

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Комп'ютерних наук і технологій

(повне найменування факультету)

Комп'ютерні системи та мережі

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістра

(ступінь вищої освіти)

на тему ІНТЕЛЕКТУАЛЬНА СИСТЕМА ІДЕНТИФІКАЦІЇ ПОРУШЕНЬ
ВИКОРИСТАННЯ ЗАСОБІВ ІНДИВІДУАЛЬНОГО ЗАХИСТУ

Виконав: студент 2 курсу, групи КНТ-522м

Спеціальності 123 Комп'ютерна інженерія

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні системи та мережі

РИБАЧЕНКО Н.А.

(ПРИЗВИЩЕ та ініціали)

Керівник МР

ТЯГУНОВА М.Ю.

(ПРИЗВИЩЕ та ініціали)

Рецензент

СТЕПАНЕНКО О.О.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій

Кафедра «Комп'ютерні системи та мережі»

Ступінь вищої освіти магістерський

Спеціальність 123 Комп'ютерна інженерія

(код і найменування)

Освітня програма (спеціалізація) «Комп'ютерні системи та мережі»

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Зав. кафедри Кудерметов Р.К.


 “ ” 2023 року

**З А В Д А Н Н Я
 НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА**

РИБАЧЕНКО Наталії Анатоліївни

1. Тема проєкту (роботи) Інтелектуальна система ідентифікації порушень використання засобів індивідуального захисту

керівник проєкту (роботи) ТЯГУНОВА Марія Юріївна

(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом вищого навчального закладу від “ 24 ” жовтня 2023 року №400

2. Строк подання студентом проєкту (роботи) 10 грудня 2023 року

3. Вихідні дані до проєкту (роботи) наявні методи та алгоритми для виявлення облич та розпізнавання засобів індивідуального захисту (захисні окуляри і робітничка каска)

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) Аналіз технічного завдання, дослідження наявних методів виявлення особи з відеопотоку в режимі реального часу;

2) Розробка структури та алгоритму роботи системи;

3) Реалізація та дослідження системи;

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

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	Тягунова М.Ю., доцент		
нормоконтроль	Польська О.В., ст.викл.		

7. Дата видачі завдання 01.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз існуючих методів та систем розпізнавання образів, виділення недоліків та переваг	10.10.2023 р.	
2	Аналіз існуючих систем ідентифікації ЗІЗ	15.10.2023 р.	
3	Розробка алгоритму роботи системи розпізнавання ЗІЗ	20.10.2023 р.	
4	Проектування інтелектуальної системи розпізнавання ЗІЗ	05.10.2023 р.	
5	Реалізувати систему.	20.11.2023 р.	
6	Провести тестування системи	25.11.2023 р.	
7	Оформлення отриманих результатів у ПЗ	01.12.2023 р.	
8	Оформлення допоміжного матеріалу	10.12.2023 р.	

Студент


(підпис)**Наталія РИБАЧЕНКО**

(ім'я, ПРИЗВИЩЕ)

Керівник проєкту (роботи)


(підпис)**Марія ТЯГУНОВА**

(ім'я, ПРИЗВИЩЕ)

РЕФЕРАТ

ПЗ: 70 с., 41 рис., 2 таблиці, 28 джерел, 1 додаток.

КОМП'ЮТЕРНИЙ ЗІР, РОЗПІЗНАВАННЯ ОБРАЗІВ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ, БЕЗПЕКА НА ПІДПРИЄМСТВІ, ВИКОРИСТАННЯ ЗІЗ.

Мета роботи – підвищити рівень безпеки робітників на підприємствах за рахунок розробки системи ідентифікації порушень використання засобів індивідуального захисту.

В результаті проведених експериментів відповідно до поставлених завдань було проведено дослідження різних алгоритмів виявлення осіб та розпізнавання об'єктів. Було проведено аналіз актуальності проєкту, аналіз найбільш поширеного набору даних зображенні осіб у захисних окулярах та захисних шоломах (касках), досліджені та вивчені різні ефективні методи та підходи, що дозволяють визначити правильність одержуваних результатів, а також була спроектована система розпізнавання осіб та виявлення захисних окулярів та каски через відеопотік.

Об'єктом дослідження є методи комп'ютерного зору для виявлення облич з відеопотоку, та використання нейронних мереж для розпізнавання на обличчі засобів індивідуального захисту (захисні окуляри, робітничка каска), далі ідентифікація особи, яка порушила правила безпеки на підприємстві.

ABSTRACT

Explanatory note to the master's work: 70 p., 41 figures, 2 tables, 28 sources, 1 appendix.

COMPUTER VISION, PATTERN RECOGNITION, CONVERSION NEURAL NETWORKS, MACHINE LEARNING, SECURITY IN THE ENTERPRISE, USE OF PPE.

The purpose of the work is to increase the level of safety of workers at enterprises by developing a system for identifying violations of the use of personal protective equipment.

As a result of the conducted experiments, in accordance with the tasks, a study of various algorithms for identifying persons and recognizing objects was conducted. An analysis of the relevance of the project was carried out, an analysis of the most common data set of images of persons in protective glasses and protective helmets was carried out, various effective methods and approaches that allow determining the correctness of the obtained results were researched and studied, and a system of face recognition and detection of protective glasses was designed and helmets through the video stream.

The object of the research is computer vision methods for detecting faces from a video stream, and the use of neural networks for facial recognition of personal protective equipment (safety glasses, hard hats), and identification of a person who violated safety rules at the enterprise.

ЗМІСТ

Вступ	8
1 Аналіз систем і методів розпізнавання	10
1.1 Машинне навчання для ідентифікації облич	10
1.2 Згорткові нейронні мережі.....	12
1.3 Системи розпізнавання облич	13
1.4 Методи виявлення осіб.....	14
1.4.1 Методи виявлення обличчя.....	16
1.4.2 Методи попередньої обробки зображень	23
1.5 Аналіз існуючих систем ідентифікації	25
2 Проєктування системи	30
2.1 Опис та призначення системи	30
2.1.1 Підсистема виявлення робітника	31
2.1.2 Підсистема виявлення окулярів	31
2.1.3 Підсистема розпізнавання особи порушника.....	31
2.1.4 Підсистема виявлення шолому.....	33
2.1.5 Головний модуль керування.....	33
2.2 Функціонування системи	34
3 Реалізація системи	35
3.1 Вимоги до апаратного та програмного забезпечення	35
3.1.1 Налаштування апаратного забезпечення.....	38
3.1.2 Бібліотеки	44
3.1.3 Алгоритм роботи системи ідентифікації обличчя.....	46
3.1.4 Датасети	47
3.2 Навчання НМ.....	52
3.3 Архітектура проєкту	54
4 Результати роботи.....	58
4.1 Результати роботи з датасетом	59
4.2 Результати роботи системи на прикладах	61
Висновки.....	65
Додаток А	69

ВСТУП

У промисловій галузі зараз найактуальнішим питанням є підвищення рівня безпеки на підприємстві та зниження кількості нещасних випадків.

Використання засобів індивідуального захисту (ЗІЗ) - це ефективний захід, спрямований на підвищення рівня безпеки працівника. Однією зі складових безпечної праці є використання працівниками засобів індивідуального захисту. Нехтування цими засобами під час виконання робіт може призвести до важких травмувань та навіть смертельних наслідків [1].

Працівники часто нехтують такими ЗІЗ [2], як окуляри та захисний шолом (каска) [3, 4]. Для підвищення безпеки охорони праці на підприємстві пропонується використання системи розпізнавання порушень, яка ідентифікує, якщо працівник не використовує ЗІЗ. Безпека завжди є на першому місці при організації трудового процесу. Тому обрана тема магістерської роботи є актуальною.

Мета роботи – підвищити рівень безпеки робітників на підприємствах за рахунок розробки системи ідентифікації порушень використання засобів індивідуального захисту.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести аналіз існуючих методів та систем розпізнавання образів;
- провести аналіз існуючих систем ідентифікації засобів індивідуального захисту та визначити вимоги до розроблюваної системи;
- спроектувати інтелектуальну систему ідентифікації порушень використання засобів індивідуального захисту;
- реалізувати спроектовану систему;
- провести тестування розробленої системи.

У розробленій системі розпізнавання відбуватиметься з відеопотоку у режимі реального часу. Кадр з порушенням техніки безпеки з відеопотоку буде надсилатися на сервер для збереження як доказу. Подія фіксується у журналі порушників. Далі

автоматично вмикається гучномовець та повідомляє, що правила безпеки порушені з вимогою виправити їх.

Запропонована система ідентифікації порушень надає підприємству низку переваг:

- уникнути суперечок з порушником щодо наявності порушення, пред'явивши докази з відеопотоку;
- відслідковуючи кількість порушень, можна застосовувати систему штрафів для зниження їх кількості надалі;
- підвищити безпеку працівників на підприємстві використовуючи постійний моніторинг на предмет використання ЗІЗ [5].

1 АНАЛІЗ СИСТЕМ І МЕТОДІВ РОЗПІЗНАВАННЯ

Магістерська робота заснована на використанні методів машинного навчання та комп'ютерного зору з використанням штучних нейронних мереж. Застосування такої технології використовується на підприємствах, банках для розпізнавання осіб для та надання їм доступу. Наприклад на підприємствах встановлюють електронні турнікети з ідентифікацією відбитків пальців або зі зчитувачами карток, та завдяки бурхливому розвитку біометричних технологій, підприємства все частіше переходять на більш ефективні, більш зручні та точні методи розпізнавання [6]. Одними з таких є розпізнавання осіб у відеопотоці в режимі реального часу. Більш конкретна сфера застосування даної технології — виявлення людей [7], які не дотримуються заходів безпеки на підприємстві з особливо небезпечними умовами праці, порушення використання засобів індивідуального захисту, для фіксації даних порушень, та через систему штрафів зниження кількості нещасних випадків на виробництві.

Актуальність роботи, якщо масштабувати дану систему, полягає в тому, що результати дослідження дозволяють вдосконалити процес контролю доступу чи допуску на певну територію або громадське місце таких як: метро, аеропорт, університет, для виявлення порушників техніки безпеки.

1.1 Машинне навчання для ідентифікації облич

Для розробки системи ідентифікації ЗІЗ на обличчі робітника методами машинного навчання, необхідно зібрати масив даних з різними моделями захисних окулярів і робітничих шоломів, який буде використаний для напіваавтоматичної побудови моделі. Зібравши достатній набір даних [8], необхідно завантажити їх у різні алгоритми машинного навчання. Результатом буде модель, яка може

прогнозувати новий результат, отримуючи нові дані. Тестуючи систему можна використовувати різні «регулятори», щоб «підкрутити» алгоритм навчання та отримати різні моделі. Модель, яка видає найкращий результат, буде використана для реалізації системи.

Розробка рішення за допомогою машинного навчання показана на рис.1.1.

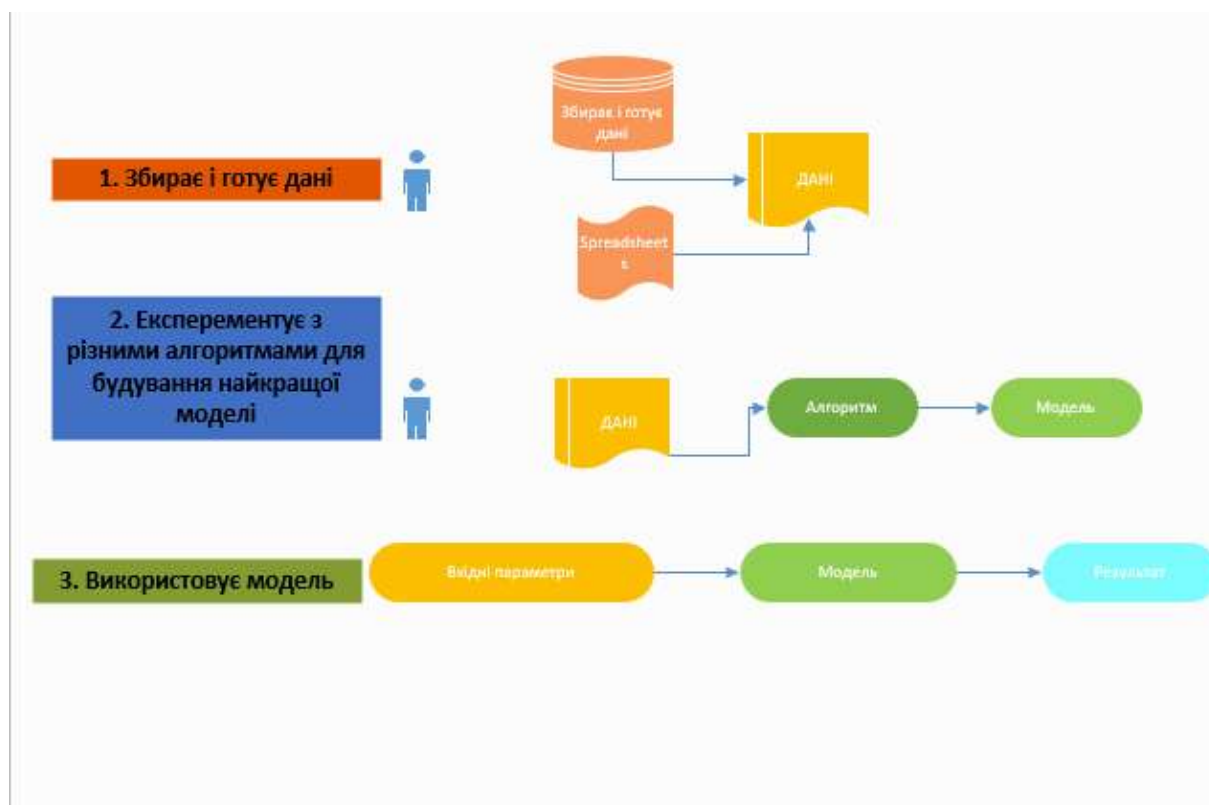


Рисунок 1.1 – Розробка моделі за допомогою машинного навчання

Переваги машинного навчання в тому, що нам не потрібно будувати модель самостійно, цю задачу виконують алгоритми машинного навчання. Щоб реалізувати систему ідентифікації ЗІЗ, необхідно опрацювати щонайменше 200 тисяч вхідних параметрів, які вплинуть на результат. Людина апріорі не зможе побудувати алгоритм, який буде використовувати всі ці параметри розумним чином. Для машинного навчання таких обмежень не існує. Головне мати достатньо потужний процесор і багато пам'яті, тоді можна реалізувати задачу будь якої складності.

Традиційно машинне навчання поділяють на навчання з вчителем (supervised), навчання без вчителя (unsupervised) та навчання з підкріпленням (reinforcement learning) [9].

Навчання з вчителем на сьогодні є найбільш розвиненим і застосовним різновидом машинного навчання. Щоб реалізувати його на практиці, нам необхідне завдання, яке може бути сформульоване у вигляді проблеми класифікаційного чи регресійного аналізу, а також достатній набір маркованих даних. Зараз існують десятки готових класичних алгоритмів машинного, а також різні алгоритми глибинного навчання (Deep Learning) для вирішення складніших завдань, таких як обробка зображення, тексту та голосу (рис.1.2).



Рисунок 1.2 – Алгоритм навчання

1.2 Згорткові нейронні мережі

Не меншою популярністю користується ще один вид машинного навчання - згорткова нейронна мережа (також CNN або ConvNet) при якому модель вчиться виконувати завдання класифікації безпосередньо на зображенні, відео, текст або звук. Згорткові нейронні мережі особливо корисні для пошуку шаблонів (патернів) на зображеннях для розпізнавання об'єктів, облич та сцен. Вони вчаться безпосередньо

на зображеннях, використовуючи шаблони для класифікації зображень та усуваючи потребу в ручному вилученні ознак.

1.3 Системи розпізнавання облич

Розпізнавання облич - це програмний алгоритм, що використовується для перевірки чи ідентифікації обличчя людини шляхом обробки відеокادру або цифрового зображення, на якому видно обличчя людини. Технології розпізнавання обличч використовують декілька різних методів, але зазвичай вони порівнюють риси особи на зображенні з особами, що містяться в базі даних.

Процес розпізнавання обличч – це набір різних завдань, які служать для ідентифікації особи за цифровим зображенням або відео. У загальному вигляді цей процес відбувається наступним чином: після того, як система отримала зображення з камери, першим кроком у системі є виявлення особи, у якому визначається, чи містяться будь-які обличчя на зображенні чи відео [10]. Виявлення обличчя визначається розташуванням ключових функцій, таких як очі, ніс тощо. Далі за допомогою алгоритмів визначаються границі обличчя (етап виявлення). Наступний етап, це етап розпізнавання, на якому обличчя трансформується (змінюється його яскравість, воно вирівнюється, масштабується, та ін.) і приводиться до певного виду. На заключному етапі відбувається виявлення ознак особи і безпосередньо порівняння їх із ознаками з баз даних. Цей етап порівняння називається ідентифікація чи верифікація.

Верифікація: цей процес зіставлення зразків за схемою «один до одного» («1:1»). Для визначення особистості система порівнює біометричний зразок з одним біометричним шаблоном, що зберігається в базі даних, щоб передбачити, чи вони однакові.

Ідентифікація: даний процес зіставлення за схемою багатьом» (1: N) – де система порівнює обличчя на зображенні та біометричний зразок з усіма шаблонами осіб, що зберігаються у базі даних.

На рис. 1.3 (див. нижче) зображено загальний алгоритм розпізнавання облич за зображенням або відео.



Рисунок 1.3 – Алгоритм розпізнавання облич

1.4 Методи виявлення осіб

Існує безліч алгоритмів виявлення осіб. Більшість з них доступні у вигляді статей з описом математичних моделей, інші представлені у вигляді програмної реалізації. Зображення у вигляді цифрових даних із камер передається на комп'ютер – воно обробляється за допомогою спеціального алгоритму, який визначає розташування області особи за його основним рисам (очам, роту, бровам, носу тощо). Подібних методів виявлення осіб існує багато і більшість з них є комбінацією різних методів [11].

У цьому розділі наведено класифікацію алгоритмів виявлення осіб, а також описані переваги та недоліки кожної групи алгоритмів.

Але їх можна розбити на дві категорії: методи на основі знань і методи виявлення особи за зовнішніми ознаками. Класифікацію методів виявлення наведено на рис. 1.4

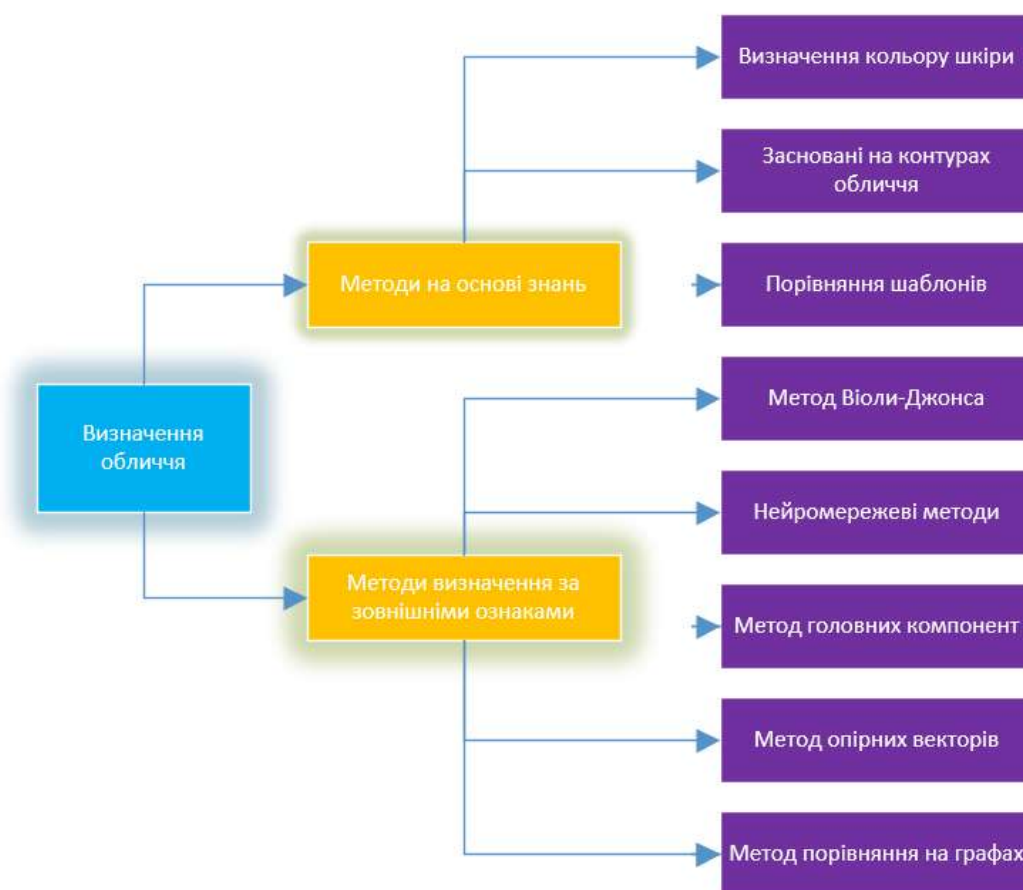


Рисунок 1.4 – Класифікація методів визначення облич

Засновані на знаннях методи залежать від набору правил, які засновані на людських знаннях виявлення осіб. Ці методи використовують інформацію про особу (особа повинна мати ніс, очі та рот на певній відстані один від одного), формі, кольорі шкіри, текстурі або її рисах. Усі правила - це формалізовані знання, якими керується людина. Наприклад, основні правила: області очей, носа та рота відрізняються за яскравістю щодо решти особи; очі на обличчі завжди розташовуються симетрично щодо один одного. Велика проблема із цими методами - складність у створенні відповідного набору правил. Там може бути багато помилкових спрацьовувань, якщо правила були надто спільними або занадто докладними. Одного цього підходу недостатньо і неможливо знайти багато обличч на кількох зображеннях. До цієї групи методів відносять більш загальний метод - метод шаблон відповідності. Метод зіставлення із шаблоном використовує попередньо певні або параметризовані

шаблони особи для визначення або виявлення осіб по кореляції між шаблонами та вхідними зображеннями. Наприклад, людське обличчя можна розділити на очі, контур обличчя, носа і рота. У даному методі описи властивостей окремих областей особи та їх заданого взаємного розташування, які допоможе за шаблонами та стандарту особи знайти відповідності з вихідним зображенням [12].

Методи на основі знань набули досить широкого поширення і мають непогані показники, однак вони показують добрі результати тільки на зображеннях з гарною роздільною здатністю, без шумів і з нескладним фоном. На кадрах з відеопотоку або камер, де можливі різні ракурси та повороти осіб, а також мінливе освітлення і велика кількість об'єктів, що заважають, на задньому плані, існує велика ймовірність виникнення помилок. Метод виявлення осіб, заснований за зовнішніми ознаками, залежить від набору зображень особи для визначення моделей особи.

Підхід, заснований на зовнішності, краще, ніж інші способи виконання. Загалом, метод, заснований на зовнішніх ознаках, ґрунтується на методах статистичного аналізу та машинного навчання, щоб знайти відповідні показники зображень обличчя. Цей метод також використовується для виділення ознак та розпізнавання осіб. Відповідно методи даної групи не мають вище зазначених недоліків і їх стали частіше застосовувати в системах відеоспостереження. Виявлення осіб у таких методах здійснюється перебором всіх фрагментів зображення визначення, до якого класу відноситься зображення: до класу зображень без обличчя або до класу що містять обличчя. Через великий обсяг роботи, методи мають надмірність і відрізняються великою обчислювальною складністю. Для того, щоб зменшити кількість обчислень та прискорити процес пошуку осіб, застосовуються додаткові методи для скорочення кількості розглянутих фрагментів [13].

1.4.1 Методи виявлення обличчя

Метод Віоли-Джоса (Viola-Jones object detection) був названий у честь двох дослідників комп'ютерного зору, які запропонували цей метод 2001 року, Паула Віоли та Майкла Джонса у своїй статті «Швидке виявлення об'єктів з використанням посиленого каскаду простих функцій». Незважаючи на те, що метод Viola-Jones

застарів, він досить потужний, та її застосування виявилось виключно помітним при виявленні осіб у реальний час. Цей алгоритм може виявляти обличчя в режимі реального часу з вражаючою швидкістю [14]. Цей метод має безліч реалізацій, у тому числі у складі бібліотеки комп'ютерного зору OpenCV.

Основні етапи роботи методу:

- використовуються зображення в інтегральному уявленні, що дозволяє обчислювати швидко необхідні об'єкти;
- використовуються ознаки Хаара, за допомогою яких відбувається пошук необхідного об'єкта (у цьому контексті, особи та її характеристик);
- використовується бустинг (від англ. boost – покращення, посилення) для вибору найбільш підходящих ознак для шуканого об'єкта наданої частини зображення;
- всі ознаки надходять на вхід класифікатора, що дає результат «правильно» чи «брехня»;
- використовуються каскади ознак для швидкого відкидання вікон, де не знайдено обличчя.

Елементи каскаду Хаара складаються з темних та світлих областей. Він виробляє одне значення, взявши суму інтенсивностей світлих областей та віднімаючи її із суми інтенсивностей темних областей. Ці ознаки Хаара дають точкове значення перепаду яскравості по осі X та Y відповідно.

Тому загальна ознака Хаара для розпізнавання облич представляє набір двох суміжних прямокутників, які лежать вище очей і на щоках. Значення ознаки обчислюється за такою формулою:

$$F = X - Y \quad (1.1)$$

де X – це сума значень яскравостей точок, що закриваються світлою частиною ознаки, а Y – це сума значень яскравостей точок, що закриваються темною частиною ознаки.

Інтегральне представлення зображень можна подати у вигляді матриці, розміри якої збігаються з розмірами вихідного зображення I , і, кожному її елементі зберігається сума інтенсивностей усіх пікселів, що знаходяться ліворуч і вище цього елемента. Елементи матриці розраховуються за такою формулою:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j) \quad (1.2)$$

де $I(i, j)$ – яскравість пікселя вихідного зображення.

Кожен елемент матриці $L(x, y)$ являє собою суму пікселів в прямокутнику від $(0, 0)$ до (x, y) , тобто. значення кожного пікселя (x, y) дорівнює сумі значень всіх пікселів ліворуч і вище від даного пікселя (x, y) .

Для того, щоб обчислити суму прямокутної області в інтегральному поданні зображення потрібно всього чотири операції звернення до масиву та три арифметичні операції. Це дозволяє швидко розраховувати ознаки Хаара для зображення у навчанні та розпізнаванні образів [15].

За такою інтегральною матрицею можна дуже швидко обчислити суму пікселів довільного прямокутника, довільної площі. Нехай у прямокутнику ABCD (рис.2.2) є цікавий для нас об'єкт D.



Рисунок 1.5 - Прямокутна область з використанням інтегрального зображення

Суму всередині прямокутника ABCD можна виразити через суми та різниці суміжних прямокутників за формулою:

$$S(ABCD) = L(A) + L(C) - L(B) - L(D) \quad (1.3)$$

Алгоритм AdaBoost (Adaptive Boosting) – це алгоритм машинного навчання для вибору кращої підмножини функцій серед усіх доступних функцій. Результатом роботи алгоритму є класифікатор (він же функція прогнозування, функція гіпотез), так званий «сильний класифікатор» [15]. Сильний класифікатор складається з лінійних комбінацій «Слабких класифікаторів» (найкращих характеристик). З високого рівня, щоб знайти ці слабкі класифікатори, алгоритм запускається для T ітерацій, де T - кількість слабких класифікаторів, які потрібні знайти. На кожній ітерації алгоритм знаходить частоту помилок для всіх функцій, а потім вибирає функцію з найнижчою частотою помилок для цієї ітерації. AdaBoost вибирає набір слабких класифікаторів для об'єднання і надає кожному з них свою вагу. Ця виважена комбінація і є сильним класифікатором (рис.1.6).

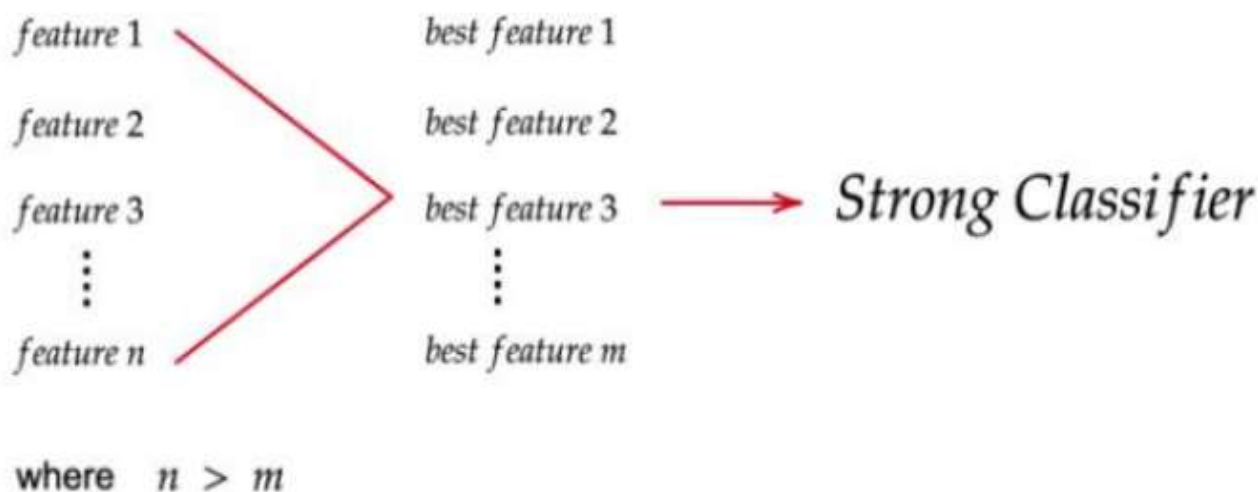


Рисунок 1.6 - Мета використання алгоритму AdaBoost

Для кожної ознаки слабкий класифікатор визначає оптимальну граничну функцію класифікації, що забезпечує мінімальне число неправильно класифікованих прикладів. Слабкий класифікатор h таким чином, складається зі значень ознаки об'єкта f , порога та паритету p , зазначенням напрямку знака нерівності.

В результаті роботи алгоритму бустингу на кожній ітерації формується простий класифікатор виду:

$$h_j(z) = \begin{cases} 1, \text{ якщо } p_j f_j(z) < p_j Q_j \\ 0, \text{ інакше} \end{cases} \quad (1.4)$$

Де, p_j показує напрямок знаку нерівності, Q_j – порогове значення j , $f_j(z)$ – обчислене значення ознаки, z – вікно зображення. Каскадний класифікатор - це багатоступінчастий класифікатор, який може виконувати виявлення швидко та точно. Кожен етап складається із сильного класифікатора, створеного алгоритмом AdaBoost. Від етапу до етапу кількість слабких класифікаторів у сильному класифікаторі збільшується [15]. Вхідні дані оцінюються послідовно (поетапно). Якщо класифікатор для певного етапу дає негативний результат, введення негайно відкидається. Якщо вихід позитивний, вхід перенаправляється на наступний етап, як наведено на рис.1.7.

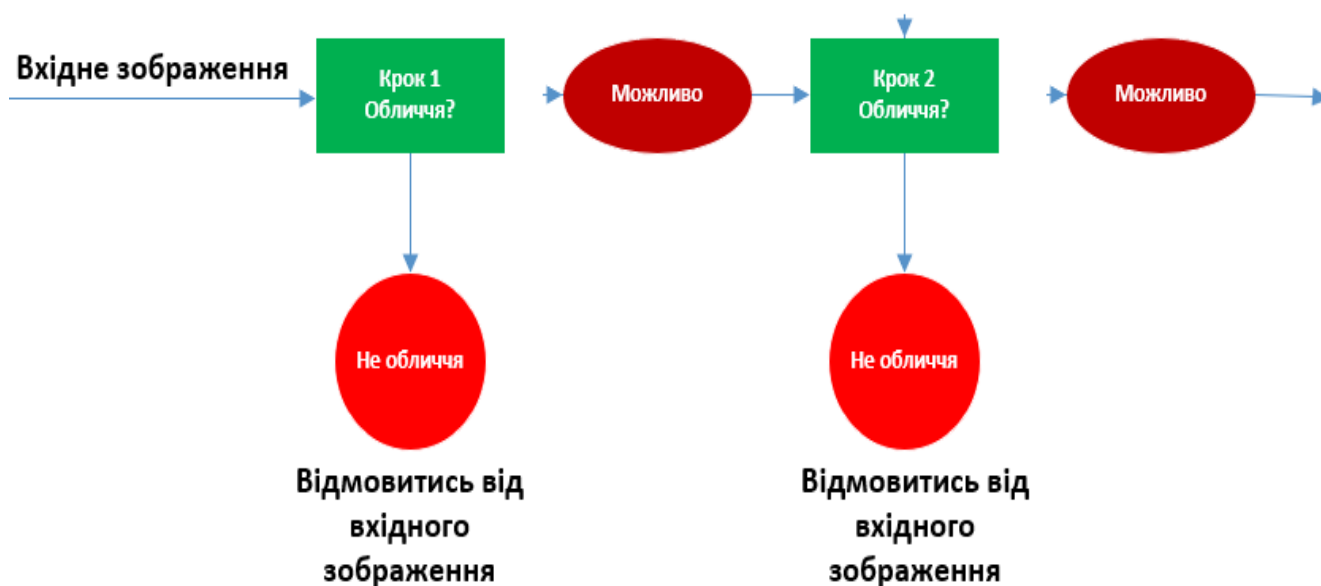


Рисунок 1.7 - Каскадний класифікатор

Коли фільтр пропускає область зображення, вона переходить до наступного фільтра у послідовності. Коли вся область зображення, пройшла через усі фільтри, потім класифікується як «обличчя», а інші класифікуються як "не обличчя".

Переваги методу Віоли-Джонса – висока точність виявлення повернутих осіб під різними кутами (до 60–65) градусів та швидкодія роботи.

Недоліки – велика кількість близько розташованих один до одного результатів через застосування різних масштабів та ковзного вікна, тривалий час навчання; алгоритму необхідно проаналізувати велику кількість тестових зображень.

Мащини опорних векторів – це лінійні класифікатори, які максимізують різницю між гіперплощиною рішення та прикладами в навчальному наборі. Суть методу полягає у зіставленні графів, які описують особи (обличчя подається у вигляді сітки з індивідуальним розташуванням вершин та ребер).

Процедура розпізнавання відбувається так — граф, характеризує параметр розпізнавання залишається без зміни, а інші деформуються під впливом структури особи з основними антропометричними точками (відстань між вухами, очима, ширина губ, лінія носа тощо) [16]. Чим більше цих точок використовується відповідно точніше буде результат розпізнавання.

Недоліки методу:

- складність алгоритму розпізнавання призводить до необхідності використання значних обчислювальних потужностей;
- складна процедура введення нових шаблонів в базу даних;
- швидкодія аналітичної системи зворотно пропорційно до розмірів баз даних.

Сховані Марківські моделі (СММ) – це набір статистичних моделей, що використовуються для характеристики статистичних властивостей сигналу та враховують їх просторові характеристики для порівняння об'єктів з шаблонами із бази [16].

Приховані Марківські моделі особливо відомі своїм застосуванням у розпізнаванні одновимірних образів, таких як розпізнавання мови, аналіз музичної

партитури та завдання визначення послідовності в біоінформатиці. Нещодавно вони були застосовані до більш складних 2D завдань, і в цьому огляді основна увага приділяється їх використанню в області автоматичного розпізнавання осіб, відстежуючи еволюцію використання СММ з початку 1990-х до наших днів [16]. Під час розпізнавання людини, відбувається перевірка всіх згенерованих Марківських моделей і шукають з можливих ймовірність того, що послідовність спостережень для об'єкта згенерована відповідна модель.

Недоліки:

- повільна швидкість роботи;
- система може оптимізувати час обробки даних та відгуку, але не може мінімізувати час перебору інших моделей;
- низька здатність розрізняти і не оптимальний алгоритм навчання.

Метод основних компонентів (РСА). Алгоритми, такі як РСА та дискримінант Фішера можуть використовуватися для визначення підпростору, що представляє шаблони обличчя. Головною метою РСА є зменшення простору ознак без значної втрати інформації і так, щоб воно якнайкраще описувало «типові» образи, що належать безлічі осіб. Існує навчений класифікатор, який ідентифікує екземпляри класу шаблонів із шаблонів фонового зображення. Алгоритми використовують для того, щоб уявити особу як вектор малої розмірності, який потім порівнюється з еталонними векторами, що представлені в базі.

Нейронні мережі успішно зіткнулися з багатьма проблемами виявлення, такими як виявлення об'єктів, виявлення осіб, виявлення емоцій, розпізнавання осіб, тощо. Довільно поширені методи, які включають близько десятка різних алгоритмів. можуть використовуватися для класифікації зображень осіб на основі таких характеристик, як відстані між деякими специфічними частинами особи (ніс, рот, очі). Також існують гібридні системи, наприклад, поєднання з марківською моделлю. У класичній НМ міжшарові нейронні сполуки пов'язані, і зображення представлене у вигляді одновимірного вектора, хоча воно двовимірне. Архітектура згорткової НМ спрямована на подолання цих недоліків [15]. У ній використовуються локальні

рецепторні поля (забезпечують локальну двовимірну зв'язність нейронів), загальні ваги (забезпечують детектування деяких рис у будь-якому місці зображення) та ієрархічна організація з просторовими підвибірками (spatial subsampling). Крім цього, будь-яка навчена НМ певною мірою може визначати належність образів до "своїх" класів, її можна спеціально навчити надійному детектуванню певних класів [16]. У цьому випадку вихідними класами будуть класи, що належать і які не належать до заданого типу образів.

Нейромережні методи досягають одні з найкращих результатів для розпізнавання, але вважаються найскладнішими для реалізації.

Переваги даного методу – висока точність виявлення при правильному налаштуванні параметрів мережі.

Недоліки:

- складна процедура внесення змін у процесі навчання (будь-яка зміна вимагає перенавчання мережі);
- складність формування архітектури мережі;
- потрібна висока обчислювальна потужність.

1.4.2 Методи попередньої обробки зображень

Для підвищення ефективності розпізнавання та якості виділення осіб, у систем розпізнавання проводять етап передобробки вхідних зображень.

Після того, як система знаходить особу в кадрі та визначає положення голови, розмір, позу та її ключові риси, вона, як правило, піддається нормалізації: кодується, масштабується, перетворюється до горизонтального положення лінії, що з'єднує центри очей. Також, при передобробці особи, застосовуються різноманітні фільтри для зниження рівня шуму (медіанний, гаусівський та ін.).

Методи попередньої обробки досить різноманітні та залежать від завдань досліджень. Найчастіше зустрічаються з них:

- нелінійні фільтри – фільтри, які використовують для видалення імпульсного шуму (окремих точок з максимальною (білою) або мінімальною

(чорною) яскравістю) на зображенні. Дії таких фільтрів полягають у визначенні позиції кожного імпульсу та заміні їх значеннями фіксованої чи випадкової величини;

- фільтр Гауса – фільтр розмиття зображення, який використовують, якщо на зображенні існують дрібні деталі, які не вимагають відділення від фону та їх можна розмити;

- медіанні фільтри – фільтри, які використовуються для збереження перепадів яскравості контурів та придушення імпульсних шумів. Медіанні фільтри досить добре працюють у таких випадках, при яких щільність шуму невелика.

Інша важлива частина систем автоматичного розпізнавання осіб – це безпосередньо самі алгоритми розпізнавання людини. На сьогоднішній день таких алгоритмів багато, і кожен із них має свою специфіку, свою швидкість роботи та свою надійність розпізнавання. Алгоритми розпізнавання поділяються на дві категорії, залежно від застосовуваної технології розпізнавання – двовірні, у яких розпізнавання відбувається за геометрією особи (2D-технології) та тривимірні, у яких розпізнавання відбувається за будовою черепа (3D-технології) [17].

Системи 2D-розпізнавання працюють із «пласкими», двовірними зображеннями і розпізнають обличчя, аналізуючи його текстуру та ділянки особи з високою контрастністю, тому при порушенні освітлення чи становища особи її розпізнати дуже складно.

Системи 3D-розпізнавання більш стійкі до таких змін, тому що при створенні моделі обличчя, у них враховуються особливості будови черепа.

Але тим не менш, хоч системи 3D і мають такий великий плюс, через відсутність можливості обслуговувати велику кількість користувачів у режимі ідентифікації та через невисоку швидкість, 3D-технології поки не набули широкого застосування. Також, 3D-технології розпізнавання потребують великих обчислювальних ресурсів, та вартість обладнання для таких систем набагато вища, ніж у 2D [17].

1.5 Аналіз існуючих систем ідентифікації

Аналогічних систем не знайдено, тому розглянемо систему, що схожа за принципом дії з розроблюваною системою ідентифікації захисних окулярів і каски на обличчі працівника.

Розглянемо систему розпізнавання облич у захисній масці, яка була розроблена через пандемію коронавірусу, щоб зменшити розповсюдження захворювання. Ситуація, коли державні установи, магазини, аптеки вимагають, щоб клієнти носили маски за визначеними правилами (наприклад, щоб маска закривала рот і ніс) при відвідуванні певних місць чи територій. Ця ситуація у всьому світі підштовхнула на дослідження автоматизованих (заснованих на комп'ютерному зорі) методів виявлення масок, які можуть допомогти контролювати громадську поведінку та запобігти поширенню пандемії COVID-19.

Проаналізуємо розроблену систему розпізнавання облич у захисній масці.

Дана система ґрунтується на використанні методів машинного навчання, та комп'ютерного зору з використанням штучних нейронних мереж. Дана технологія розроблена для виявлення людей, які не дотримуються карантинних заходів, соціальне дистанціювання та не носять захисні медичні маски, а також подальшої заборони їм доступу до приватних чи громадських місць.

Роздивимось далі, яким методом та алгоритмом була розроблена система розпізнавання захисних масок на обличчі.

MaskTheFace – це заснований на комп'ютерному зорі скрипт для маскуванню облич на зображеннях. Він використовує dlib для визначення нахилу особи та шести основних рис осіб, необхідних для нанесення маски. На підставі нахилу особи з бібліотеки масок обирається відповідний шаблон маски. Потім шаблонна маска трансформується на основі шести ключових особливостей, щоб вона ідеально відповідала особі. Повну блок-схему можна побачити нижче (рис. 1.8).

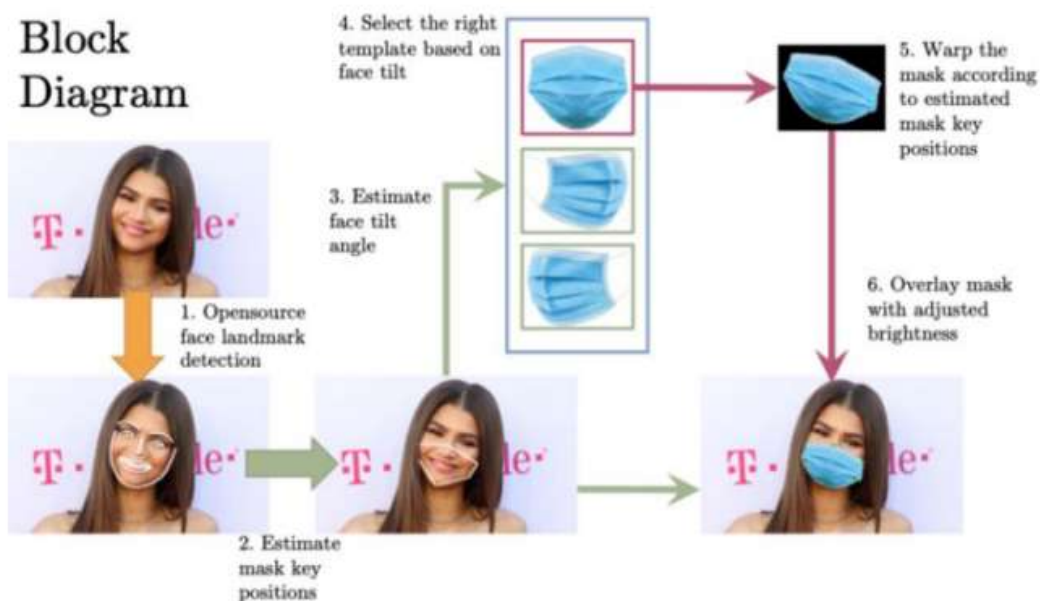


Рисунок 1.8 – схема створення датасету MaskTheFace

Наразі датасет MaskTheFace містить наступні 4 види масок (рис.1.9).



Рисунок 1.9 – Види масок присутніх у датасеті

MaskTheFace можна використовувати для створення набору даних синтетичної маски обличчя, який потім можна використовувати для навчання різних прикладних мереж машинного навчання, таких як розпізнавання осіб для системи відвідуваності у приміщенні, виявлення масок, класифікація масок тощо.

MFR2 – це невеликий набір даних з 53 особистостями публічних осіб, загалом 269 зображень, зібраних з мережі Інтернет. Кожна особа має у середньому 5 зображень. Набір даних містить як замасковані, і немасковані обличчя осіб. Набір даних обробляється з точки зору вирівнювання особи та розмірів зображення. Кожне зображення має розмір $(160 \times 160 \times 3)$. Приклади зображень з набору даних MFR2 та розподіл масок представлено на далі рисунку (рисунок 1.10).

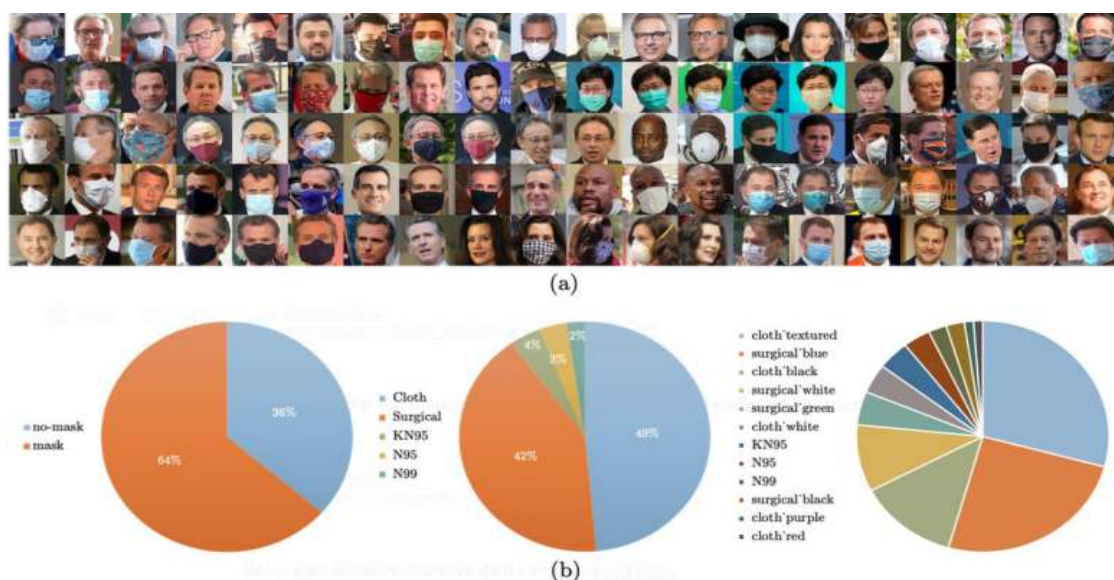


Рисунок 1.10 – Набори даних MFR2

Обробка кадрів відеопотоку складається з двох основних етапів. На першому етапі відбувається виявлення осіб у кадрі, а на другому – виявлення захисної маски на обличчі [18].

При розробці системи, був використаний метод виявлення осіб за ознаками каскаду Хаара, а для розпізнавання захисної маски на обличчі був застосований нейромережевий метод. Як датасет для навчання нейронної мережі був використаний набір зображень осіб у масці та без масок. Загальна кількість зображень у датасеті складає 1376 штук.

Для розробки системи розпізнавання був обраний метод виявлення осіб Віюлі-Джонса. Цей метод був розроблений у 2001 році, але завдяки своїй високій швидкості, а також вкрай низької ймовірності помилкового виявлення особи, досі є одним з основних методів пошуку об'єктів на зображеннях.

Основні принципи, на яких ґрунтується робота даного методу:

- подання зображення в інтегральному вигляді;
- пошук осіб за допомогою ознак Хаара;
- каскадна класифікація;

У розширеному методі Віюлі-Джонса, який використовується в даному проєкті, використовуються додаткові ознаки Хаара. Каскади Хаара являють собою прямокутні

області, які складені з кількох сусідніх прямокутних областей, відзначених як світла або темна, приклади таких додаткових ознак представлені рисунку 1.11.

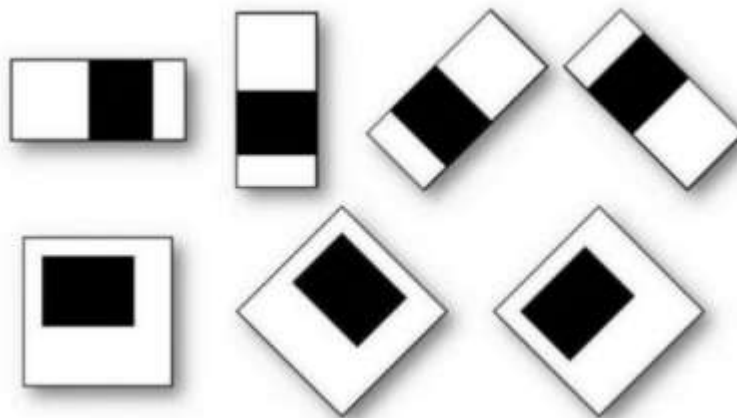


Рисунок 1.11 – Додаткові ознаки Хаара

Для того щоб навчити модель нейромережевого детектора маски на обличчі, необхідно розбити ядро програми на дві окремі фази, кожна зі своїм відповідним етапом наведені на рис.1.12:



Рисунок 1.12 - Алгоритм обробки кадрів відеопотоку для виявлення осіб та маски на обличчі

Навчання моделі згорткової нейронної мережі: на даному етапі, акцентується увага на завантаженні набору даних з пам'яті комп'ютера для навчання моделі для виявлення маски на обличчі (з використанням Keras / TensorFlow), далі зберігається модель з найкращими показниками [19].

Розгортання: після того, як нейромережева модель для детектування маски навчена, вона завантажується до детектора маски з диску для застосування етапу виявлення осіб, а потім класифікації осіб в захисній масці або без захисної маски.

Для того, щоб вбудувати видимі риси обличчя в 128-розмірності, необхідно обрізати виявлених осіб, і для цього використовується InceptionV3, через її легку архітектуру та точність. Реалізовано за допомогою Keras - це високорівневий API для глибокого навчання, який дозволяє легко будувати, навчати, оцінювати та використовувати різноманітні нейронні мережі на платформі TensorFlow. Бібліотека TensorFlow — це повнофункціональна платформа машинного навчання з відкритим кодом [20]. Її можна сприймати як рівень інфраструктури для диференційованого програмування. TensorFlow поєднує в собі чотири ключові можливості:

- ефективне виконання тензорних операцій низького рівня на CPU, GPU або TPU;
- обчислення градієнта довільних диференційованих виразів;
- масштабування обчислень до багатьох пристроїв (наприклад, суперкомп'ютер Summit в Національній лабораторії Oak Ridge, який охоплює 27 000 графічних процесорів);
- експорт програм ("графів" програм) до зовнішніх середовищ виконання, таких як сервери, браузері, мобільні та вбудовані пристрої.

Для доступу до засобів TensorFlow використовується API, написане на мові Python.

Такий підхід є ефективним, оскільки нейромережа намагається виявляти ознаки присутності чи відсутності маски, лише у той фрагмент зображення, де справді є особи. Це дозволяє покращити швидкість розпізнавання маски на обличчі.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Опис та призначення системи

Система ідентифікації порушень призначена для підвищення безпеки працівників на підприємстві використовуючи постійний моніторинг на предмет використання ЗІЗ. Моніторинг відбувається із кадрів відеопотоку в режимі реального часу.

Для реалізації даної системи необхідно в цеху, де є особливо небезпечні умови праці та підвищені вимоги до наявності захисних окулярів та каски, встановити відеокамеру, відеокадри з якої в режимі реального часу передають дані на головний модуль керування.

Система ідентифікації порушень використання ЗІЗ складається з декількох підсистем. Схема системи ідентифікації порушень використання індивідуальних засобів в загальному вигляді наведено на рис. 2.1.

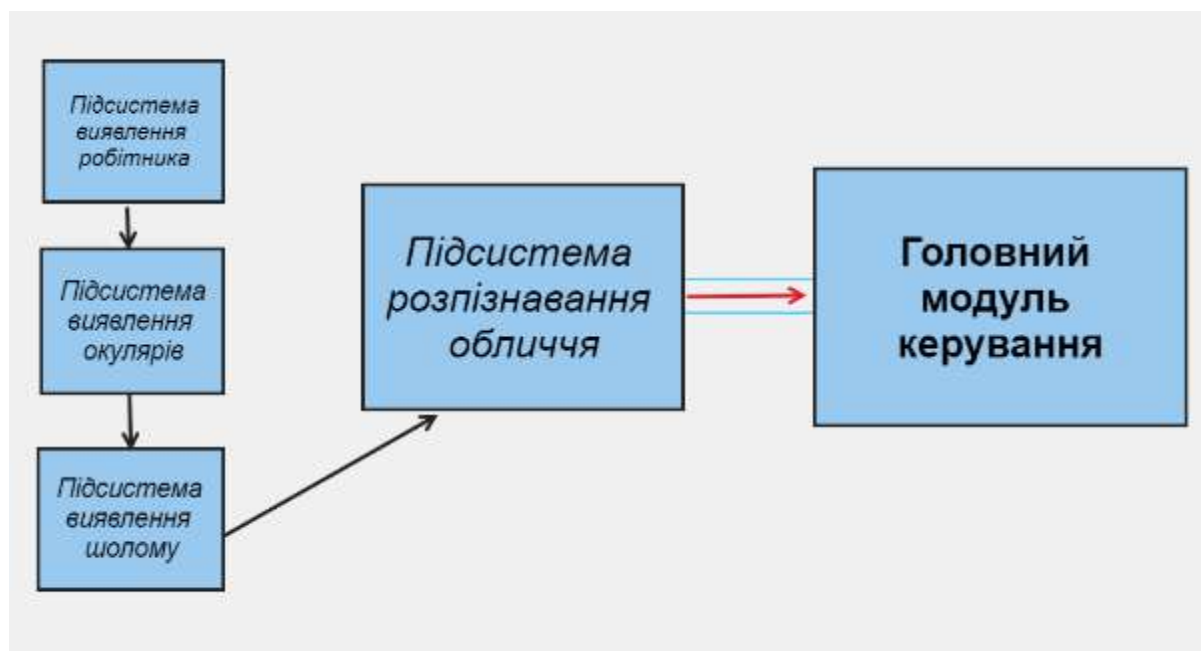


Рисунок 2.1 - Схема системи ідентифікації порушень використання ЗІЗ

2.1.1 Підсистема виявлення робітника

Отримані відеокадри обробляє нейронна мережа, щоб ідентифікувати зображення людини, та виділити її обличчя для подальшого розпізнавання.

Далі виділене зображення обличчя передається на підсистему виявлення окулярів (рис.2.2).



Рисунок 2.2 – Підсистема виявлення робітника

2.1.2 Підсистема виявлення окулярів

На цьому етапі нейронна мережа проводить порівняльний аналіз використовуючи зразки захисних окулярів із бази даних на наявність на обличчі захисних окулярів. Якщо окулярів не виявлено, то підсистема одразу передає інформацію на підсистему розпізнавання обличчя порушника. У іншому випадку, відеокادر передається на підсистему виявлення шолому (рис.2.3).

2.1.3 Підсистема розпізнавання особи порушника

Призначена для ідентифікації особи порушника техніки безпеки на підприємстві. Нейронна мережа, використовуючи фото співробітників із бази даних, проводить порівняльний аналіз для виявлення особи порушника. Ідентифікувавши особу, передає інформацію на головний модуль керування з червоним індикатором (рис.2.4).



Рисунок 2.3 – Підсистема виявлення окулярів



Рисунок 2.4 – Підсистема розпізнавання особи порушника

2.1.4 Підсистема виявлення шолому

Призначена для виявлення шолому (робітничої каски) на особі, що працює в особливо небезпечних умовах. Для цього нейронна мережа проводить порівняльний аналіз використовуючи зразки шоломів із бази даних на наявність на особі захисної. Якщо каски не виявлено, то підсистема передає інформацію на підсистему розпізнавання обличчя порушника для подальшої його ідентифікації (рис.2.5).

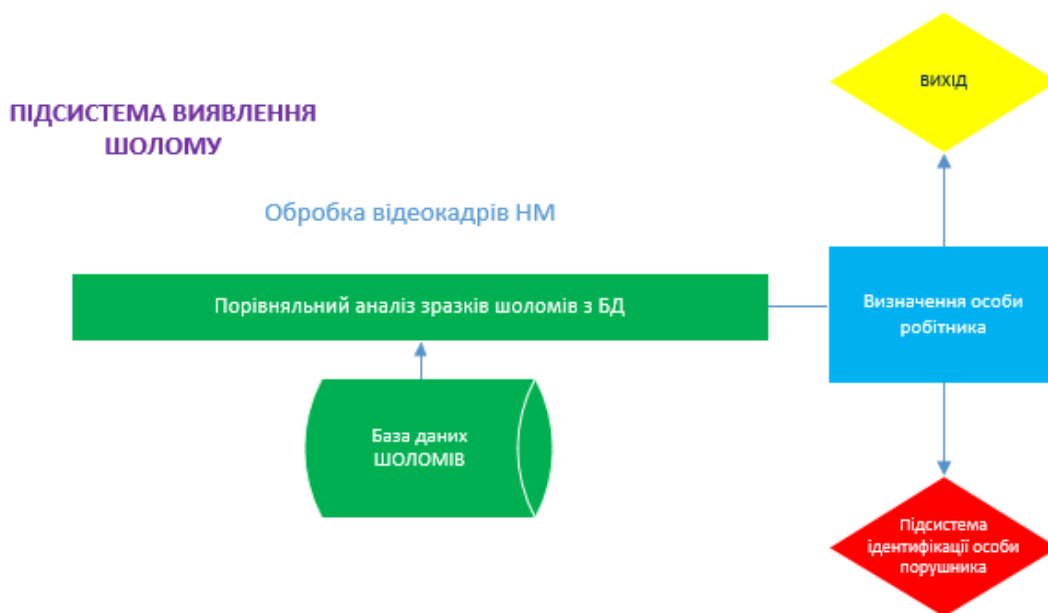


Рисунок 2.5 – Підсистема виявлення шолому

2.1.5 Головний модуль керування

Призначений для моніторингу та керування процесом фіксації порушників техніки безпеки на підприємстві. Коли на пульт керування подається інформація про порушника, диспетчер у першу чергу, використовуючи гучномовець (встановлений в цеху) звертається до порушника з проханням надіти окуляри та (або) шолом. Якщо порушено техніку безпеки перший раз, то порушнику виноситься попередження, та фіксується інформація в БД. Якщо другий раз, то інформація передається в бухгалтерію та виписується штраф 15% заробітної плати. Якщо третій раз, то штраф збільшується до 30%, та робітника відправляють на повторне навчання з техніки

безпеки на підприємстві. На четвертий раз, розглядається звільнення робітника, який систематично порушує техніку безпеки на підприємстві (рис.2.6).

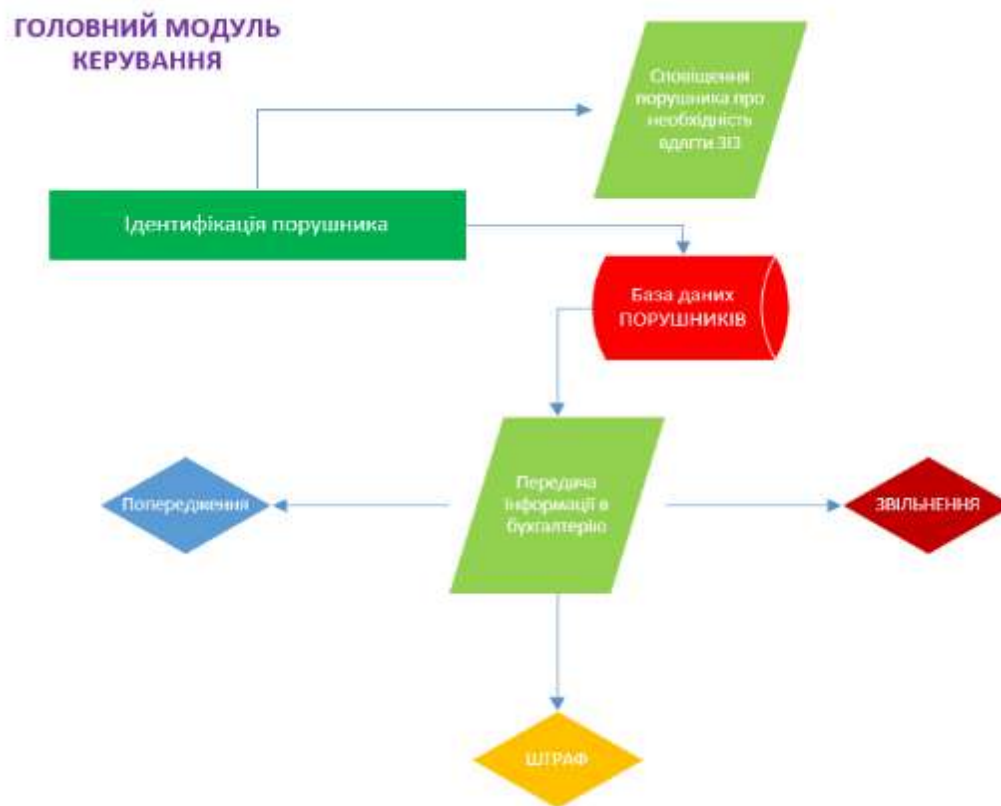


Рисунок 2.6 – Головний модуль керування

2.2 Функціонування системи

Узагальнений алгоритм роботи системи ідентифікації порушень використання засобів індивідуального захисту представлено на рис.2.7.

Спершу ми завантажуюмо навчену модель з диску, потім виконуємо пошук обличчя із відеопотоку. Для виявлення окулярів, ми виділяємо область очей, та приміняємо до неї класифікатор для виявлення присутності або відсутності окулярів на обличчі. Далі працює підсистема виявлення шолому, для цього виділяємо лобну частину і голову, приміняємо класифікатор для виявлення присутності або

відсутності шолому. Далі виділяємо обличчя і передаємо на головний пульт керування.

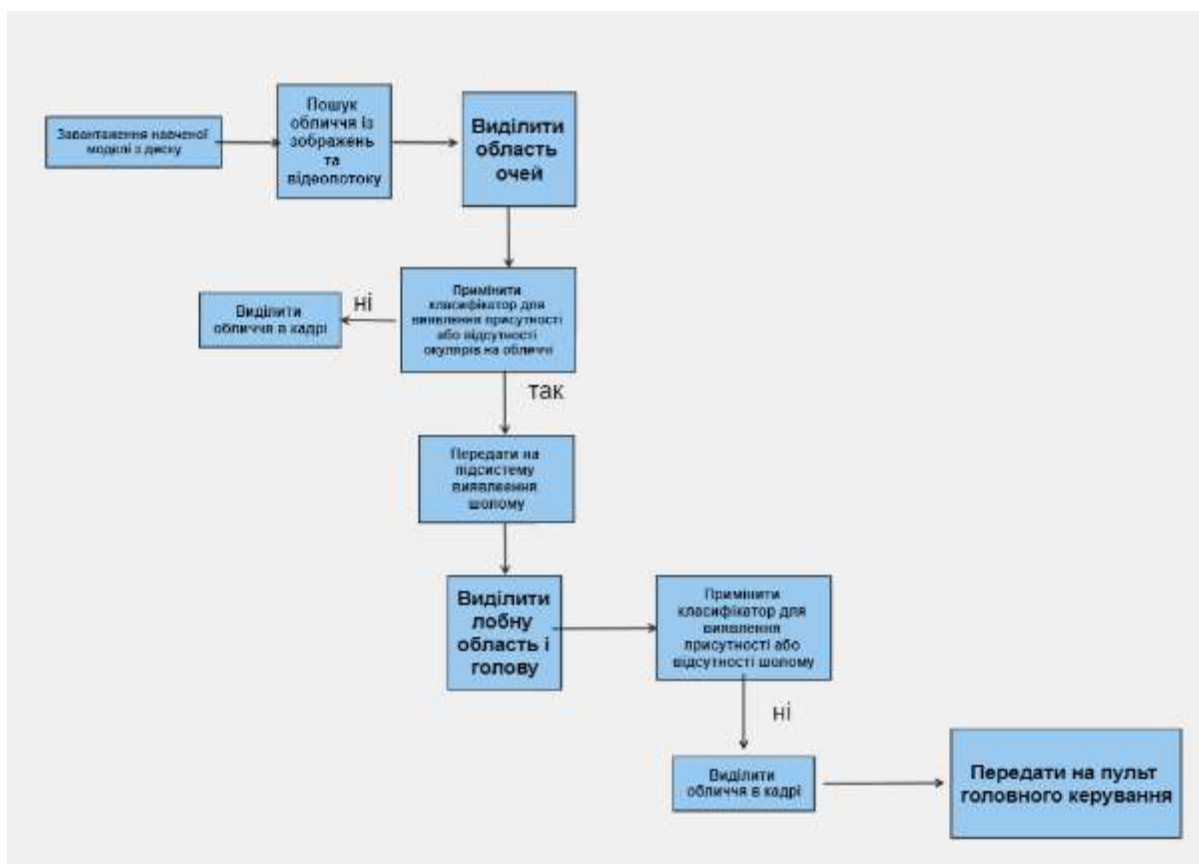


Рисунок 2.7 – Блок-схема алгоритму роботи системи ідентифікації порушень використання ЗІЗ

3 РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Вимоги до апаратного та програмного забезпечення

Спроектована система ідентифікації порушень передбачена для підприємств, що мають особливо небезпечні умови праці. Наприклад, для реалізації в цеху на металургійному підприємстві, тому апаратне забезпечення системи повинно працювати при несприятливих умовах праці. Нашою системою передбачено ідентифікація осіб, що нехтують засобами індивідуального захисту (ЗІЗ) і потенційно

наражають себе на небезпеку [28]. Задачею системи знизити кількість нещасних випадків на підприємстві, шляхом фіксування порушень та розпізнавання порушників.

Для реалізації проєкту передбачається встановити в цеху чотири відеокамери. Обираємо розумну камеру відеоспостереження Logitech Circle 2 Wire-Free Indoor (рис.3.1).



Рисунок 3.1 – Камера відеоспостереження Logitech Circle 2 Wire-Free Indoor

Відеокамера Logitech Circle 2 призначена для встановлення всередині приміщення або на вулиці. Герметичний корпус захищений від води, вологи, вітру, піску та перепадів температури, тому зміна погодних умов не впливає на працездатність пристрою. Вона має високу якість відео, широкий кут огляду, трансляцію в режимі реального часу. Дана камера підтримує функцію нічного бачення та зйомку в умовах низького освітлення, це дуже корисна функція, так як в цеху підвищена запиленість повітря. Кут огляду 180 градусів охоплює максимальну площу і записує відео з роздільною здатністю 1080p Full HD. Цифрове збільшення об'єкта здійснюється автоматично за потреби користувача, а дані передаються по Wi-Fi.

Відзняті матеріали передаються у хмару з використанням протоколів безпеки, що дуже важливо на підприємстві яке має вимоги до політики конфіденційності інформації. Процес ідентифікації та цифровий захист гарантують конфіденційність даних та неможливість отримання інформації по стороннім особам. Живлення камери здійснюється від мережі або вбудованого акумулятора 6400 мАч, що актуально при аварійних відключеннях світла.

Доступ до відеозапису з камери відкривається через програму Circle, яку необхідно встановити на комп'ютер.

Для побудови моделі ідентифікації ЗІЗ у режимі реального часу беремо мобільну станцію Lenovo ThinkPad P16v (рис.3.2).



Рисунок 3.2 - Мобільна станція Lenovo ThinkPad P16v

Даний ноутбук забезпечить високу продуктивність роботи, за рахунок потужності робочої станції високого класу. Мобільна станція має процесор Intel Core 13-го покоління, а також дискретну відеокарту.

Мобільна станція захищена системою ThinkShield. Це комплекс апаратних і програмних рішень для забезпечення безпеки, що включає вбудований сканер відбитків пальців для зручного входу в систему і BIOS, що відновлюється, або дозволяє відновлювати попередні налаштування системи. Мобільна станція Lenovo

ThinkPad P16v також пройшла випробування за стандартами Міністерства оборони США для забезпечення максимальної надійності та довговічності.

Ми використовували операційну систему Ubuntu 18.04 з усім необхідним програмним забезпеченням. Ubuntu - одна з найпопулярніших операційних систем, її використовують, як і на комп'ютерах, так і на серверах.

3.1.1 Налаштування апаратного забезпечення

По-перше, необхідно перевірити чи правильно налаштована камера. З Ubuntu це зробити просто: щоб перевірити, чи пристрій розпізнаний операційною системою. Для цього необхідно перевірити всі підключені відео пристрої, надруквавши командний рядок `ls /dev/video*`. В результаті вийде список усіх відеопристроїв, що підключені до системи. Якщо камера правильно підключена, новий пристрій буде відображено в результаті виконання команди.

Встановимо програму Circle для перегляду прямої трансляції в HD-якості, та бачити історію відео. Дані максимально убезпечені за допомогою сучасного цифрового захисту запису та унікального процесу ідентифікації. Така процедура запобігає доступу до особистої інформації іншим особам. Усі відеоматеріали надійно передаються у хмару з використанням найновіших стандартних протоколів безпеки.

Передача даних відбувається по Wi-Fi 802.11 a/b/g/n 2.4 ГГц/5.0 ГГц. Якщо немає інтернету, то дані зберігаються у папці на мобільній станції, а при відтворенні доступу до інтернету, автоматично завантажуються в хмару.

Лістинг А.1, зазначений у цій роботі, написаний за допомогою Python (версія 3.5). Для встановлення Python ми будемо використовувати Anaconda – популярний дистрибутив Python для обробки та аналізу даних.

Anaconda - це відкрите програмне забезпечення, яке містить набір інструментів для обробки та аналізу даних, машинного навчання, наукових обчислень та роботи з штучним інтелектом. Основний фокус Anaconda - це створення зручного середовища для роботи з науковими обчисленнями, де ви можете легко управляти пакетами та залежностями. Це інтегроване середовище розробки (IDE) для Python, зокрема орієнтоване на роботу з науковими обчисленнями. Воно має потужні інструменти для

відлагодження, профілювання та інтерактивного використання Python. Anaconda включає в себе Python, однак вона також має власне управління пакетами, що робить його окремим від стандартного Python, який можна завантажити з офіційного веб-сайту.

Одним із основних компонентів Anaconda є Conda Package Manager. Conda - це система управління пакетами, яка дозволяє вам легко встановлювати, оновлювати та видаляти пакети. Вона автоматично вирішує проблеми залежностей, що робить процес управління пакетами більш простим і консистентним. За допомогою Conda ви можете створювати та керувати віртуальними середовищами, що дозволяє ізолювати проекти та їх залежності один від одного. Це корисно для управління версіями пакетів та забезпечення консистентності середовищ.

Anaconda включає в себе багато популярних пакетів для наукових обчислень та аналізу даних, таких як NumPy, SciPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, PyTorch та інші. Це робить його ідеальним для роботи з широким спектром завдань у сфері даних та машинного навчання.

Anaconda також містить Jupyter, інтерактивне середовище для програмування, що дозволяє створювати та ділитися документами, які містять код, графіки, текст та виконуються живо.

Anaconda підтримує Windows, macOS та Linux, що дозволяє користувачам працювати на різних операційних системах.

Anaconda також має Anaconda Navigator. Це графічний інтерфейс для управління середовищами, пакетами та інструментами Anaconda. Крім того, Anaconda має велику спільноту користувачів і високоякісну документацію, що допомагає користувачам швидко оволодіти інструментами та розв'язати будь-які питання.

Загалом, Anaconda є потужним інструментом для наукових обчислень і роботи з даними в Python, що робить його популярним серед дослідників, аналітиків даних і спеціалістів у галузі штучного інтелекту.

Встановлення OpenCV. OpenCV – бібліотека з відкритим кодом, яка призначена для створення програм комп'ютерного зору. OpenCV (Open Source Computer Vision Library) - це відкрите програмне забезпечення для комп'ютерного зору та обробки зображень. Розроблено з метою надання набору інструментів для створення програм з комп'ютерного зору в реальному часі. Основні характеристики та можливості OpenCV включають:

- підтримка різних мов програмування;
- обробка та аналіз зображень;
- комп'ютерний зір і відеоаналітика;
- машинне навчання;
- робота в реальному часі;
- підтримка роботи з відеопотоками;
- загальна призначеність.

OpenCV розроблено для підтримки різних мов програмування, включаючи C++, Python, Java та інші. Це дозволяє розробникам використовувати OpenCV в рамках їхнього улюбленого середовища розробки. Ця бібліотека має широкий спектр функцій для обробки та аналізу зображень, включаючи зчитування та запис зображень, обрізку, зміну розміру, фільтрацію, конвертацію кольорів та інші операції.

OpenCV містить алгоритми для комп'ютерного зору, такі як виявлення облич, розпізнавання облич, слідування об'єктів, детекція руху та багато інших. Це робить його корисним для застосувань від відеоспостереження до робототехніки. Вона надає підтримку для використання моделей машинного навчання, включаючи інтеграцію з бібліотеками, такими як TensorFlow та PyTorch. Це робить OpenCV потужним інструментом для застосувань у сфері машинного навчання та глибокого навчання.

OpenCV оптимізовано для використання в реальному часі, що робить його ідеальним для застосувань, де потрібна швидка обробка зображень, наприклад, в робототехніці, мобільних додатках або системах відеоспостереження. Вона може обробляти відеопотоки з веб-камер, файлів відео або інших джерел, дозволяючи створювати застосунки для реального часу та обробки великих обсягів відеоданих.

OpenCV є універсальною бібліотекою, яка може бути використана в різних галузях, включаючи комп'ютерний зір, медичне зображення, астрономію, аналіз зображень та інше. OpenCV активно підтримується та розвивається великою спільнотою розробників та використовується в численних проєктах з комп'ютерного зору та обробки зображень.

Встановлення OpenCV здійснюється за допомогою `pip`:

Лістинг 3.1 – Встановлення OpenCV

```
pip3 install opencv-python
```

Далі встановимо `face_recognition` API. Ми будемо використовувати `face_recognition` API, який вважається найпростішим API для розпізнавання облич на Python у всьому світі. Для встановлення будемо використовувати:

Лістинг 3.2 – встановлення `face_recognition` API

```
pip install dlib
pip install face_recognition
```

Далі створимо файл `face_detector.py` і скопіюємо наведений нижче код:

Лістинг 3.3 – код для розпізнавання облич на Python

```
# import libraries
import cv2
import face_recognition

# Get a reference to webcam
video_capture = cv2.VideoCapture("/dev/video1")

# Initialize variables
face_locations = []

while True:

# Grab a single frame of video
```

```

ret, frame = video_capture.read()

# Convert the image from BGR color (which OpenCV uses) to RGB color (which
face_recognition uses)

rgb_frame = frame[:, :, ::-1]

# Find all the faces in the current frame of video

face_locations = face_recognition.face_locations(rgb_frame)

# Display the results

for top, right, bottom, left in face_locations:

# Draw a box around the face

cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

# Display the resulting image

cv2.imshow('Video', frame)

# Hit 'q' on the keyboard to quit!

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

break

# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()

```

Потім запусимо цей файл Python, надрукувавши:

Лістинг 3.4 - файл Python для запуску

```
python face_detector.py
```

Якщо все працює правильно, відкриється нове вікно із запущеним режимом розпізнавання облич у реальному часі.

Підіб'ємо підсумки і пояснимо, що зробив наш код:

- спочатку ми вказали апаратне забезпечення, на якому проводитиметься аналіз відео;
- далі зробили захоплення відео у реальному часі кадр за кадром;
- потім обробили кожен кадр та витягли місцезнаходження всіх осіб на зображенні;
- у результаті відтворили ці кадри у формі відео разом із вказівкою на те, де розташовані особи.

Далі встановлюємо бібліотеку Dlib.

Dlib – це крос-платформна бібліотека з відкритим вихідним кодом для розпізнавання та обробки зображень. Вона надає можливості виявлення та розпізнавання осіб, визначення ключових точок на обличчі, а також вирішення інших завдань комп'ютерного зору.

Встановимо попередні залежності, виконавши команду:

Лістинг 3.5 – команда для встановлення бібліотеки

```
udo apt-get install build-essential cmake
```

Далі виконаємо такі команди по черзі:

Лістинг 3.6 – команди для встановлення бібліотеки

```
git clone https://github.com/davisking/dlib.git
cd dlib
mkdir build
cd build
cmake ..
cmake --build .
sudo make install
sudo ldconfig
```

Лістинг 3.7 – команда для перевірки, чи встановлено бібліотеку

```
python -c "import dlib; print(dlib.__version__)"
```

Після успішного встановлення Dlib бібліотеки на комп'ютер, ми можемо перевірити її роботу, виконавши кілька простих кроків описаних нижче.

Відкриємо середовище розробки на комп'ютері. Далі імпортуємо бібліотеку Dlib у проєкт за допомогою команди

Лістинг 3.8 – імпорт бібліотеки Dlib

```
import dlib
```

Перевіримо, чи Dlib працює правильно. Для цього виконаємо код та перевіримо його результати.

Лістинг 3.9 – код для перевірки бібліотеки

```
import dlib
# Створіть об'єкт детектора обличчя
detector = dlib.get_frontal_face_detector()
# Завантажне зображення для обробки
image = dlib.load_rgb_image('image.jpg')
# Застосуйте детектор обличчя до зображення
faces = detector(image)
# Виведіть кількість виявлених осіб
print("Кількість облич:", len(faces))
```

Якщо Dlib встановлений правильно і працює, після виконання цього коду ми повинні побачити кількість виявлених осіб на зображенні. Тепер можна почати використовувати її для різних завдань комп'ютерного зору.

3.1.2 Бібліотеки

Для спроектованої системи ідентифікації контролю носіння ЗІЗ, щоб мати можливість реалізувати роботу з потоковим відео, та аналізувати отримані дані в режимі реального часу, ми обрали бібліотеку OpenCV. Дана бібліотека містить зручний функціонал по роботі із зображеннями, та має понад 2500 сучасних

алгоритмів для комп'ютерного зору машинного навчання, а також для та розпізнавання осіб [19]. Ця бібліотека дозволяє представити отримані відеокадри як набір фреймів, кожен з яких ми можемо розглядати як окреме зображення.

Обрана бібліотека поділена на функціональні модулі. У проєкті були використані такі модулі та функції:

- модуль `videoio`, що містить функції, які надають можливість читання та запису відео та зображень;
- функція `VideoCapture()` дозволяє створювати підключення до камери для отримання зображення в режимі реального часу;
- функція `read()` дозволяє зчитувати поточний кадр з камери. Повертає тривимірну матрицю із кольоровою моделлю BGR (Blue, Green, Red);
- модуль `imgproc` дозволяє проводити обробку зображень;
- функція `resize()` змінює розмір (масштаб) зображень;
- функція `rectangle()` додає прямокутну рамку до кадру;
- модуль `highgui` дозволяє візуалізувати зображення без необхідності написання інтерфейсу користувача;
- функція `imshow()` — відображає кадри в окремому вікні;
- `destroyAllWindows()` — функція, що припиняє роботу всіх вікон OpenCV;
- функція `waitKey()` обробляє сигнали натискання клавіш.

Для виділення ключових точок на обличчі обираємо бібліотеку `dlib` [20], що розробляється на C++. Вона містить безліч готових інструментів:

- кілька реалізацій нейронних мереж;
- функції матричної геометрії;
- методи градієнтного спуску.

Далі ми використовуємо бібліотеку `Face Recognition`[21] — вона, заснована на `dlib` і розробляється мовою `python`. Позиціонується як найпростіший інструмент, що дозволяє проводити розпізнавання осіб та управління зібраними даними. У цій роботі були використані такі функції:

- функція `load_image_file()` зчитує зображення та перетворює його на матрицю;
- функція `face_encodings()` здійснює виявлення та вилучення характеристик особи з кадру (матриці);
- функція `face_locations()` повертає матрицю (прямокутник), яка відповідає місцезнаходженню осіб у кадрі;
- функція `compare_faces()` здійснює порівняння списку з кодуванням робітників та кодуванням особи, яку необхідно перевірити. Повертає список значень True/False, що вказує робітника, якому відповідає особа, що визначається;
- функція `face_distance()` — аналог функції `compare_faces()`, що повертає евклідову відстань між порівнюваними особами, що показує чисельну схожість осіб.

3.1.3 Алгоритм роботи системи ідентифікації обличчя

Система працює відповідно до алгоритма, представленого у вигляді блок-схеми на рис.3.3. Алгоритм працює наступним чином:

- считуємо з файлу характеристики обличч робітників;
- перевіряємо чи є підключення до камери;
- отримані відеокадри розбиваємо на фрейми;
- перевіряємо чи є в кадрі обличчя;
- якщо в кадрі є розпізнано обличчя, то отримуємо список облич;
- порівнюємо отримані обличчя зі списком робітників;
- перевіряємо чи розпізнано обличчя;
- якщо «так», то додаємо в кадр підпис з піб робітника;
- якщо «ні», то «не ідентифіковано»;
- малюємо фрейм у вікні;
- перевіряємо чи натиснута кнопка «вихід»;

Якщо натиснута кнопка «Вихід» завершуємо алгоритм, закриваємо всі вікна та відключаємось від камери, інакше алгоритм починає роботу знову.

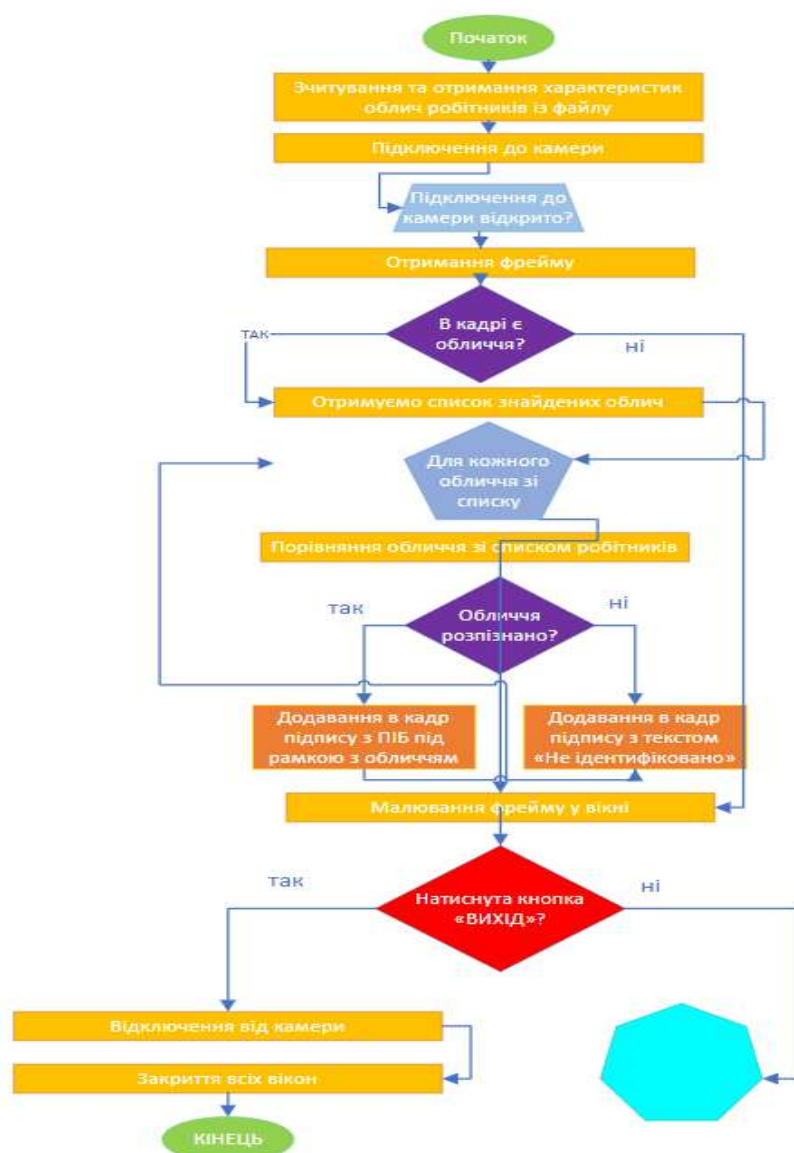


Рисунок 3.3 – Алгоритм роботи системи розпізнавання обличчя

3.1.4 Датасети

Для навчання нейронної мережі, в якості набору даних використовуємо:

- датасет з робітниками;
- датасет з робітничими касками;
- датасет з робітничими окулярами;
- датасет з робітниками в касці та окулярах.

Бібліотека `dlib` в нашому проєкті використовується для визначення нахилу голови та виділення основних дванадцяти основних точок на обличчі для нанесення

робітничого шолому (каска) та захисних окулярів. Після того як визначено нахил голови з бібліотеки обирається шаблонна каска і шаблонні окуляри, які на основі 12 ознак трансформуються на обличчя (рис.3.4).



Рисунок 3.4 – Схема створення датасету

Датасет робітничих шоломів (касок) зображено на рис.3.5



Рисунок 3.5 – Датасет робітничих шоломів

Датасет захисних окулярів зображено на рис.3.6.



Рисунок 3.6 – Датасет захисних окулярів

Отже спочатку необхідно ідентифікувати людей на зображеннях (рис.3.7), потім ми ідентифікуємо їх на класи: «в окулярах», «в шоломі», «без окулярів», «без шолому».

Результати роботи моделей для класифікації шоломів зображено на рис.3.8.

Роздивившись різні архітектури нейронних мереж для класифікації зображень ми вирішили зупинитися на Inception v3, вирішивши скористатися тим, що він призначений для роботи зі змінним розміром зображень. Inception v3 реалізований за допомогою Keras – неймережевої бібліотеки з відкритим вихідним кодом, що працює на базі бібліотеки TensorFlow, що розроблена для реалізації завдань комп'ютерного зору [22]. Такий підхід є ефективним, оскільки дана неймережа намагається виявляти ознаки присутності чи відсутності ЗІЗ, лише у тому фрагменті зображення, де справді є особи.



Рисунок 3.7 – Ідентифікація людей в кадрі

Для проєкту було підготовлено багато вирізаних фото з робітниками. Для навчання класифікаторів необхідно буде використовувати зображення розміром 150x400.

Щоб навчити мережу розпізнавати ЗІЗ, передусім потрібно зібрати датасет з розмічених прикладів.



Рисунок 3.8 – Результат ідентифікації шоломів

В нашому проєкті, для навчання було обрано нейронну мережу Faster R-CNN [22]. Дана нейронна мережа буде навчатись на датасеті із 4 класами (рис.3.9):

- glasses;
- not_glasses;
- helmet;
- not_helmet.



Рисунок 3.9 – Результат навчання нейронної мережі Faster R-CNN

Основні проблеми навчання які можуть бути пов'язані з датасетом для робітничих шоломів (каска).

Для покращення розпізнавання датасету шоломів, необхідно прибрати зображення, на яких обрізані каски більше, ніж наполовину, щоб уникнути плутанини з кепками (рис.3.10). Ще одна проблема з якою можна стикнутися, це якщо у людини в руках каска, то модель може не робачити порушень: «Каска є? Є!». Тому необхідно також видалити з навчального датасету всі кадри, на яких людина притримує каску рукою, навіть якщо каска в цей момент перебуває на голові.



Рисунок 3.10 – Етап створення датасету касок

Після видалення невідповідних зображень необхідно розпочати безпосередньо розмітку (для ідентифікації об'єктів). При розмітці касок та окулярів оптимально захоплювати обличчя людини до кінчика носа (рис.3.11).



Рисунок 3.11 – Тестування моделі

Результати на валідаційній вибірці роботи над датасетом представлено у розділі 4.

3.2 Навчання НМ

Після виконання попередньої роботи з отримання та підготовки даних для навчання моделі, була реалізована архітектура та навчена модель згорткової нейронної мережі.

Результати навчання наведено на рис.3.12. Можна помітити, що модель добре навчилася і досягла точності розпізнавання, що дорівнює 98%.

В результаті тестування були отримані очікувані результати, вони полягають у тому, що застосування навчених згорткових нейронних мереж, дозволяє виявляти окуляри та шоломи на розпізнаних обличчях.

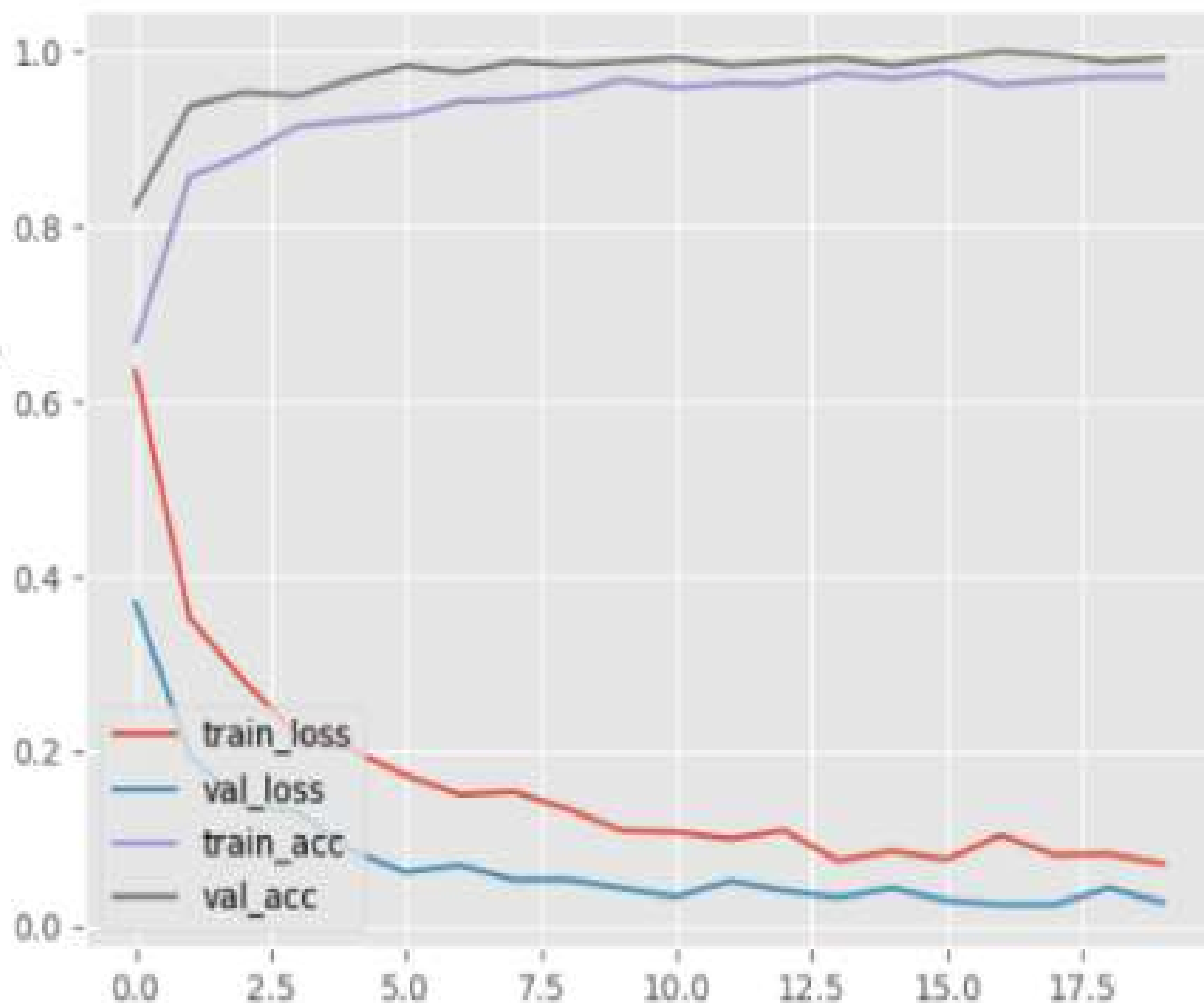


Рисунок 3.12 – Результати навчання моделі

Досягнута точність розпізнавання ЗІЗ становить 98%, дозволяючи досить добре визначити носить людина ЗІЗ чи ні.

Поворот навколо вертикальної осі.

Використовувалися відео, в якому розташування осіб було змінено від під різними кутами. Поворот голови навколо вертикальної осі в діапазоні $[60; +60]$ градусів з кроком в 15 градусів. Результати роботи занесені у табл.3.2.

Можна бачити, що при 60 градусах алгоритм має хороші показники, що дорівнюють 86% точності, при 45 – більше 88%. Таким чином, алгоритм стабільно працює якщо кут повороту 60 градусів.

При використанні алгоритму, розпізнавалося додатково одне зображення, що містить обличчя, при куті 75% і два - при куті 65%, при цьому при меншому куті повороту змін немає.

Таблиця 3.2 – Результати роботи

Кут, градуси	Точність розпізнаних обличь в окулярах та шоломі (1/100%)
-60	0,82
-45	0,86
-30	0,92
-15	0,96
0	0,98
+15	0,95
+30	0,93
+45	0,88
+60	0,85

Це покращення незначне, але говорить про те, що початкове припущення про можливість апроксимації правильно. Варто зазначити, що в реальних умовах до системи потрапляють особи з поворотом по вертикальній осі менше 30-35 градусів.

Тест полягає у повороті осіб на 360 градусів інкрементами по 15 градусів. Результат: при поворотах аж до +/- 30 градусів обличчя ідентифікується.

3.3 Архітектура проєкту

Архітектура подільшої реалізації проєкту виглядає наступним чином (рис.3.13).

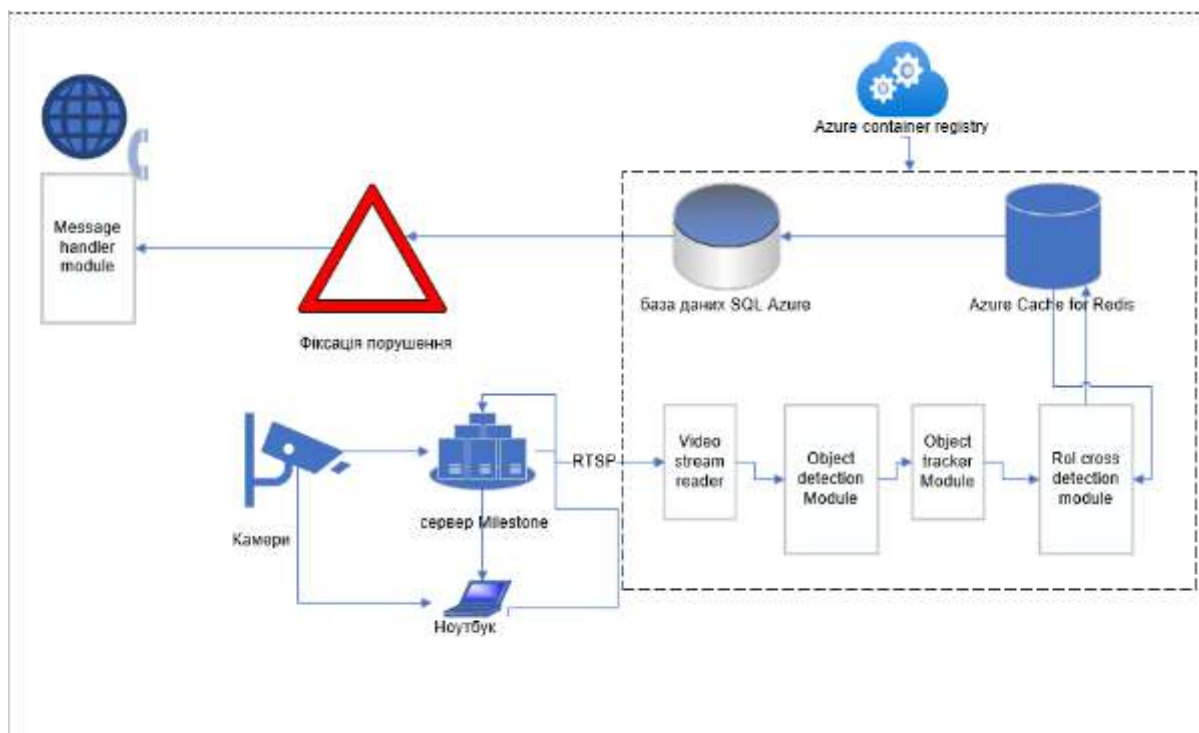


Рисунок 3.13 – Архітектура проєкту

Нижче приведемо більш детальний опис компонентів архітектури проєкту.

Дані з камер відеоспостереження надходять на хмарний сервер Milestone (рис.3.14). Використовуємо RTSP основний протокол, для передачі відеопотоку. Далі буде відбуватись розбиття відеопотоку (Video stream reader) на кадри (фрейми). Наступним буде спрацьовувати модуль по ідентифікації робочого персоналу (Object detection Module). Даний модуль працює на базі навченої нейронної мережі. Для наступного автоматичного запуску представляється у вигляді контейнера. Ми плануємо використовувати платформу для розробки Docker, вона працює для доставки та запуску контейнерних програм. Docker дозволяє створювати контейнери, автоматизувати їх запуск та розгортання, керує життєвим циклом. Він дозволяє запускати безліч контейнерів на одній машині.

Наступним буде запускатись модуль (Rol cross detection module), який буде відстежувати факт порушення (відсутність ЗІЗ). Для наступного автоматичного запуску представляється у вигляді Docker контейнера. Образи Docker гарантують єднання середовища виконання за рахунок їх віртуалізації (рис.3.15).

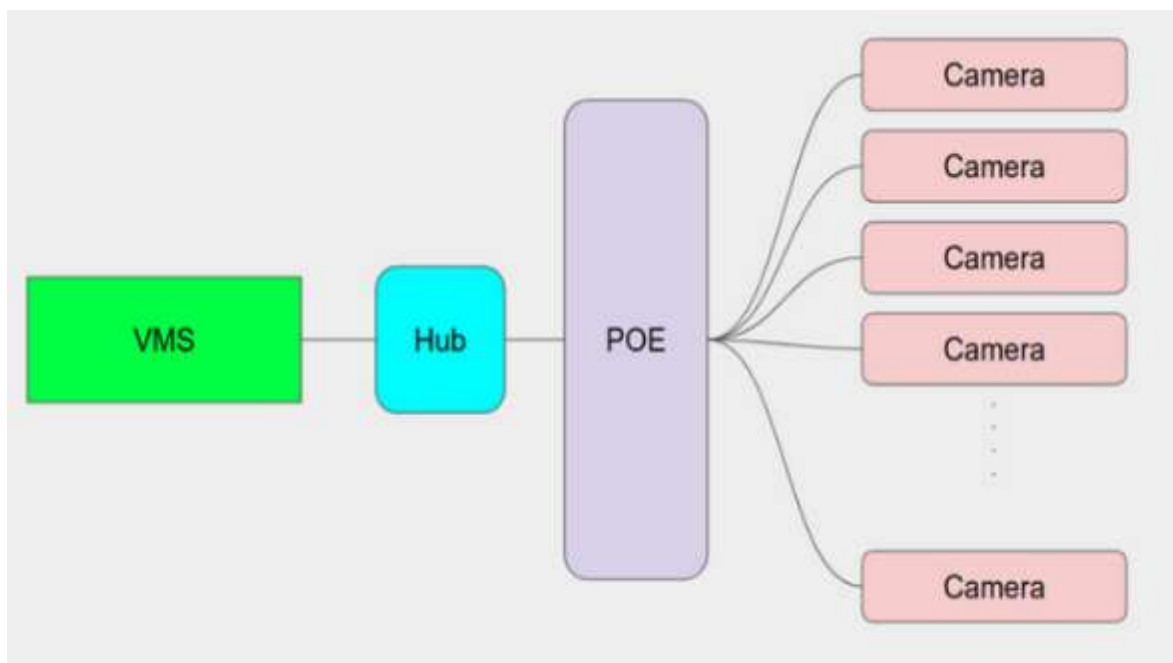


Рисунок 3.14 - Читання RTSP з відео потоку з Milestone

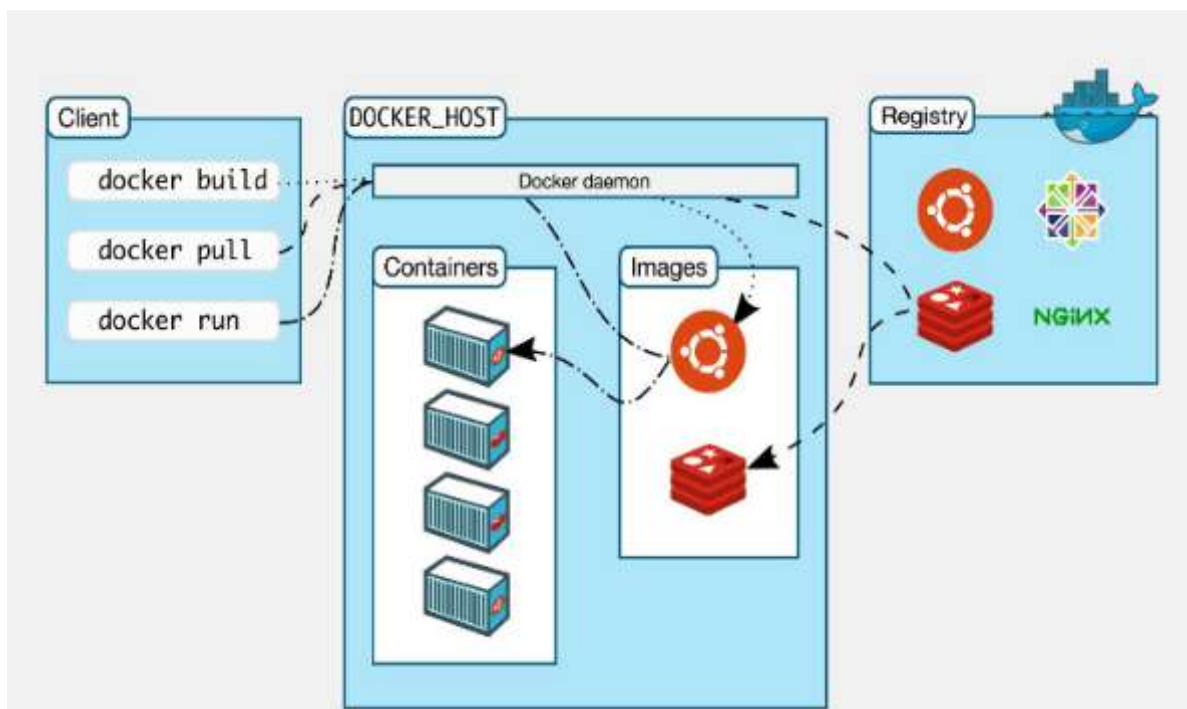


Рисунок 3.15 - Docker контейнер

Наступним запускається модуль (Object tracker Module), який буде ідентифікувати об'єкт, за рахунок присвоювання йому унікального id, який буде зберігатися доти поки об'єкт знаходиться в кадрі. Для наступного автоматичного запуску модуль представляється у вигляді Docker контейнера.

Для запису даних: час фіксованої події, координати bounding boxes, коротке текстове поле події, IP-адреси камери, використовуємо базу даних SQL Azure. Вона надає підтримку сучасних хмарних програм в інтелектуальній керованій службі баз даних, що включає безсерверні обчислення. Має такі переваги: об'єм до 3,5 млн повідомлень, зберігання даних протягом року, синхронізація з Power Bi (рис.3.16).

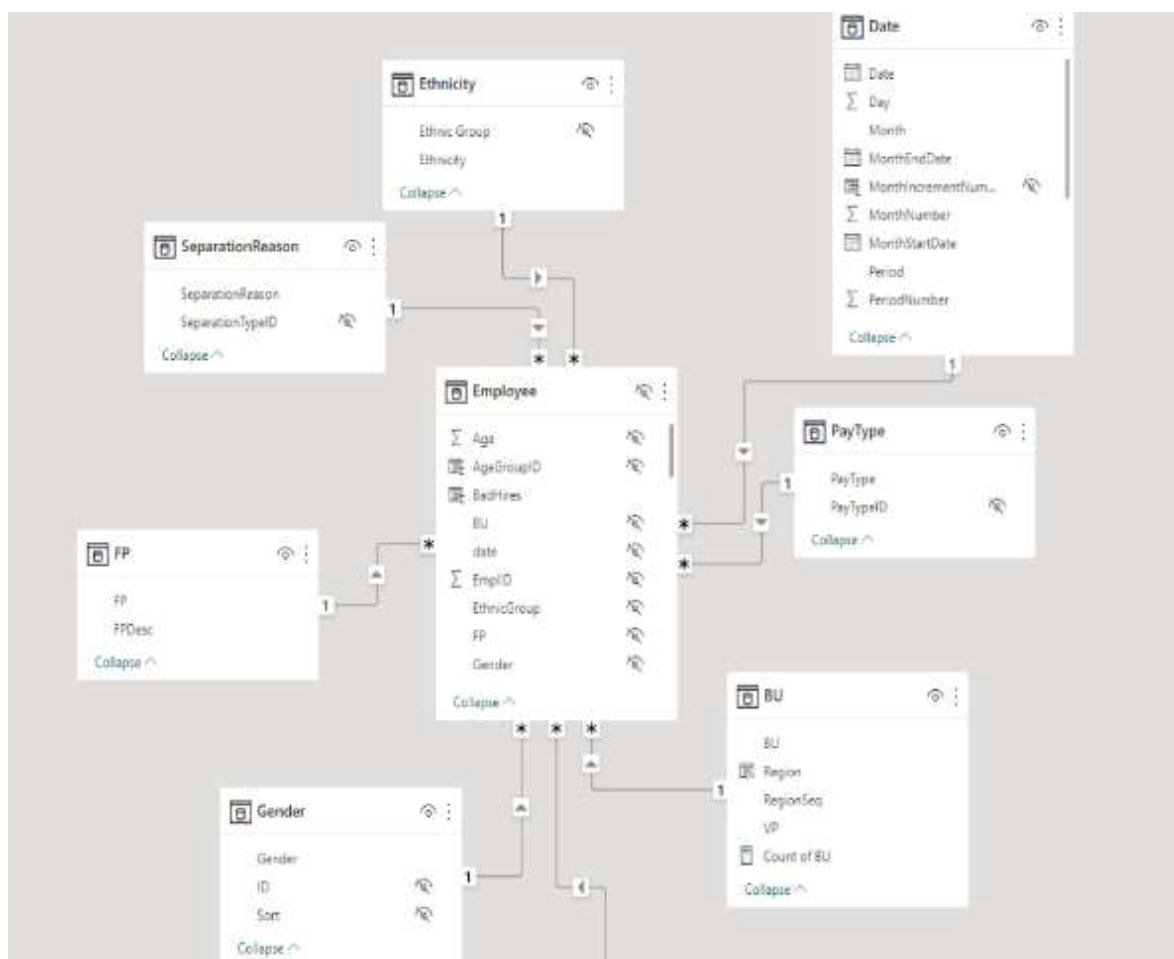


Рисунок 3.16 - Синхронізація з Power Bi

Azure Container Registry. Обираємо Azure Container Registry службу приватного реєстру, що буде відповідати за зберігання та обслуговування образів контейнерів та зв'язаних з ними артефактів. Дана служба використовує команди Docker для відправки образів контейнера в реєстр, після чого вони витягнуться за допомогою Azure Kubernetes service [24].

Наступним модулем буде запускатись Message handler module:

- відправка повідомлень в Milestone [25];
- відправка повідомлень в SQL DB [26];
- відправка письмових оповіщень (email) замовнику.

Останнім буде спрацьовувати модуль Azure Cache for Redis:

Сервіс Azure [27], представляє собою сховище даних у пам'яті (зображення, координати), яке використовується в якості розподіленої бази даних «ключ-значення» в кеш пам'яті.

Oracle Database Service для Microsoft Azure - це керований Oracle сервіс для клієнтів Azure, що забезпечує зручні та прості можливості підготовки, доступу та використання сервісів СУБД Oracle Database корпоративного рівня Oracle Cloud Infrastructure (OCI) за допомогою знайомого користувачам Azure інтерфейсу. Користувачі можуть легко створювати програми Azure за допомогою високопродуктивних, високодоступних і автоматично керованих сервісів СУБД Oracle Database, таких як Autonomous Database, що працюють в OCI.

Даний сервіс надає можливість розробки, виконання програм та зберігання даних на серверах, розташованих у розподілених дата-центрах. Він забезпечує передачу повідомлень (комунікацію) між модулями.

4 РЕЗУЛЬТАТИ РОБОТИ

У цьому розділі розглянемо основні результати роботи системи. По-перше, розглянемо результати роботи системи на наборі зображень, коли відбувалося розпізнавання наявності шолома, а коли окулярів, як далі система розпізнає порушника. По-друге, роздивимося роботу системи в цілому, зокрема зазначимо алгоритм опрацювання зафіксованого порушення.

Подані приклади візуалізують, як ідентифікувати людину в небезпечній зоні, та момент на якому фіксується порушення.

4.1 Результати роботи з датасетом

Точність (Precision) та Повнота (Recall) є двома метриками, які використовуються для оцінки ефективності класифікаційних моделей, включаючи нейронні мережі. Ці метрики особливо важливі у випадках, коли є дисбаланс класів у даних.

Точність визначає, яка частка об'єктів, визначених моделлю як позитивні, дійсно належать до позитивного класу.

Визначається формулою 4.1:

$$\text{Точність} = \frac{K_{\text{п}}}{K_{\text{в}}} \quad (4.1)$$

де $K_{\text{п}}$ – кількість правильно визначених позитивних екземплярів,

$K_{\text{в}}$ – кількість всіх визначених позитивних екземплярів,

Іншими словами, це відношення кількості правильно визначених позитивних екземплярів до загальної кількості визначених позитивних екземплярів.

Повнота (Recall) визначає, яка частка всіх дійсних позитивних об'єктів була правильно визначена моделлю як позитивні.

Визначається формулою 4.2:

$$\text{Повнота} = \frac{K_{\text{п}}}{K_{\text{вд}}} \quad (4.2)$$

де $K_{\text{п}}$ – кількість правильно визначених позитивних екземплярів,

$K_{\text{вд}}$ – кількість всіх дійсних визначених позитивних екземплярів,

Іншими словами, це відношення кількості правильно визначених позитивних екземплярів до загальної кількості дійсних позитивних екземплярів.

Важливо зауважити, що точність та повнота є взаємодоповнюючими метриками, і їх вибір залежить від конкретної задачі та вимог до моделі:

Висока точність важлива, коли важливо уникнути неправильної класифікації негативних прикладів як позитивних.

Висока повнота важлива, коли важливо виявити всі дійсні позитивні об'єкти, навіть за умови, що деякі з них можуть бути помилково визначені як негативні.

У роботі також було проведено дослідження залежності точності розпізнавання обличчя в захисних окулярах та робітничому шоломі в залежності від кута нахилу голови. Наочні результати представлено на рис. 4.2.

Бачимо, що в залежності від кута повороту голови, точність розпізнавання дещо падає, але все ж дає непоганий результат.

Таблиця 4.1 – Валідаційна вибірка для розпізнавання шолому і окулярів

Варіант	Точність (precision)	Повнота (recall)
“helmet” каска	0,983	0,995
“not helmet” немає каски	0,959	0,99
“glasses” окуляри	0,981	0,994
“not glasses” немає окулярів	0,989	0,991

Валідаційну вибірку у більш наочному вигляді представлено на рис. 4.1.

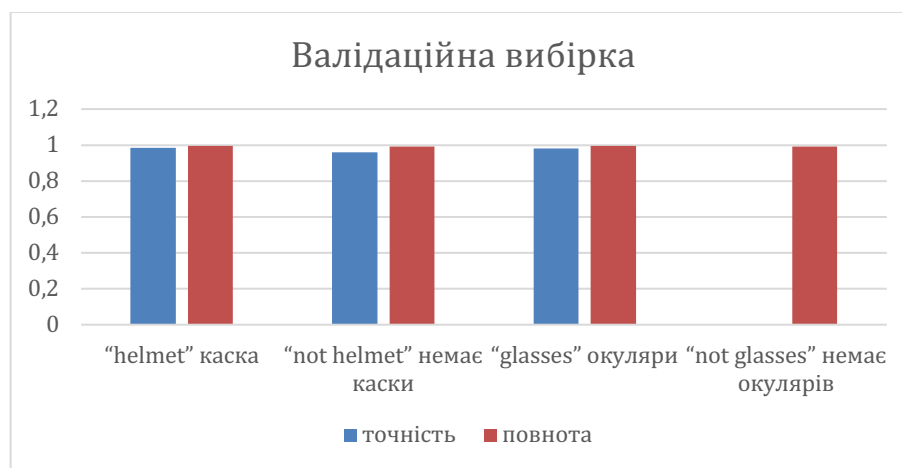


Рисунок 4.1 – Валідаційна вибірка

Бачимо, що в залежності від кута повороту голови, точність розпізнавання дещо падає, але все ж дає непоганий результат.



Рисунок 4.2 – Залежність точності результатів від кута нахилу голови

4.2 Результати роботи системи на прикладах

Отже, процес ідентифікації ЗІЗ виглядає наступним чином.

Кожній камері відведена своя особливо небезпечна зона на підприємстві. На рис. 4.3 в лівому верхньому куті, можна побачити номер унікальний камери відеоспостереження, кожен номер відповідає за свою зону. Зона з особливо небезпечними умовами, виділена зеленим кольором (рис.4.3-4.6). Червоним кольором виділено особу, що знаходиться в межах зони з особливо небезпечними умовами (див. рис.4.3, 4.4, 4.6). Знаходячись в межах (поряд) даної небезпечної зони, необхідно дотримуватись правил охорони праці, та обов'язково використовувати такі ЗІЗ, як робітничка каска та захисні окуляри. Працівник, що попадає в зону відеоспостереження, наділяється унікальним id номером, і якщо буде ідентифіковано подію – порушення носіння ЗІЗ, то фіксується кадр з обличчям порушника. На рис.

4.5 особа яка потрапила в межу небезпечної зони, виділена синім кольором. Дана особа фіксується і передається на обробку НМ на предмет ідентифікації ЗІЗ.



Рисунок 4.3 – Приклад 1

Далі зафіксоване обличчя оброблюється нейронною мережею для розпізнавання. Після того, як обличчя порушника буде ідентифіковано і визначено його ПІБ та посада, за допомогою чат-боту відбувається розсилка повідомлень на відповідальних за дану ділянку на підприємстві: начальнику зміни з охорони праці, безпосередньому керівнику порушника.



Рисунок 4.4 – Приклад 2

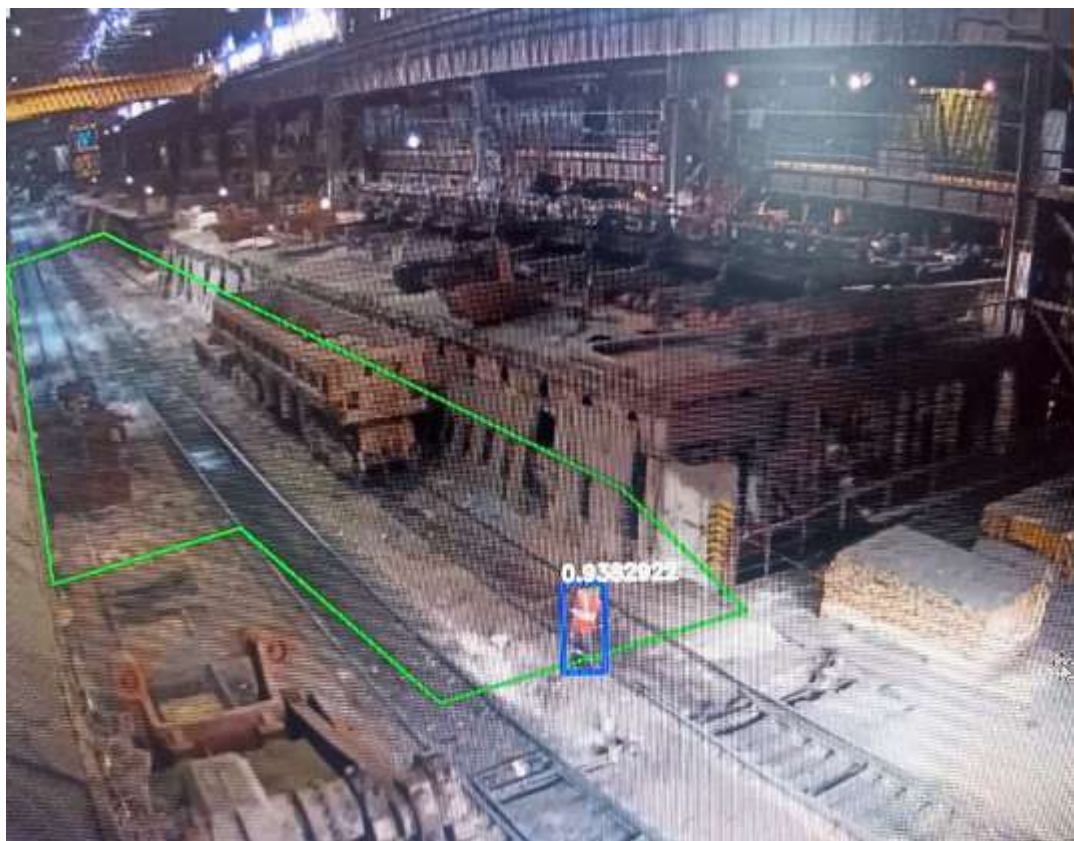


Рисунок 4.5 – Приклад 3

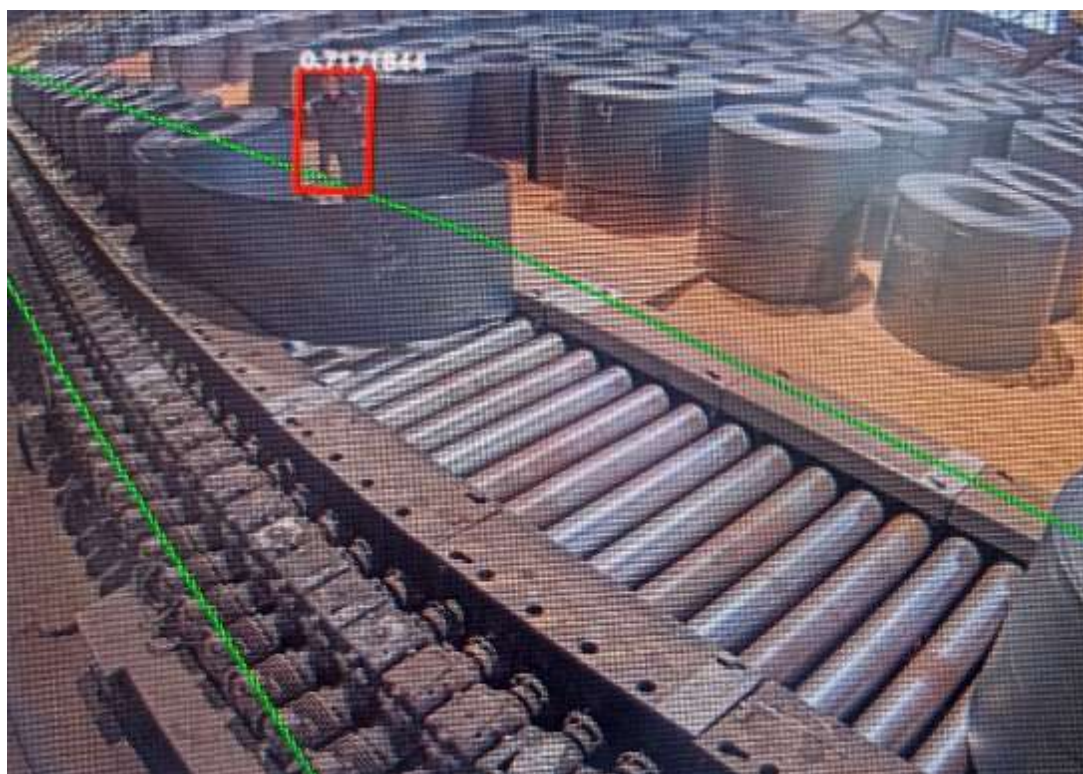


Рисунок 4.6 - Приклад 4

Після отримання повідомлення, через гучномовець, що планується за проектом встановити в зоні особливо небезпечної ділянки підприємства, порушника сповіщають про фіксацію порушення носіння ЗІЗ.

Порушення фіксуються у базі даних порушників. За перше порушення, з заробітної плати працівника вираховують – 15%, за кожне наступне порушення -50%. Якщо охорону праці порушено більше трьох разів працівника відлучають від роботи. Порушник зобов'язаний пройти повторний курс навчання з охорони праці на підприємстві і тільки після успішної здачі екзамену його буде допущено до роботи. Якщо порушення охорони праці відбудеться вчетверте, працівника буде звільнено.

ВИСНОВКИ

Через порушення техніки безпеки підприємства та заводи зазнають багатомільйонних збитків, якщо працівники зазнають травм.

Таким чином, у магістерській роботі розроблено систему, яка дозволяє підвищити рівень безпеки робітників на підприємствах за рахунок розробки системи ідентифікації порушень використання засобів індивідуального захисту.

Для досягнення поставленої мети було проведено:

- аналіз існуючих методів та систем розпізнавання образів;
- аналіз можливих методів та рішень реалізації поставленої задачі, на основі якого переваги та недоліки розглянутих методів;
- пошук аналогічних до розроблюваної систем, але оскільки не було знайдено такі, то як аналог розглянуто систему виявлення захисної маски на обличчі;
- спроєктовано систему та наведено блок-схему алгоритму роботи інтелектуальної системи ідентифікації порушень використання засобів індивідуального захисту;
- створено датасет з захисних окулярів і робітничих шоломів;
- навчено нейронну мережу для розпізнавання відсутності ЗІЗ;
- спроєктовано архітектуру для реалізації проєкту на підприємстві в майбутньому.

У роботі вирішено задачу, як ідентифікувати такі порушення автоматично та якомога раніше, через ідентифікацію ЗІЗ на відео та визначенням людей, що порушили техніку безпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Законодавство України. Постанова Кабінету Міністрів від 27 серпня 2008 р. N 761 про затвердження Технічного регламенту засобів індивідуального захисту відповідно до статті 14 Закону України "Про стандарти, технічні регламенти та процедури оцінки відповідності" (3164–15). URL: <http://zakon2.rada.gov.ua/laws/show/761-2008-%D0%BF>
2. Охорона праці та промислова безпека. Монографія /К. Н. Ткачук, Л. Д. Третьякова, Д. В. Зеркалов, О. І. Полукаров, С. Ф. Каштанов, / – К.: «Основа». 2014. – 823 с
3. Засоби індивідуального захисту голови. URL: <https://euroservis.com.ua>
4. Вимоги до захисних окулярів. URL: pro-op.com
5. Класифікація ЗІЗ. URL: <https://euroservis.com.ua/ua/sredstva-individualnoy-zashchity-raznovidnosti-i-klassifikatsiya/>
6. Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1600–1609, 2015. 16
7. Y. Wang, X. Ji, Z. Zhou, H. Wang, and Z. Li. Detecting faces using region-based fully convolutional networks. arXiv preprint arXiv:1709.05256, 2017. 31
8. Khan A. I. Al-Habsi S. Machine Learning in Computer Vision, 2020. – [Електроний ресурс]. Режим доступу: <https://www.sciencedirect.com/science/article/pii/S1877050920308218?via%3Dihub> – 10.10.2023.
9. Du Y., Wang W., Wang L. Hierarchical recurrent neural network for skeleton based action recognition. // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015. – Vol. 7. – No. 12. – P. 1110–1118
10. Image-based Face Recognition – Issues and Methods. URL: https://www.face-rec.org/interestingpapers/general/chapter_figure.pdf
11. Торстен, А Відеоаналітика: Міфи і реальність / А. Торстен, К. Іво, Л. Харальд // Security Focus – 2012. – с 111-116.

12. . Методи, алгоритми і програмні засоби опрацювання біомедичних зображень / Березький О. М., Батько Ю.М., Березька К.М. і ін. Тернопіль: Економічна думка, ТНЕУ, 2017, 330 с.
13. Саймон Хайкін. Нейронні мережі. Повний курс. – М.: Вил'ямс, 2017 – 1104 с
14. H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, “RMPE: Regional multi-person pose estimation,” in ICCV, 2017
15. Kiranyaz, S., Ince, T., Iosifidis, A., Gabbouj, M. Operational neural networks. Neural Computing and Applications, 2020. URL: <https://doi.org/10.1007/s00521-020-04780-3> – 19.10.2023
16. Toshev A., Szegedy C. Deeppose: Human pose estimation via deep neural networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2014. – с. 1653—1660.
17. Веретельник, Д.М. Детекція і розпізнавання осіб з використанням відкритого комп'ютерного зору OpenCV / Д.М. Веретельник, Е.А. Норкене, Д.М. Бочаров // Інформатика, управляючі системи, математичне та комп'ютерне моделювання, 2020. – с. 325-330
18. Веретельник, Д.М. Методології детекції людського обличчя на зображеннях / Сучасні інформаційні технології в освіті і наукових дослідженнях. 2019. – с. 143-149
19. Open Source Computer Vision URL: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html (дата звернення: 14.10.2023)
20. dlib C++ Library.URL: <http://dlib.net/> (дата звернення: 24.11.2023)
21. Ageitgey. Face_recognition. URL: https://github.com/ageitgey/face_recognition (дата звернення: 28.11.2023)
22. R. Girshick, J. Donahue, T. Darrell, and J. Malik. «Region-based convolutional networks for accurate object detection and segmentation.» TPAMI, 2015

23. S. Ren, K. He, R. Girshick, and J. Sun, «Faster R-CNN: Towards real-time object detection with region proposal networks,» in Neural Information Processing Systems (NIPS), 2015.
24. Microsoft Build URL: <https://learn.microsoft.com/ua-ua/azure/architecture/reference-architectures/containers/aks/baseline-aks> (дата звернення: 28.11.2023)
25. Milestone keep RTSP Stream alive URL: <https://developer.milestonesys.com/s/question/0D53X00006tRyVjSAK/how-milestone-keep-rtsp-stream-alive/> (дата звернення: 17.11.2023)
26. My SQL URL: <https://www.mysql.com/> (дата звернення: 15.11.2023)
27. AZURE URL: <https://azure.microsoft.com/en-us/> (дата звернення: 30.11.2023)
28. Рибаченко Н.А.Інтелектуальна система ідентифікації порушень використання засобів індивідуального захисту / Н.А. Рибаченко, М.Ю. Тягунова // Тиждень науки-2023. Факультет комп'ютерних наук і технологій : наук.-техн. конф., 24-28 квітня 2023 р. : тези доп. – / Редкол. : Вадим ШАЛОМЄЄВ (відпов. ред.) Електрон. дані. – Запоріжжя : НУ «Запорізька політехніка», 2023. – С. 81-82 – 1 електрон. опт. диск (DVD-ROM); 12 см. – назва з тит. екрана.

ДОДАТОК А

Лістинг А.1 – Вихідний код процесу навчання і перевірки моделей

```
import tensorflow as tf from tensorflow import keras from sklearn.preprocessing
import LabelBinarizer from sklearn.model_selection import train_test_split from
sklearn.metrics import classification_report # Допоміжні бібліотеки:
import numpy as np import matplotlib import matplotlib.pyplot as plt # ініціалізуємо
початкову швидкість навчання, кількість епох тренування:
INIT_LR = 1e-4 EPOCHS = 10 BS = 32 #
Список даних (зображень) та зображень класів:
print("[INFO] loading images...") imagePath =
list(paths.list_images(args["dataset"])) data = [] labels = [] # #
Перебираємо шляхи до зображень:
for imagePath in imagePath: # #
Вилучаємо мітку класа з імені файла:
label = imagePath.split(os.path.sep)[-2] # #
Завантажуємо вихідне зображення: (224x224) та обробляємо його:
image = load_img(imagePath, target_size=(224, 224)) image = img_to_array(image) image
= preprocess_input(image) 119 #
Онвлюємо списки даних і міток відповідно:
data.append(image) labels.append(label) #
ПЕРетворюємо дані і мітки в масиви за допомогою:
NumPy data = np.array(data, dtype="float32") labels = np.array(labels) model =
Sequential([ Conv2D(100, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2), Conv2D(100, (3, 3), activation='relu'), MaxPooling2D(2, 2),
Flatten(), #
Згладжуємо ввід:
Dropout(0.5), Dense(50, activation='relu'), Dense(2, activation='softmax') ])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc']) #
Генеруємо дані для навчання:
TRAINING_DIR = "./train" train_datagen = ImageDataGenerator(rescale=1.0/255,
rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True, fill_mode='nearest') 120 train_generator =
train_datagen.flow_from_directory (TRAINING_DIR, batch_size=10, target_size=(150,
150)) VALIDATION_DIR = "./test" validation_datagen =
ImageDataGenerator(rescale=1.0/255) validation_generator =
validation_datagen.flow_from_directory(VALIDATION_DIR, batch_size=10,
target_size=(150, 150)) #
```

Створюємо контрольну точку зворотнього виклику, щоб зберегти найкращу модель після кожної епохи під час навчання:

```
checkpoint = ModelCheckpoint('model2-{epoch:03d}.model',
monitor='val_loss',verbose=0,save_best_only=True,mode='auto') #
```

Навчання моделі:

```
history = model.fit_generator(train_generator, epochs=10,
validation_data=validation_generator, callbacks=[checkpoint])
```

Візуалізація результатів навчання по епохам:

```
Epoch 1/10 132/132 [=====] - 171s 1s/step - loss: 0.6702 - a
cc: 0.7141 - val_loss: 0.1702 - val_acc: 0.9588
Epoch 2/10 132/132 [=====] - 167s 1s/step - loss: 0.3395 - a
cc: 0.8730 - val_loss: 0.0491 - val_acc: 0.9072 121
Epoch 3/10 132/132 [=====] - 166s 1s/step - loss: 0.3382 - a
cc: 0.8852 - val_loss: 0.2210 - val_acc: 0.9021
Epoch 4/10 132/132 [=====] - 166s 1s/step - loss: 0.2634 - a
cc: 0.8958 - val_loss: 0.0169 - val_acc: 0.9742
Epoch 5/10 132/132 [=====] - 171s 1s/step - loss: 0.2149 - a
cc: 0.9118 - val_loss: 0.1482 - val_acc: 0.9588
Epoch 6/10 132/132 [=====] - 166s 1s/step - loss: 0.2057 - a
cc: 0.9278 - val_loss: 0.0036 - val_acc: 0.9691
Epoch 7/10 132/132 [=====] - 166s 1s/step - loss: 0.2048 - a
cc: 0.9346 - val_loss: 0.0698 - val_acc: 0.9588
Epoch 8/10 132/132 [=====] - 166s 1s/step - loss: 0.1803 - a
cc: 0.9346 - val_loss: 0.0073 - val_acc: 0.9845
Epoch 9/10 132/132 [=====] - 178s 1s/step - loss: 0.1570 - a
cc: 0.9414 - val_loss: 0.0013 - val_acc: 0.9845
Epoch 10/10 132/132 [=====] - 167s 1s/step - loss: 0.1334 -
a cc: 0.9536 - val_loss: 0.0335 - val_acc: 0.9536
```

В процесі навчання моделі відображаються метрики втрат (loss) і точності (accuracy). Дана модель досягає на тренувальних даних точності, яка дорівнює приблизно 0.98 (98%).

Далі протестуємо результати моделі знаходження окулярів на обличчі за допомогою відеопотоку, що отримуємо з камер:

```
import cv2 import numpy as np from keras.models import load_model
model=load_model("./model-09") results={0:'without glasses',1:' glasses'}
GR_dict={0:(0,0,255),1:(0,255,0)} rect_size = 4 cap = cv2.VideoCapture(0) haarcascade
=
cv2.CascadeClassifier('C:/Users/Ansor/anaconda3/envs/LABML/Lib/sitepackages/cv2/data/
haarcascade_frontalface_default.xml') while True: (rval, im) = cap.read() 122
im=cv2.flip(im,1,1) rrect_size = cv2.resize(im, (im.shape[1] // rect_size,
im.shape[0] // rect_size)) faces = haarcascade.detectMultiScale(rrect_size) for f in
faces: (x, y, w, h) = [v * rect_size for v in f] face_img = im[y:y+h, x:x+w]
```

```

rerect_sized=cv2.resize(face_img,(150,150)) normalized=rerect_sized/255.0
reshaped=np.reshape(normalized,(1,150,150,3)) reshaped = np.vstack([reshaped])
result=model.predict(reshaped) label=np.argmax(result,axis=1)[0]
cv2.rectangle(im,(x,y),(x+w,y+h),GR_dict[label],2) cv2.rectangle(im,(x,y-
40),(x+w,y),GR_dict[label],-1) cv2.putText(im, results[label], (x,
y10),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2) cv2.imshow('LIVE', im) key =
cv2.waitKey(10) if key == 27: break cap.release() cv2.destroyAllWindows(
Далі протестуємо результати моделі знаходження каски на обличчі за допомогою
відеопотоку, що отримуємо з камер:
import cv2 import numpy as np from keras.models import load_model
model=load_model("./model-09") results={0:'without helmet',1:' helmet'}
GR_dict={0:(0,0,255),1:(0,255,0)} rect_size = 4 cap = cv2.VideoCapture(0) haarcascade
=
cv2.CascadeClassifier('C:/Users/Ansor/anaconda3/envs/LABML/Lib/sitepackages/cv2/data/
haarcascade_frontalface_default.xml') while True: (rval, im) = cap.read() 122
im=cv2.flip(im,1,1) rerect_size = cv2.resize(im, (im.shape[1] // rect_size,
im.shape[0] // rect_size)) faces = haarcascade.detectMultiScale(rerect_size) for f in
faces: (x, y, w, h) = [v * rect_size for v in f] face_img = im[y:y+h, x:x+w]
rerect_sized=cv2.resize(face_img,(150,150)) normalized=rerect_sized/255.0
reshaped=np.reshape(normalized,(1,150,150,3)) reshaped = np.vstack([reshaped])
result=model.predict(reshaped) label=np.argmax(result,axis=1)[0]
cv2.rectangle(im,(x,y),(x+w,y+h),GR_dict[label],2) cv2.rectangle(im,(x,y-
40),(x+w,y),GR_dict[label],-1) cv2.putText(im, results[label], (x, y10),
cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2) cv2.imshow('LIVE', im) key =
cv2.waitKey(10) if key == 27: break cap.release() cv2.destroyAllWindows

```