміри, наприклад, 128, 160, для виявлення більш дрібних облич використовуйте більші розміри, наприклад, 512, 608.

Оскільки в системі обліку робочого часу існує два окремих типи камер: настінні і встановлені на комп'ютер, то необхідно протестувати якість розпізнавання облич на різних відстанях. Як вже було зазначено, для різних відстаней рекомендується використовувати різні розмірності, тому розмір зображення, що оброблюються необхідно занести в параметри для тестування у наступному вигляді:

Таблиця 4.4 - Значення розміру зображення для тестувань

| | | | | |
|---|-----|-----|-----|-----|
| Розмір зображення, що оброблюються, пікселі | 608 | 416 | 224 | 128 |
|---|-----|-----|-----|-----|

4.1.3 Відстань до камери

Як було зазначено вище в системі обліку робочого часу існує два окремих типи камер: настінні і встановлені на комп'ютер, це означає що відстані між працівником і камерою будуть завжди різні і необхідно дізнатись як буде працювати програма ідентифікації на різних відстанях до працівника.

Для тестування було обрано наступні значення, занесені в таблицю:

Таблиця 4.5 - Значення відстані до камери для тестувань

| | | | | |
|------------------------|----|----|----|-----|
| Відстань до камери, см | 20 | 40 | 80 | 160 |
|------------------------|----|----|----|-----|

Де 40 – звичайна відстань до монітору, рекомендована міністерством охорони здоров'я,

відповідно 20 – екстремально-близька відстань до монітору, через що це може мати критичний вплив на програму ідентифікації,

80 – перехідна відстань між камерою комп'ютера та настінною камерою

160 – відстань настінної відеокамери, на такій відстані настінна відеокамера має розпізнати людину, без прихованих опорних частин обличчя.

4.1.4 Вихідні данні для дослідження

Опорні чинники які можуть впливати на роботу програми ідентифікації – це:

- Освітленість приміщення;
- Відстані до камери;
- Розмір зображення, що оброблюються.

Але нажаль, це не усі фактори які впливають на роботу програму ідентифікації. Кількість зображень в базі даних певної особи має не менший вплив на розрахунок евклідової відстані між дескрипторами зображення.

Тому потрібно проаналізувати вплив додаткових чинників, а саме:

- Кількість зображень працівника, яка ураховується в базі облич;
- Кут повороту голови відносно прямого погляду в камеру;
- Вплив прихованих точок обличчя на роботу програми.

Результати тестувань будуть розглянуті в наступних розділах.

При цьому кількість працівників в базі даних буде сталою і дорівнювати 20.

Для отримання результату для кожної таблиці було зроблено по 3 тести і в таблицю з трьох результатів записано середнє арифметичне.

4.2 Дослідження впливу освітленості приміщення та відстані до камери

За допомогою таблиці 4.6 буде розглянуто результати впливу освітленості приміщення на відстань до камери. Варто зазначити, що для отримання результатів розмір зображення, що оброблювався був сталим та дорівнював 416.

Частину отриманих результатів можна побачити в Додатку К.

Таблиця 4.6 - Розпізнавання особи та розпізнавання облич в залежності від освітленості приміщення та відстані до камери

| Розпізнавання особи та розпізнавання облич в залежності від освітленості приміщення та відстані до камери | | | | | | | | |
|---|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|------------------------------|--------------------|-----------------------------|
| Освітлення (у відсотках до ДБН) | 120% | | 100% | | 50% | | 16,6% | |
| Відстань до камери, см | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Відсоток розпізнавання облич | Евклідова відстань | Імовірність наявності облич |
| 20 | 0,47 | 0,99 | 0,48 | 0,94 | 0,49 | 0,8 | 0,5 | 0,5 |
| 40 | 0,46 | 0,99 | 0,47 | 0,97 | 0,48 | 0,99 | 0,5 | 0,9 |
| 80 | 0,47 | 0,94 | 0,47 | 0,91 | 0,48 | 0,89 | 0,51 | 0,82 |
| 160 | 0,48 | 0,68 | 0,49 | 0,67 | 0,49 | 0,81 | 0,5 | 0,75 |
| Кількість суб'єктів порівняння | 20 | | | | | | | |

На отриманих результатах тестування таблиці 4.6 видно кореляцію між зміною освітленості приміщення та відстанню до камери. За умови, що обличчя дивиться в камеру в усіх тестах можна побудувати графік впливу освітленості та відстані до камери на якість роботи програми на рисунку 4.2.

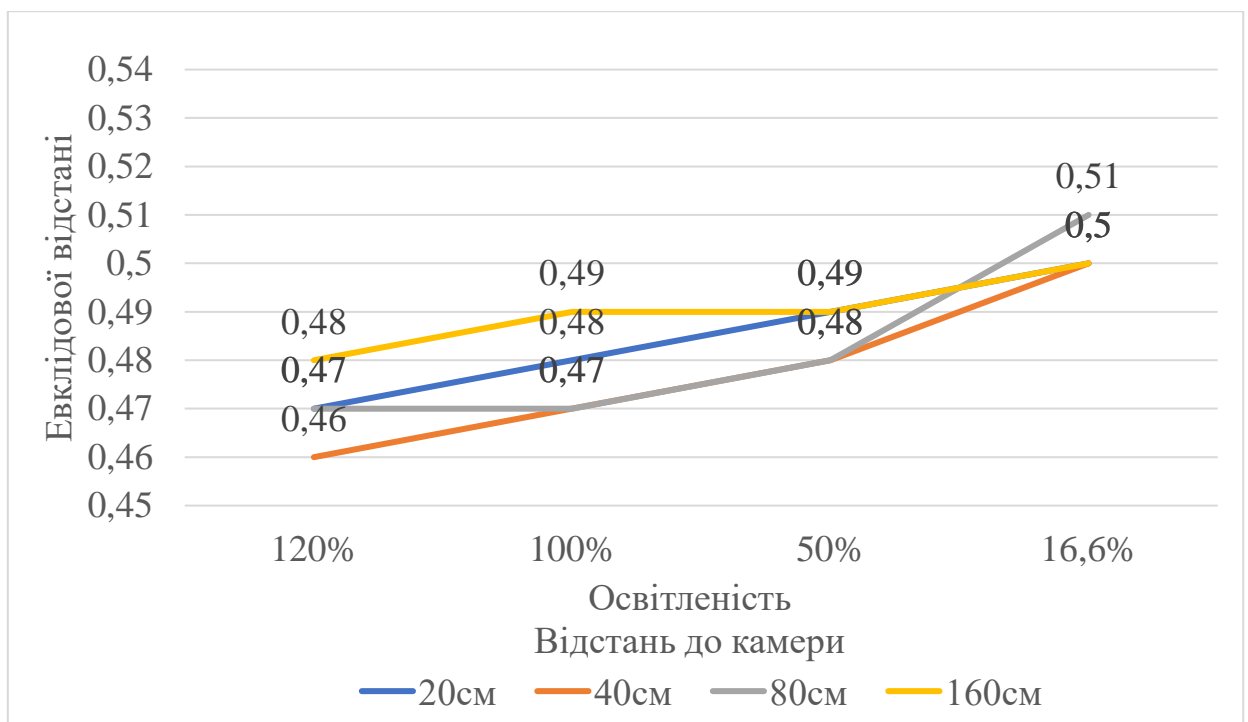


Рисунок 4.2 - Вплив на евклідову відстань освітленості приміщення та відстані до камери

З рисунку 4.2 бачимо, що вплив справді існує, якщо брати 100% освітленості приміщення та 40см відстані до камери, тобто до монітору, то:

При збільшенні відстані до камери, і освітленості 100% маємо різницю в 0,02 пікселі евклідової відстані при відстані 160см.

А при сталій відстані до камери 40см, бачимо, що посилене освітлення позитивно впливає на евклідову відстань. Виграємо 0,01 піксель відстані, але зменшення освітленості до 50% призводить до втрати того самого 0,01 пікселя. Освітленість на рівні 16,6% збільшує евклідову відстань ще на 0,02 пікселі, тобто на 0,03 якщо дивитись різницю між 100% та 16,6%.

Тобто якщо казати про вплив освітленості, або відстані до камери, де протилежний параметр дорівнює нормі - вплив освітленості вищий, ніж вплив відстані до камери.

Якщо ж дивитись на графік в цілому, чітко видно, що як освітленість, так і відстань має вплив. Якщо брати за норму значення 0,47 – 100% освітленості та 40см відстані до камери, то як зі збільшенням відстані, так і зі збільшенням освітленості додається евклідова відстань, в найгіршому випадку, а саме 80см відстані до камери та 16,6% освітленості значення евклідової відстані піднімається до 0,51 – це на 0,09 менше стелі розпізнавання.

Однак під час виконання усіх тестів програма змогла розпізнати працівника, що є гарним результатом, а значення 0,51 хоч і можна вважати негативним впливом зменшення освітленості та збільшення відстані до камери, але не створило критичних проблем для розрахунку евклідової відстані.

Далі треба перевірити вплив цих параметрів на ідентифікацію обличчя і вірогідність наявності обличчя на зображенні. Отриманий графік впливу зображений на рис. 4.3.

З рисунку 4.3 маємо, очевидно суттєвий вплив освітленості та відстані до камери.

В першу чергу треба зазначити, що найбільший вплив має критична відстань у 20см. Якщо брати по відношенню до початкового значення вплив склав 50%, імовірність наявності обличчя при такій дистанції суттєво зменшується.

Загалом усі значення впали щонайменше на 10% відносно ста на значенні в 16,6% освітленості.

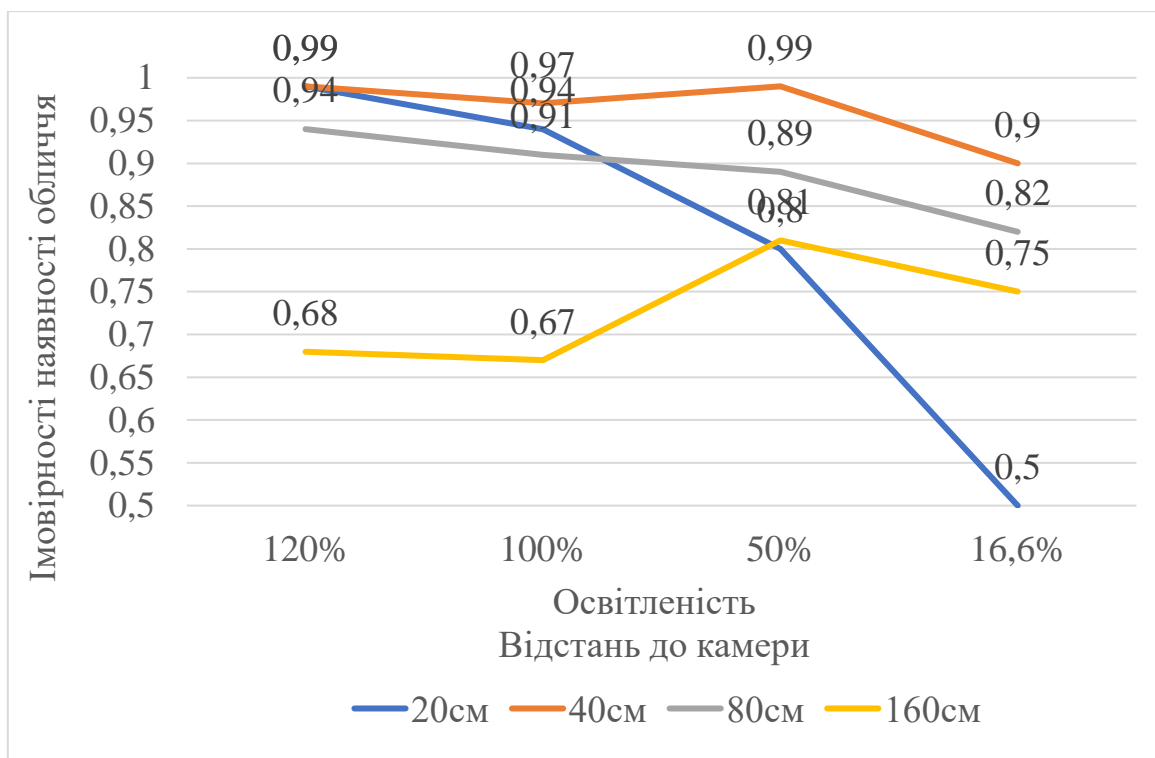


Рисунок 4.3 - Вплив на імовірність наявності облич освітленості приміщення та відстані до камери

Також на графіку чітко видно, що відстань має більший вплив на імовірність наявності обличчя. На відстані в 160см імовірність складає 0,68 при підвищеній освітленості і не сильно змінюється зі зменшенням освітленості. В той час як для іншого критичного значення 20см воно складає 0,99 і потім з кожним зменшенням освітленості падає.

4.3 Дослідження впливу освітленості приміщення та розміру зображення, що оброблюється

Далі буде розглянуто як впливає освітленість приміщення в комбінації з розміром зображення, що оброблюється програмою. Зібрані результати наведені в таблиці 4.7. Варто зазначити, що для отримання результатів відстань до камери завжди дорівнювала 40см.

Таблиця 4.7 - Розпізнавання особи та розпізнавання облич в залежності від освітленості приміщення та розміру зображення, що оброблюються

| Розпізнавання особи та розпізнавання облич в залежності від освітленості приміщення та розміру зображення, що оброблюються | | | | | | | | |
|--|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|
| Освітлення (у відсотках до ДБН) | 120% | | 100% | | 50% | | 16,6% | |
| Розмір зображення, що оброблюється, пікселі | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич |
| 608 | 0,48 | 0,99 | 0,49 | 0,96 | 0,49 | 0,84 | 0,51 | 0,85 |
| 416 | 0,46 | 0,99 | 0,47 | 0,97 | 0,48 | 0,99 | 0,45 | 0,82 |
| 224 | 0,49 | 0,92 | 0,51 | 0,89 | 0,51 | 0,89 | 0,52 | 0,87 |
| 128 | 0,5 | 0,82 | 0,52 | 0,73 | 0,5 | 0,69 | 0,54 | 0,66 |
| Кількість суб'єктів порівняння | 20 | | | | | | | |

На отриманих результатах тестування таблиці 4.7 видно кореляцію між зміною освітленості приміщення та розміру зображення. Для кращого аналізу таблиці, можна побудувати графік впливу освітленості приміщення та розміру зображення, що оброблюється на якість роботи програми, рисунок 4.4.

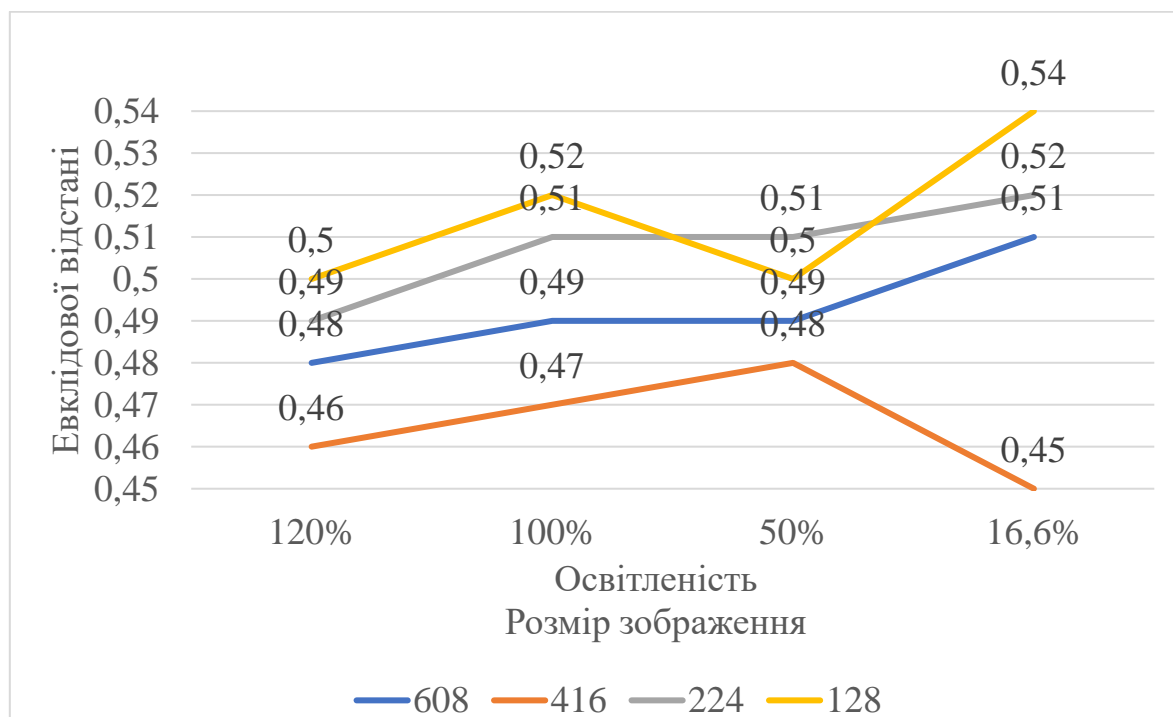


Рисунок 4.4 - Вплив на евклідову відстань освітленості приміщення та розміру зображення, що оброблюється

Як очікувалось при розмірі зображення рівному 128 програма показує стабільно поганий результат. Це пов'язано в першу чергу з тим, що при такому

розмірі як зазначалось в рис. 4.4, зображення сильно стиснуте. Програмі доводиться аналізувати сильно стиснуте зображення, де було втрачена велика кількість пікселів, які були необхідні для розрахунку дескрипторів зображення.

Відповідно 224 має не кращій результат в порівнянні з 128, результати евклідової відстані ліпші в середньому на 0,01 піксель.

Варто зазначити, що ці два розміри – це рекомендовані розміри для роботи з відеокамерою комп'ютера, вони займають менше оперативної пам'яті і за рахунок цього працюють швидше, але як видно в результаті це суттєво впливає на розрахунок евклідової відстані.

Хоча 608 є більшим значенням розміру зображення ніж 416 і мав би показувати ліпші результати, але схоже, що на відстані 40 см ця розмірність показує гірші результати, аніж 416.

416 розмірність показує стабільно нижчі значення евклідової відстані, що робить її найкращім варіантом для роботи з відеокамерою комп'ютера.

Однак окрім аналізу результатів евклідової відстані потрібно проаналізувати вплив на імовірності наявності обличчя рис. 4.8:

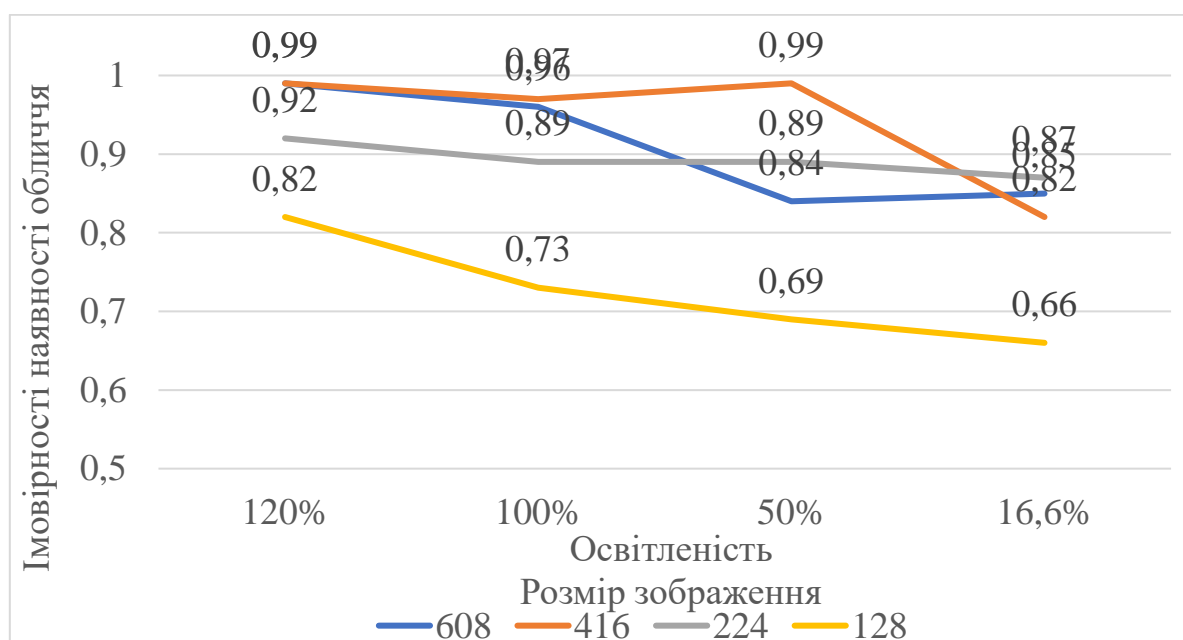


Рисунок 4.8 - Вплив на імовірність наявності обличчя освітленості приміщення та розміру зображення, що оброблюється

В середньому зі зміною освітленості імовірність падає на 20 позицій, але в випадку з 224 розміром значення середні в усіх випадках.

4.4 Дослідження впливу розміру зображення, що оброблюється та відстані до камери

Далі буде розглянуто як впливає відстань до камери з розміром зображення, що оброблюється програмою на роботу програми, зібрані результати наведені в таблиці 4.8, варто зазначити, що для отримання результатів освітленість приміщення завжди дорівнювала 100%.

Таблиця 4.8 - Розпізнавання особи та розпізнавання облич в залежності від розміру зображення, що оброблюються та відстані до камери

| Розпізнавання особи та розпізнавання облич в залежності від розміру зображення, що оброблюються та відстані до камери | | | | | | | | |
|---|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|
| Розмір зображення, що оброблюється, пікселі | 608 | | 416 | | 224 | | 128 | |
| Відстань до камери, см | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич |
| 20 | - | - | 0,51 | 0,94 | 0,51 | 0,93 | 0,5 | 0,82 |
| 40 | 0,49 | 0,96 | 0,47 | 0,97 | 0,51 | 0,89 | 0,52 | 0,73 |
| 80 | 0,47 | 0,94 | 0,49 | 0,91 | 0,51 | 0,66 | - | - |
| 160 | 0,51 | 0,91 | 0,52 | 0,67 | - | - | - | - |
| Кількість суб'єктів порівняння | 20 | | | | | | | |

На отриманих результатах тестування таблиці 4.8 видно кореляцію між зміною відстані до камери та розміру зображення. Для кращого аналізу таблиці можна побудувати графік впливу відстані до камери та розміру зображення, що оброблюється на якість роботи програми - рисунок 4.9.

На рисунку 4.9 чітко видно, що деякі розмірності на певних відстанях взагалі не працюють, тобто дана комбінація факторів при не вірному налаштуванні може призвести до фатальних проблем.

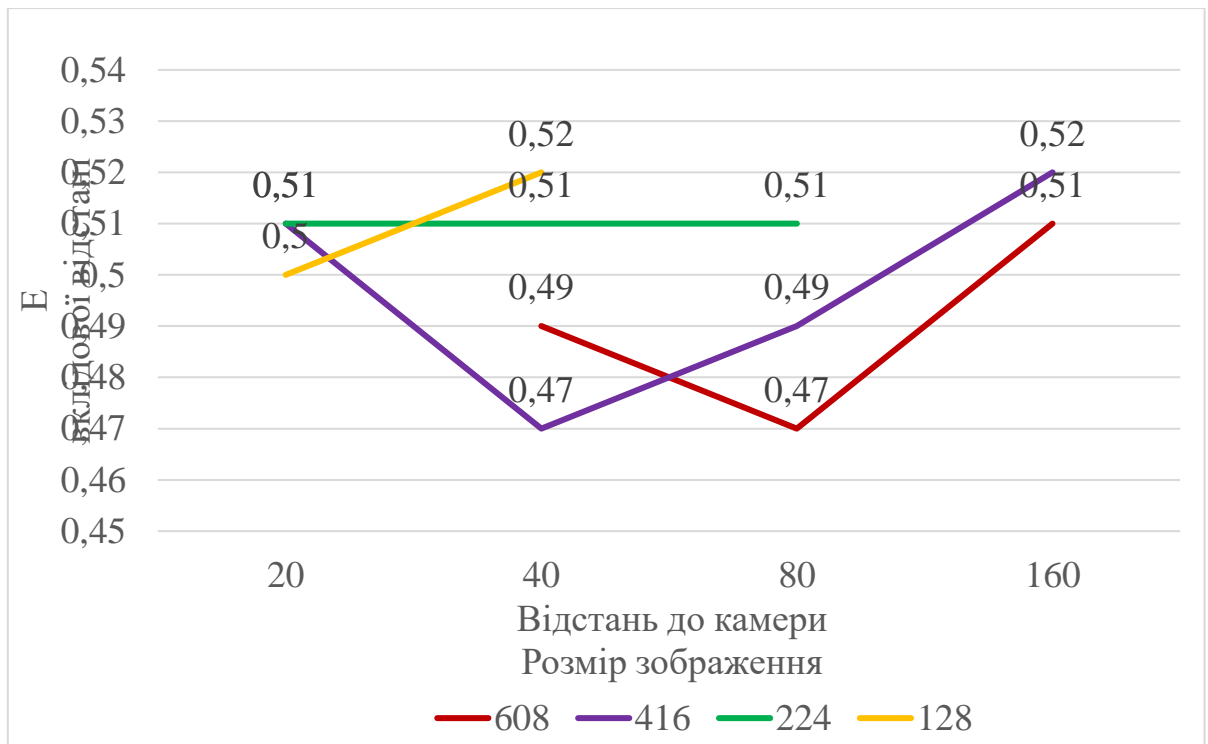


Рисунок 4.9 - Вплив на евклідову відстань розміру зображення, що оброблюється та відстані до камери

Три з чотирьох векторів не працювали на певній відстані, 416 розмірність обробки зображень є найстабільнішою відносно інших.

Розмірності 224 та 128, які вважались гарним варіантом для відеокамер комп'ютера, неочікувано, але ці значення не показують високої ефективності на малих відстанях до камери. Обидва вектори знаходяться над значенням в 0,5 пікселів евклідової відстані, що є поганим результатом в порівнянні з 416 розмірністю.

А 608 навпаки, показує свою ефективність на дальніх відстанях, як і очікувалось.

416 розмірність показує свою стабільну якісну роботу на малих та високих відстанях до камери.

На рисунку 4.10 показано вплив вище описаних факторів, на імовірність наявності обличчя.

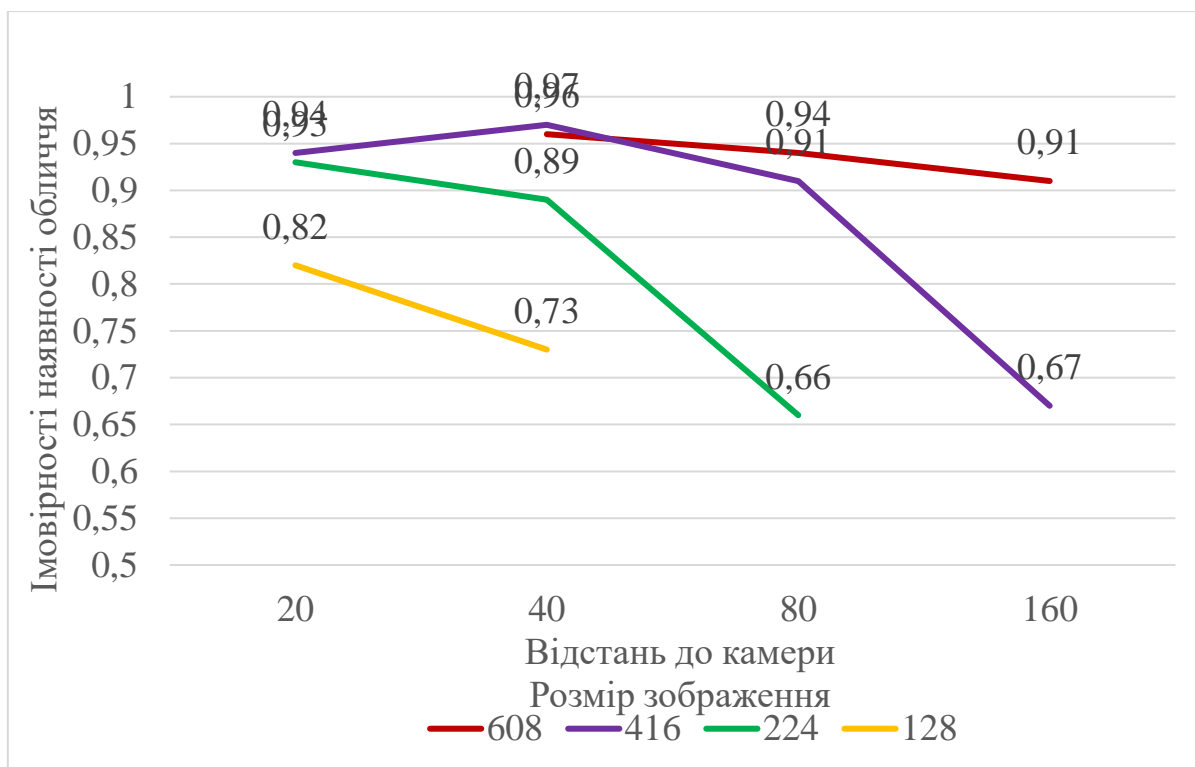


Рисунок 4.10 - Вплив на імовірність наявності обличчя розміру зображення, що оброблюється та відстані до камери

На рисунку 4.10 бачимо, що при зменшенні розміру зображення йде явна просадка імовірності наявності обличчя. Варто зазначити, що найгірше себе показує розмірність 128, а 224 виглядає досить ефективним, з урахуванням того, що ця розмірність, як і 128, рекомендовані для відеокамер комп'ютера. Але й в нього відбувається суттєва просадка на 80см від камери, згадуючи стабільно поганий результат в 0,51 евклідової відстані і з чого робиться висновок, що і ця розмірність недостатньо ефективна для програми ідентифікації.

608 і на цьому графіку показує, що це значення направлене на дальню відстань, на 160см від камери значення не просідає нижче 90, це виглядає цілком більш ніж ефективно.

416 в свою чергу показує, що він цілком задовольняє потребам комп'ютерної камери, демонструючи чудовий результат, падаючи тільки на відстані в 160.

Роблячи висновки по впливу цих трьох параметрів важливо зазначити наступне:

- Як і очікувалось усі 3 фактори мають вплив на якість роботи програми;

- Вплив «розміру зображення, обробки зображення» суттєво вищий за інші фактори, тільки цей фактор зміг викликати відсутність розпізнаного обличчя.
- Вплив «відстані до камери» досить високий, при рівності інших чинників, значення евклідової відстані можуть піднятися вище 0,5 евклідової відстані;
- Вплив «освітленості приміщенні» існує, але в порівнянні з впливом відстані до об'єкта його можна вважати не суттєвим, умовно можна вважати, що при погіршенні освітленості до рівня 16,6% значення евклідової відстані не піднімуться до 0,5, а будуть, в середньому, коливатись нижче цієї позначки.
- За винятком тесту впливу відстані до камери і розміру зображення, що оброблюється програмою на роботу програми, де обличчя навіть не було розпізнане, в усі тестах програма могла точно ідентифікувати особу, що говорить про ефективність програми.

Оскільки в цих тестах жодного разу не було проблеми з розпізнаванням особи, в наступних тестах буде розглянуто вплив відсутності частини обличчя на якість роботи програми, та вплив повороту обличчя на певний градус, при обмеженій кількості зображень в базі даних кожної людини.

4.5 Дослідження впливу обмеження кількості доступних для бази фото облич певної особи та зміни куту нахилу голови

Далі буде розглянуто як впливає кут нахилу та кількість фото облич людини в базі на якість роботи програми. Очікується, що в крайніх точках, тобто поворотах на 90° програма не зможе розпізнати особу. Зібрані результати наведені в таблиці 4.9, варто зазначити, що для отримання результатів освітленість приміщення завжди дорівнювала 100%, відстань до камери 40см, а розмір зображення, що оброблювався був сталим та дорівнював 416.

Вперше за тестування бачимо, що програма не розпізнала особу, хоча розпізнала обличчя. Варто нагадати, що особа вважається програмою розпізнаною, коли евклідова відстань між зображенням з відеопотоку та зображенням з бази даних більше 0,6.

Таблиця 4.9 - Вплив кількості фото одної людини в базі облич на результат розпізнавання

| Вплив кількості фото одної людини в базі облич на результат розпізнавання | | | | | | | | |
|---|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|--------------------|-----------------------------|
| Кількість фото в базі облич | 2 | | 6 | | 10 | | 14 | |
| | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич | Евклідова відстань | Імовірність наявності облич |
| повороти голови | | | | | | | | |
| прямо | 0,49 | 0,98 | 0,49 | 0,97 | 0,48 | 0,97 | 0,46 | 0,99 |
| вгору 30° | 0,47 | 0,68 | 0,49 | 0,93 | 0,48 | 0,7 | 0,48 | 0,94 |
| вгору 60° | 0,65 | 0,69 | 0,63 | 0,68 | 0,62 | 0,74 | 0,5 | 0,72 |
| вгору 90° | 0,68 | 0,85 | 0,67 | 0,9 | 0,63 | 0,87 | 0,59 | 0,87 |
| вниз 30° | 0,54 | 0,97 | 0,53 | 0,97 | 0,51 | 0,8 | 0,52 | 0,84 |
| вниз 60° | 0,58 | 0,92 | 0,55 | 0,54 | 0,5 | 0,86 | 0,5 | 0,77 |
| вниз 90° | - | - | - | - | - | - | - | - |
| вліво 30° | 0,45 | 0,91 | 0,47 | 0,96 | 0,46 | 0,83 | 0,46 | 0,98 |
| вліво 60° | 0,56 | 0,72 | 0,57 | 0,98 | 0,54 | 0,98 | 0,52 | 0,85 |
| вліво 90° | 0,59 | 0,77 | 0,59 | 0,55 | 0,54 | 0,96 | 0,53 | 0,75 |
| вправо 30° | 0,48 | 0,66 | 0,5 | 0,98 | 0,49 | 0,99 | 0,5 | 0,84 |
| вправо 60° | 0,51 | 0,88 | 0,51 | 0,93 | 0,53 | 0,9 | 0,53 | 0,96 |
| вправо 90° | 0,58 | 0,85 | 0,53 | 0,9 | 0,56 | 0,94 | 0,54 | 0,94 |

Нерозпізнано людину було у наступних тестах:

- Вгору 60° - 2, 6, 10;
- Вгору 90° - 2, 6, 10.

Але також було зовсім нерозпізнано обличчя при опусканні голови вниз на 90°, не вдалось ідентифікувати в жодному з тестів.

Варто розглянути графіки евклідової відстані та імовірності наявності облич для пошуку кореляцій та кращого аналізу рис. 4.11 та рис. 4.12 відповідно.

Якщо подивитись на кожен з 30°, рис. 4.11, а також поворот прямо в камеру то ці значення не виходять за 0,5 пікселі евклідової відстані.

Також, цікавим є те, що при повороті голови вліво, в середньому, програма надає ліпші значення евклідової відстані, це може бути пов'язано зі специфікою калібрування бібліотек ідентифікації.

Найгірші показники відбуваються при поворотах голови вгору та вниз, крайнє положення погляду вниз не розпізнається програмою зовсім.

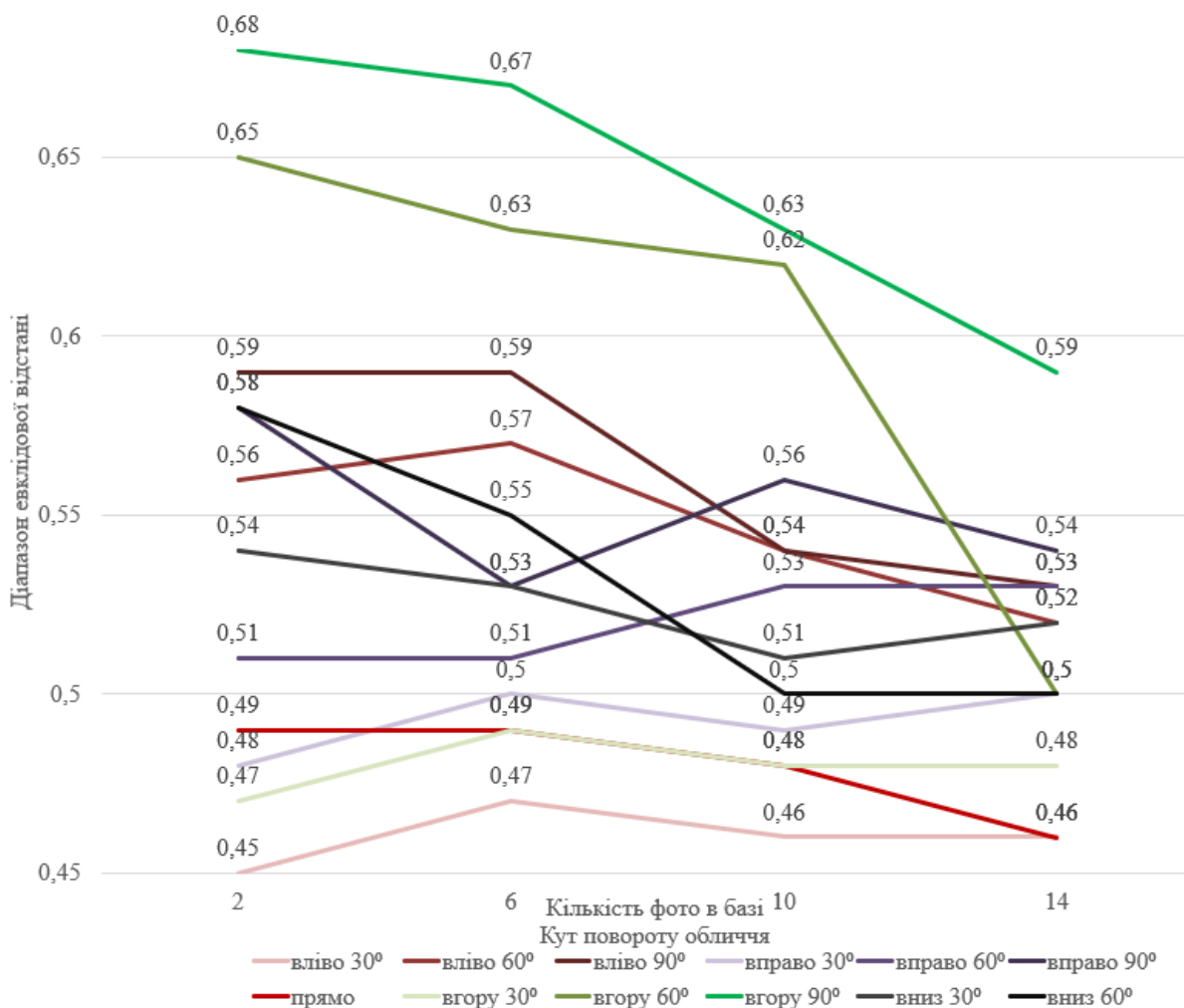


Рисунок 4.11 - Вплив на евклідову відстань кількості фото одної людини в базі облич та куту нахилу голови

Стосовно кількості зображень – на диво, навіть при 2х зображеннях в базі програма в змозі розпізнати особу в різних ракурсах, однак на графіку і таблиці чітко видно, що зі збільшенням кількості зображень в базі поліпшується і значення евклідової відстані, навіть значення піднятої голови при 14 зображеннях в базі розпізнається як конкретна особа.

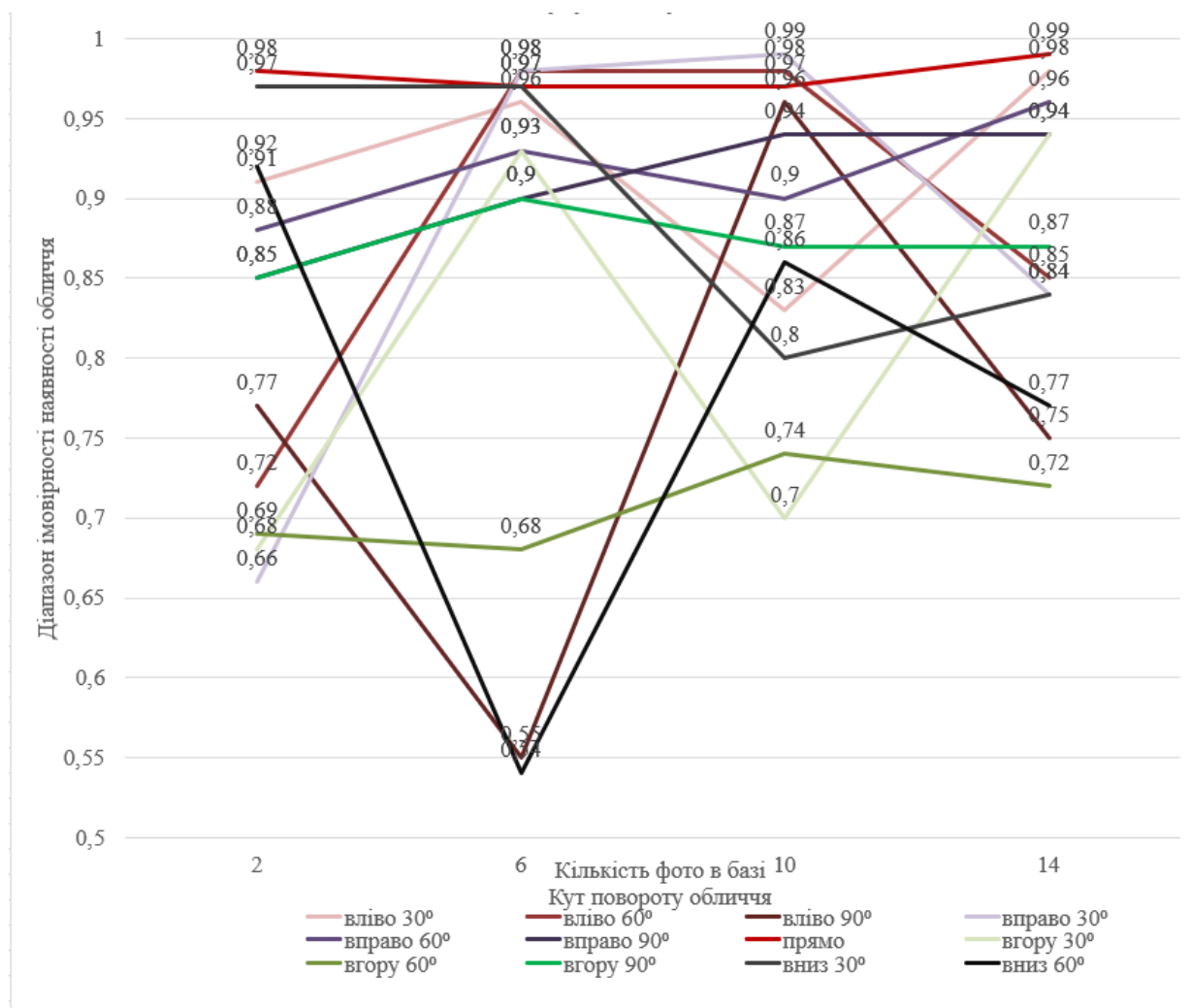


Рисунок 4.12 - Вплив на імовірність наявності обличчя кількості фото одної людини в базі обличчя та куту нахилу голови

На рисунку 4.12 бачимо, що здебільшого програма розпізнає усі положення голови, за виключенням повороту голови вниз на 90°.

Половина значень не падає впродовж тестування нижче 0,9, а саме повороти прямо, вліво 30°, вліво 60°, вниз 30°, вгору 30°, вправо 30°, вправо 60°, вправо 90°.

Однак деякі ракурси просідають більш значно, наприклад вгору 60°, вліво 90°, вгору 90°.

В більшості векторів значення імовірності не падає нижче 0,8.

Стабільно погано себе показує гору 60°.

Можна зробити висновок, що програма добре розпізнає обличчя, які повернуті вліво чи вправо, але погано вгору чи вниз. Ця тенденція може бути

пов'язана з тим, що при повороті голови вгору чи вниз приховується одна чи декілька опорних областей орієнтації.

Даний тест дає можливість стверджувати, що програма по різному оцінює наявність чи відсутність певних частин обличчя.

Для перевірки цього твердження необхідно провести тестування програми при наявності чи відсутності різних частин обличчя і різних опорних точок.

4.6 Дослідження впливу наявності та відсутності опорних точок та частин обличчя

Як вже було зазначено в попередньому розділі:

Отримані результати в розділі 4.5 дають право стверджувати, що в використаній програмі ідентифікації різні частини обличчя мають різну вагу для підтвердження особи.

Для підтвердження твердження було використано результати таблиці 4.10.

Таблиця 4.10 - Вплив кількості елементів на результат розпізнавання

| Вплив відсутніх елементами обличчя на результат розпізнавання | око-праве | око-ліве | ніс | брів-ліва, око-ліве і частина контура обличчя | В окулярах | брів-права, око-праве і частина контура обличчя | очі і брова | очі, брова, голова піднята вгору | ніс і контур обличчя | рот і контур обличчя | очі, брова, контур обличчя | рот, контур обличчя і частина носу | рот, контур обличчя і ніс |
|---|-----------|----------|------|---|------------|---|-------------|----------------------------------|----------------------|----------------------|----------------------------|------------------------------------|---------------------------|
| Кількість закритих опорних точок | 6 | 6 | 9 | 15 | 15 | 17 | 22 | 25 | 26 | 37 | 39 | 42 | 46 |
| Евклідова відстань | 0,52 | 0,55 | 0,64 | 0,51 | 0,55 | 0,54 | 0,52 | 0,5 | 0,63 | 0,53 | 0,53 | 0,58 | - |
| Імовірність наявності обличчя | 0,91 | 0,92 | 0,97 | 0,86 | 0,92 | 0,95 | 0,8 | 0,63 | 0,92 | 0,86 | 0,64 | 0,65 | - |

В таблиці 4.10 наведено результати тестування програми, відсортованої по кількості точок, завдяки чому вже видно нерівномірність між кількістю точок яка була прихована та евклідовою відстанню.

Для кращого розуміння таблиці 4.10 було створено два графіки рис. 4.13 та рис. 4.14 відповідно.

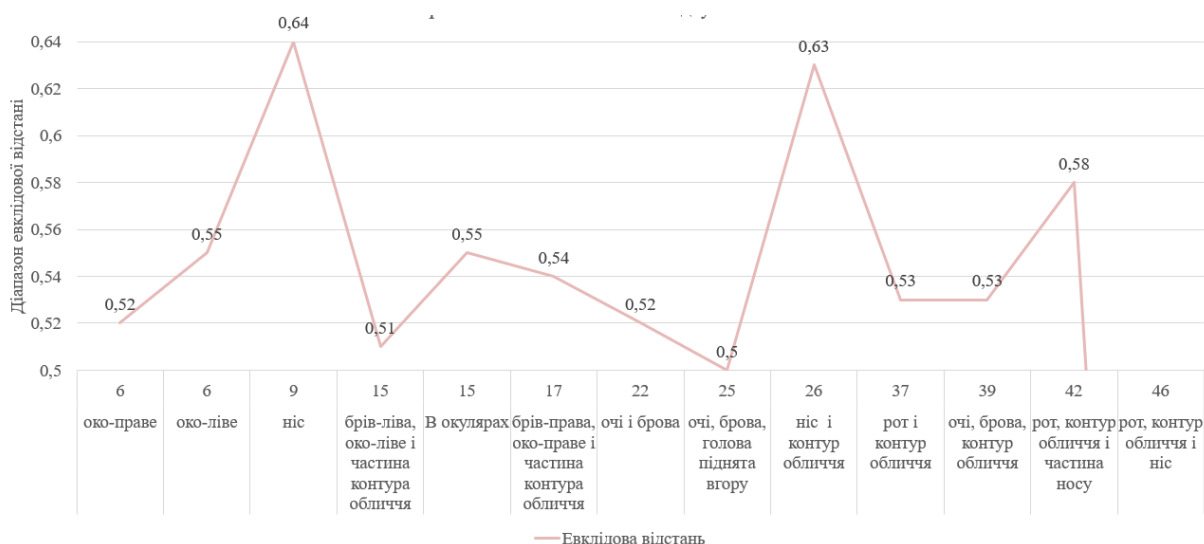


Рисунок 4.13 - Вплив прихованих елементів обличчя на величину евклідової відстані (сортування по кількості прихованих опорних точок)

На рис. 4.13 чітко видно, що піковим значенням евклідової відстані виступають тести, де був прихований ніс, в 2х з 4х випадків програма не змогла розпізнати особу, в 1му з 4х програма не розпізнала людину взагалі.

Цікавим є той факт, що при надягнутій масці, яка мала б збити значення евклідової відстані, адже контур обличчя складає 17 опорних точок, програма змогла розпізнати особу.

Якщо подивитись на усі тести, де було включено контур обличчя програма розпізнала особу, за виключенням місць, де був задіяний ніс.

В цілому чітко видно, що прихована будь-якої частини обличчя, де є опорні точки, призводить до збільшення евклідової відстані за межі 0,5

Вплив відсутності деяких частин не має високого впливу порівняно з іншими, наприклад очі, брова, контур обличчя, навіть якщо перекрити усі ці частини обличчя, а це 39 опорних точок, ефект буде менший, ніж від відсутності 9 опорних точок носу.

Вплив цих факторів на імовірність наявності обличчя був зображений на рис. 4.14:



Рисунок 4.14 - Вплив прихованих елементів обличчя на величину імовірності наявності обличчя (сортування по кількості прихованих опорних точок)

На відміну від рис. 4.13 вплив носу на імовірність наявності обличчя не настільки велика, але більше помітно вплив збільшення кількості прихованих опорних точок.

До значення в 22 приховані опорні точки, приклад на рис. 4.15, програма не надає імовірність наявності обличчя нижчу 0,85, хоча 26 і 37 демонструють результат не менший за 0,85, але після цього значення одразу йде падіння на 20%, після 42 прихованих опорних точок програма перестає фіксувати обличчя.

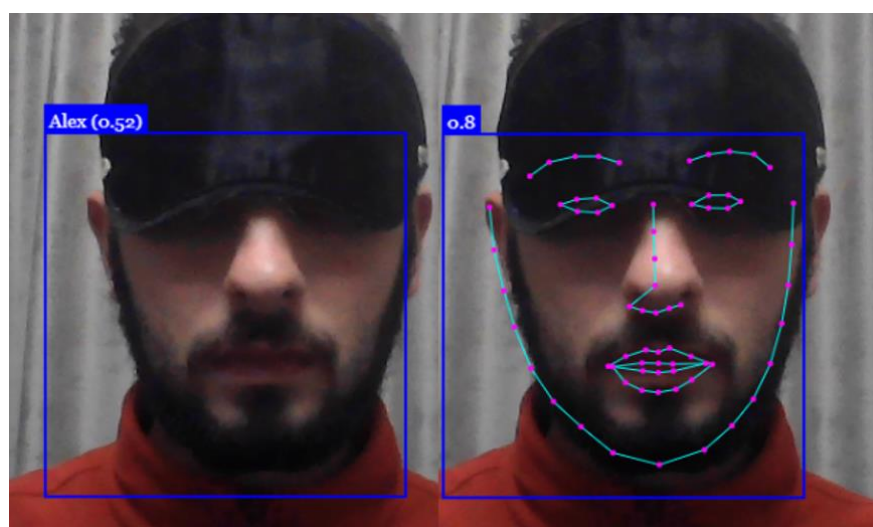


Рисунок 4.15 - Вплив прихованості 22 опорних точок на роботу програми ідентифікації

На рис. 4.16 чітко видно на скільки нерівномірний розподіл евклідової відстані відносно кількості прихованих точок.

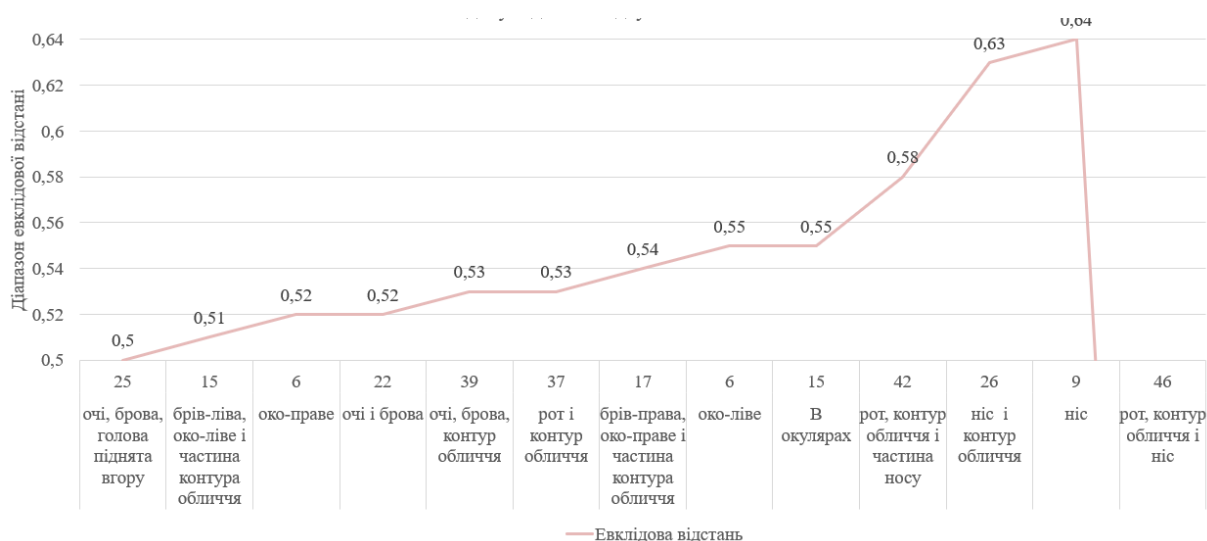


Рисунок 4.16 - Вплив прихованих елементів обличчя на величину евклідової відстані (сортування по евклідовій відстані)

На рис. 4.16 найнижчі показники евклідової відстані при і при прихованих 25 опорних точок, з чого можна зробити висновок, що навіть якщо працівник буде частково прикривати своє обличчя, програма зможе його розпізнати, але до того моменту коли людина приховає ніс.

З отриманих результатів можна зробити висновок, що ніс є опорною частиною обличчя, і хоч приховано було 9 опорних точок – програма вже не може ідентифікувати людину, що можна побачити на рис. 2.17:

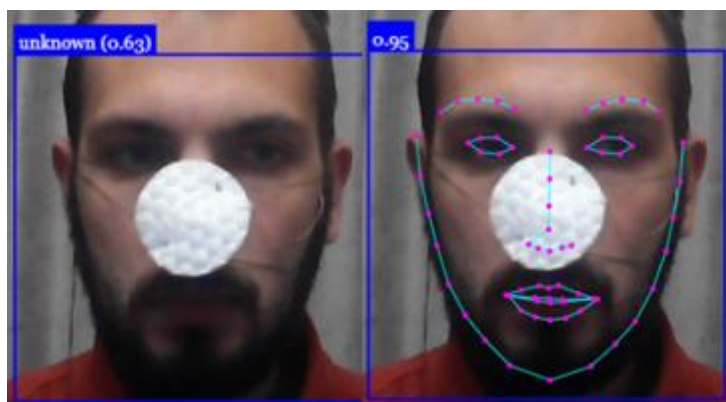


Рисунок 4.17 - Вплив прихованості 9 опорних точок носу на роботу програми ідентифікації

Однак важливо зазначити, що в усіх випадках, окрім тестів з прихованим носом програма змогла розпізнати особу перед камерою, найкращій результат – 42 приховані опорні точки обличчя і де було приховано більше 60% опорних точок, які необхідні для точного розрахунку евклідової відстані.

Розглядаючи рис. 4.18, як і очікувалось, кореляції між евклідовою відстанню та імовірністю наявності обличчя відсутня, що в свою чергу підтверджує попередні висновки: імовірність наявності обличчя залежить саме від кількості точок, які були приховані.

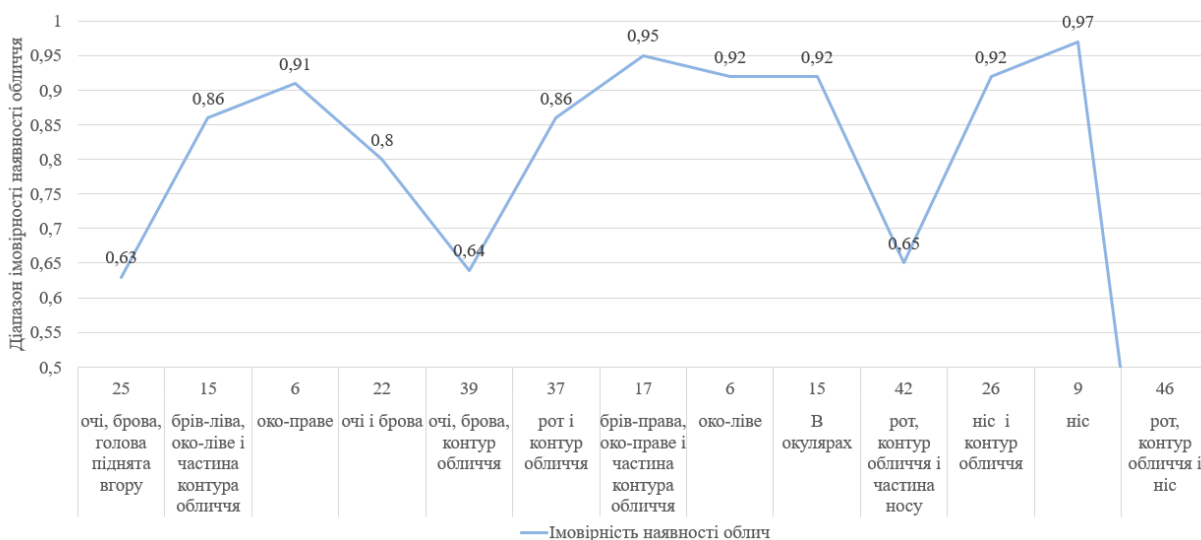


Рисунок 4.18 - Вплив прихованих елементів обличчя на величину імовірності наявності обличчя (сортування по евклідовій відстані)

На рисунку 4.18 чітко видно впадини на значеннях прихованих опорних точок більший 25, виключеннями є 26 опорних точок з прихованим носом та контуром обличчя та рот і контур обличчя.

Якщо проаналізувати вплив контура обличчя то в цьому тесті він незначний, можливо Tiny Face Detector в змозі проаналізувати місце знаходження опорних точок відносно носа чи інших частин обличчя.

З отриманих результатів можна зробити висновок, що на імовірність наявності обличчя кількість опорних точок впливає більше, ніж частини обличчя, які були приховані. В той час як на розпізнавання і розрахунок

евклідової відстані набагато більше впливає яку саме частину обличчя було приховано.

Але більш низькі значення імовірності наявності обличчя не погіршують сам факт ідентифікації особи.

4.7 Аналіз результатів

Отримані результати з рис. 4.13-4.18 кажуть про високий вплив прихованих частин обличчя на евклідову відстань, але незначний вплив на імовірність наявності обличчя.

Відносно кількості прихованих точок, рис. 4.13, вплив на евклідову відстань має незначне значення, прихованість 6 опорних точок може не суттєво відрізнитись від прихованості 22 чи 39, але має більш суттєвий вплив на імовірність наявності обличчя, рис. 4.14, однак тести показують, що зі зменшенням імовірності наявності обличчя значення евклідової відстані ростуть незначним чином.

Імовірність наявності обличчя можна вважати показником умов які впливають на результат, навіть якщо значення імовірності 0,7 – це не означає, що людину не можна розпізнати, чому можна знайти підтвердження в усіх таблицях тестування, скоріше, що варто покращити умови зовнішнього середовища, налаштування камери, для того, аби значення наявності обличчя не падало нижче 0,5.

Зовнішнє середовище має вплив на евклідову відстань в розмірі від -0,02 при покращенні зовнішніх умов до +0,04 при погіршенні, рис. 4.2, якщо казати про відстань до камери та освітленість приміщення, докладніше в розд. 4.2.

Але розмір зображення, що оброблюється має набагато більший вплив на роботу програми ідентифікації, так ,якщо брати за норму розмір 416 – 0,47, відмінність між евклідовою відстанню при інших розмірах сягає від 0,02 до 0,03 чи повної відсутності ідентифікації.

З усіх тестів обличчя було розпізнане як “unknown” в 8 випадках, це 6 – критичні повороти обличчя:

- Вгору 60° - 2, 6, 10;

- Вгору 90° - 2, 6, 10,

та в двох тестах з прихованим носом.

В 6 тестах програма не змогла розпізнати обличчя на відеопотоці, 4 в тесті на розмір зображення, що оброблюється та відстані до камери, а також 1 – опускання обличчя вниз на 90° і 1 – приховано рот, контур обличчя і ніс або 46 опорних точок.

В усіх інших тестах програма успішно розпізнала працівника – 95 тестових комбінацій.

4.8 Рекомендації, основані на отриманих результатах

Після аналізу отриманих даних варто мати на увазі наступне:

Систем розпізнавання обличчя багато і кожна краще підходить для певних умов застосування, також вони відрізняються по складності інтеграції, що дозволяє обрати той метод, який краще підходить до умов застосування.

Розрахунок опорних точок, дескрипторів та евклідової відстані між зображеннями двох обличчя дозволяє ідентифікувати людину з високою точністю.

Реалізація системи ідентифікації людини за допомогою біометрії обличчя на JavaScript потребує використання моделей, які реалізують ідентифікацію та бібліотеки методів які можуть працювати з цими моделями.

Тестування системи вказує на те, що варто обмежити використання одягу який приховують ніс

Для камер комп'ютера варто використовувати розмір обробки зображення 416, оскільки він показав найкращий результат на відстані до камери 20-80 см.

Для настінних камер спостереження варто використати розмір обробки зображення 608, оскільки він показує кращі результати в порівнянні з 416 на відстанях від 80см до камери.

Вплив освітленості на розрахунок евклідової відстані від 0,01 до 0,03 при 16,6% освітленості за ДБН [30]. З чого можна зробити висновок, що в випадку необхідної економії, де буде необхідно вимкнути частину освітлення вплив на програму буде меншим, ніж від відстані до камери.

При встановленні камер варто врахувати той факт, що програма найкраще розпізнає обличчя які дивляться прямо, таким чином буде досягнуто кращої надійності.

4.9 Висновки за розділом

В цьому розділі було зроблено серію досліджень та проаналізовано вплив на роботу програми ідентифікації наступних чинників:

- Освітленість приміщення;
- Відстані до камери;
- Розмір зображення, що оброблюються.
- Кількість зображень працівника, яка ураховується в базі обличч;
- Кут повороту голови відносно прямого погляду в камеру;
- Вплив прихованих точок обличчя на роботу програми.

Також на базі отриманих результатів було створено список рекомендацій.

ВИСНОВКИ ТА РЕКОМЕНДАЦІЇ

В роботі на основі проведеного дослідження ефективності застосування засобів біометричної ідентифікації осіб за обличчям отримано рекомендації щодо можливостей їх використання в системі обліку робочого часу.

Розглянута структура та задачі систем обліку робочого часу та методів ідентифікації обличчя.

Описано методику ідентифікації обличчя та спосіб визначення схожості двох облич.

Здійснено вибір середовища та мови програмування для розробки програмного забезпечення та розроблено алгоритми роботи підсистеми ідентифікації особи. На основі алгоритмів розроблено програмне забезпечення.

Проведено дослідження впливу низки чинників на можливість ідентифікації особи, за результатами якого сформовано основні вимоги та рекомендації з застосування біометрії обличчя в системі обліку робочого часу.

Рекомендовано використовувати ідентифікацію облич на базі Point Distribution Model, якщо кількість наявних опорних точок більша 22 та не був закритий ніс. Для камер комп'ютера варто використовувати розмір обробки зображення 416, оскільки він показав найкращий результат на відстані до камери 20-80 см. Для настінних камер спостереження варто використати розмір обробки зображення 608, оскільки він показує кращі результати в порівнянні з 416 на відстанях від 80см до камери. При встановленні камер варто врахувати той факт, що програма найкраще розпізнає обличчя які дивляться прямо, таким чином буде досягнуто кращої надійності.

Розроблені програмні засоби можуть використовуватися для реалізації в системах обліку робочого часу.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кабицький О.М., Остапець Д. О., Дослідження можливостей використання біометрії обличчя в системі обліку робочого часу// Сучасні інформаційні та комунікаційні технології на транспорті, в промисловості і освіті: Тези XVI Міжнародної науково-практичної конференції. – Д.: ДІТ, 2022, С.146.
2. A.V. Pobiyanskaya, E.A. Kipervar, A.M. Dubrovin Systems of control and accounting of working time as a direction of increasing labour productivity, ISSN 2410-1613 Russian Journal of Labor Economics, Russia, June 2021; pp. 631–642.
3. Рекомендации: выбор и применение систем контроля и управления доступом – М.: ФГУ НИЦ «Охрана» 2011. 95с.
4. В. А. Ворона, В. А. Тихонов Системы контроля и управления доступом. Серия «Обеспечение безопасности объектов»; Выпуск 2. 272с.
5. Галанкин А.В., Чащин С.В. СТРУКТУРА СИСТЕМЫ КОНТРОЛЯ И УПРАВЛЕНИЯ ДОСТУПОМ ОРГАНИЗАЦИИ// Известия Тульского государственного университета. Технические науки. Вып. 4. Тула: Изд-во ТулГУ, 2021. С. 480-485.
6. Офіційний сайт компанії Yaware.TimeTracker [Електронний ресурс] - Режим доступу до сайту : yaware.com.ua.
7. Функціональні можливості Yaware.TimeTracker [Електронний ресурс] - Режим доступу до сайту : <https://timetracker.yaware.com.ua/uk/what-is-yaware/for-business/>.
8. Ismail, N.; Sabri, M.I.M. Review of existing algorithms for face detection and recognition. In Proceedings of the 8th WSEAS International Conference on Computational Intelligence, Man-Machine Systems and Cybernetics, Kuala Lumpur, Malaysia, 14 – 16 December 2009; pp. 30–39.
9. Cootes, T.F.; Taylor, C.J. Active shape models—‘Smart snakes’. In BMVC92; Springer: London, UK, 1992; pp. 266–275.
10. Lanitis, A.; Cootes, T.; Taylor, C. Automatic tracking, coding and reconstruction of human faces, using flexible appearance models. Electron. Lett. 1994, 30, С. 1587–1588.

11. Lanitis, A.; Hill, A.; Cootes, T.F.; Taylor, C. Locating Facial Features Using Genetic Algorithms. In Proceedings of the 27 th International Conference on Digital Signal Processing, Limassol, Cyprus, 26–28 June 1995; pp. 520–525.
12. Ertugrul, Ö.F.; Tekin, R.; Kaya, Y. Randomized feed-forward artificial neural networks in estimating short-term power load of a small house: A case study. In Proceedings of the 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, 16–17 September 2017; pp. 1 – 5.
13. Chaturvedi, S.; Titre, R.N.; Sondhiya, N. Review of handwritten pattern recognition of digits and special characters using feed forward neural network and Izhikevich neural model. In Proceedings of the 2014 International Conference on Electronic Systems, Signal Processing and Computing Technologies, Nagpur, India, 9–11 January 2014; pp. 425–428.
14. Saikia, T.; Sarma, K.K. Multilevel-DWT based image de-noising using feed forward artificial neural network. In Proceedings of the 2014 International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 20–21 February 2014; pp. 791–794.
15. Sirovich, L.; Kirby, M. Low-dimensional procedure for the characterization of human faces. *Josa A* 1987, 4, 519–524.
16. Kirby, M.; Sirovich, L. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.* 1990, 12, 103–108.
17. Islam, M.R.; Azam, M.S.; Ahmed, S. Speaker identification system using PCA & eigenface. In Proceedings of the 2009 12th International Conference on Computers and Information Technology, Dhaka, Bangladesh, 21–23 December 2009; pp. 261–266.
18. Face Recognition Using Eigenfaces, Matthew A. Turk and Alex P. Pentland, MIT Vision and Modeling Lab, CVPR '91.
19. Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven.

20. Andrea Maricela Plaza Cordero, Jorge Luis Zambrano Martínez, " Estudio y Selección de las Técnicas SIFT, SURF y ASIFT de Reconocimiento de Imágenes para el Diseño de un Prototipo en Dispositivos Móviles" , 15° Concurso de Trabajos Estudiantiles, EST 2012.

21. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008.

22. Leo Liberti, Carlile Lavor, Euclidean Distance Geometry, Springer Cham, 2017

23. Baker, Loren (November 24, 2004). "Mozilla Firefox Internet Browser Market Share Gains to 7.4%". Search Engine Journal. Archived from the original on May 7, 2021. Retrieved May 8, 2021.

24. Lardinois, Frederic, "Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows". TechCrunch. Retrieved April 15, 2018.

25. JavaScript API for face detection and face recognition in the browser implemented on top of the tensorflow.js core API [Електронний ресурс] - Режим доступу до сайту : <https://justadudewhohacks.github.io/face-api.js>.

26. TensorFlow.js is an open-source hardware-accelerated JavaScript library for training and deploying machine learning models [Електронний ресурс] - Режим доступу до сайту : <https://github.com/tensorflow/tfjs>.

27. Tensorflow Face Detection mobilenet SSD(single shot multibox detector) based face detector with pretrained model provided, Yeephycho [Електронний ресурс] - Режим доступу до сайту : <https://github.com/yeephycho/tensorflow-face-detection>

28. One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan, CVPR, 2014.

29. Adrian Rosebrock, Liveness Detection with OpenCV, 2019 [Електронний ресурс] - Режим доступу до сайту : <https://www.pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>.

30. Державні Будівничі Норми України В.2.5-28:2018, Мінрегіон України, Київ, 2018.

ДОДАТОК А**Тези доповіді на міжнародній науково-практичній конференції**

ДОДАТОК Б
Вихідний код програми script.js

ДОДАТОК В
Вихідний код програми script.js

ДОДАТОК Г**Файл локального сервера index.html**

ДОДАТОК Д**Таблиця значень дескриптора для однієї опорної точки зображення в базі**

ДОДАТОК Е**Таблиця значень дескриптора для однієї опорної точки зображення з камери**

ДОДАТОК Ж

**Приклад програми розрахунку евклідової відстані між дескрипторами
двома зображеннями**

ДОДАТОК И

Файлова структура програми в Visual Studio Code

ДОДАТОК К**Результати дослідження впливу освітленості приміщення та відстані до камери**