

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ВИРОБНИЦТВІ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
OBJECT RECOGNITION AT PRODUCTION SITE

Виконав: студент 2 курсу, групи КНТ-129м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Шемберко Ю.А.

(прізвище та ініціали)

Керівник Терещенко Е.В.

(прізвище та ініціали)

Рецензент Неласа Г. В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (КОД І НАЙМЕНУВАННЯ)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Шемберко Юрію Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів розпізнавання об'єктів на виробництві

Research and Software Implementation of Methods for Object Recognition at Production Site.

керівник проєкту (роботи) Терещенко Еліна Валентинівна, к.ф.-м.н., доцент,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року
 № 301

2. Строк подання студентом проєкту (роботи) 14 грудня 2020 року

3. Вихідні дані до проєкту (роботи) навчальна та тестова вибірка зображень для навчання моделі розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1 Аналіз проблеми та постановка завдань дослідження.

2 Дослідження інформаційної технології розпізнавання об'єктів на виробництві.

3 Програмна реалізація системи розпізнавання об'єктів на виробництві

4 Експериментальне дослідження розробленої системи

5 Економіко-організаційна частина

6 Охорона праці та безпека в надзвичайних ситуаціях.

Висновки. Додатки.

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

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-6 Основна частина	Терещенко Е.В., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та цивільна безпека	Коробко О.В., ст. викладач		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “ 4 ” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Тестування та експериментальне дослідження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка економіко-організаційної частини	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та цивільної безпеки.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент

(підпис)

Шемберко Ю.А.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Терещенко Е.В.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
93 с., 10 табл., 22 рис., 2 дод., 35 джерел.

МАШИННЕ НАВЧАННЯ, КОМП'ЮТЕРНИЙ ЗІР, СИСТЕМА
МАШИННОГО ЗОРУ, РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, OPENCV.

Об'єкт дослідження – процеси розпізнавання об'єктів у відеопотоці.

Предмет дослідження – методи та системи розпізнавання об'єктів у відеопотоці.

Мета роботи – розробка системи машинного зору, що дозволяє виконувати розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках з високим ступенем достовірності розпізнавання.

Матеріали, методи та технічні засоби – мова програмування Python, для обробки зображень: компонент Image Processing, для відтворення користувацького інтерфейсу: компонент High-level GUI, для розпізнавання об'єктів: компонент Object Detection бібліотеки OpenCV.

Результати. Розроблено систему машинного зору для розпізнавання об'єктів заданого класу в реальному часі у відеопотоці з високою достовірністю розпізнавання за рахунок використання методу Віоли-Джонса та побудови каскаду класифікаторів, навчених з допомогою алгоритму AdaBoost.

Висновки. Створено систему машинного зору, що вирішує поставлені задачі.

Галузь використання – розроблене програмне забезпечення може бути використано в якості модуля для автоматизованої системи контролю заходів безпеки на промислових підприємствах та виробництві.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 93 pages, 10 tables, 22 figures, 2 appendixes, 35 sources.

MACHINE LEARNING, COMPUTER VISION, MACHINE VISION SYSTEM, OBJECT RECOGNITION, OPENCV

Object of study – the object recognition processes in the video stream.

Subject of study – methods and systems for recognizing objects in the video stream.

The purpose of the work is development of a machine vision system that allows the recognition of objects of a given class: mask, helmet and goggles in a person working in production in real time in video streams with a high degree of recognition.

Design tools – Python programming language, for image processing: Image Processing component, for user interface design: High-level GUI component, for object recognition: Object Detection component of the OpenCV library .

Results. A machine vision system for real-time object recognition of a given class in a video stream with high recognition reliability has been developed by using the Viola-Jones method and constructing a cascade of classifiers trained using the AdaBoost algorithm.

Conclusion. The machine vision system has been created that solves the given set of problems.

Field of use – the developed software can be used as a module for the automated control system of safety measures at industrial enterprises and production.

ЗМІСТ

	С.
РЕФЕРАТ	4
ABSTRACT	5
Зміст	6
Перелік скорочень та умовних познач	9
Вступ	10
1 Аналіз проблеми та постановка завдань дослідження	12
1.1 Постановка задачі розпізнавання об'єктів у відеопотоці	12
1.2 Дослідження підходу скануючих вікон для розпізнавання об'єктів у відеопотоці	13
1.3 Дослідження методів розпізнавання об'єктів у відеопотоці	14
1.3.1 Алгоритм адаптивного бустингу AdaBoost	15
1.3.2 Алгоритм градієнтного бустингу AnyBoost	16
1.3.3 Метод Віюли-Джонса	17
1.3.4 Методи розпізнавання на основі штучних нейронних мереж	18
1.4 Дослідження програмних засобів для розпізнавання об'єктів у відеопотоці	19
1.4.1 Xeoma Mask Detector PRO	19
1.4.2 Aerialtronics Face Mask Detection Software	20
1.4.3 Hikvision Mask Detection	20
1.5 Висновки з розділу 1	21
2 Дослідження інформаційної технології розпізнавання об'єктів на виробництві	23
2.1 Опис технології розпізнавання об'єктів у відеопотоці	23
2.2 Дослідження програмних засобів розробки системи	25
2.2.1 Вибір мови програмування	25

2.2.2 Вибір бібліотеки для розробки системи з використанням алгоритмів комп'ютерного зору.....	27
2.3 Висновки з розділу 2	29
3 Програмна реалізація системи розпізнавання об'єктів на виробництві	30
3.1 Опис основних рішень щодо реалізації системи.....	30
3.2 Основні рішення щодо модульного представлення системи.....	30
3.3 Опис особливостей реалізованих алгоритмів.....	35
3.3.1 Опис процесу тренування моделі (навчання)	35
3.3.2 Опис процесу розпізнавання.....	36
3.4 Висновки з розділу 3	38
4 Експериментальне дослідження розробленої системи	39
4.1 Функціональні можливості системи.....	39
4.2 Порівняльний аналіз функціональних можливостей системи з відомими аналогами.....	41
4.3 Оцінка показника достовірності розпізнавання	42
4.5 Висновки з розділу 4	43
5 Економіко-організаційна частина.....	45
5.1 Планування розробки програмного виробу.....	45
5.2 Визначення витрат на розробку	48
5.2.1 Розрахунок основної та додаткової заробітної плати	48
5.2.2 Відрахування на єдиний соціальний внесок	49
5.2.3 Визначення витрат на матеріали	49
5.2.4 Витрати на спеціальне обладнання	50
5.2.5 Інші прямі витрати	54
5.2.6 Розрахунок накладних витрат.....	54
5.3 Оцінка економічної ефективності.....	55
6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	56
6.1 Аналіз потенційних небезпек	56

6.2 Заходи із забезпечення безпеки.....	57
6.3.Заходи з виробничої санітарії та гігієни праці	60
6.4 Заходи безпеки у надзвичайних ситуаціях	65
6.4.1 Заходи з пожежної безпеки	65
6.4.2 Заходи з цивільного захисту	68
Висновки	72
Перелік джерел посилання	74
Додаток А Текст програми.....	78
Додаток Б Слайди презентації	86

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

API	– Application Programming Interface;
АПС	– автоматична пожежна сигналізація;
ПЗ	– програмне забезпечення;
ПЗВ	– пристрій захисного відключення;
СТСН	– синдром тривалих статичних навантажень.

ВСТУП

Актуальність теми. Пандемія COVID-19 вплинула на існування та розвиток суспільства, на умови соціального життя, навчання, на умови виробництва.

Карантинні заходи вимагають використання систем розпізнавання в місцях великого скупчення людей: в торгових центрах, вокзалах, аеропортах, громадському транспорті, виробництві. Насьогодні найбільш популярними є такі системи розпізнавання обличчя в масках та без масок: Xeoma Mask Detector PRO, Aerialtronics Face Mask Detection Software, Hikvision Mask Detection. Ці системи мають певні недоліки: високу ціну, складність в використанні, необхідність потужних апаратних ресурсів, недостатню продуктивність та достовірність результатів в умовах браку освітленості, й, головне, неможливість використання цих систем в умовах виробництва.

Для умов виробництва було поставлено задачу по розробці програмного забезпечення, що дозволяє розпізнавати наявність об'єктів певного заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках.

Мета і задачі дослідження. Метою кваліфікаційної роботи магістра є розробка системи машинного зору, що дозволяє виконувати розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках з високим ступенем достовірності розпізнавання.

Для досягнення поставленої мети потрібно вирішити наступні задачі:

- виконати дослідження методів розпізнавання об'єктів у відеопотоці;
- виконати дослідження програмних засобів для розпізнавання об'єктів у відеопотоці;
- розробити систему розпізнавання об'єктів заданого класу у відеопотоці;
- виконати експериментальне дослідження розробленої системи.

Об'єкт дослідження – процеси розпізнавання об'єктів у відеопотоці.

Предмет дослідження – методи та системи розпізнавання об'єктів у відеопотоці.

Методи дослідження. Для розробки системи машинного зору застосовано методи теорії комп'ютерного зору та інтелектуального аналізу даних.

Наукова новизна роботи полягає в тому, що розроблено систему машинного зору для розпізнавання об'єктів заданого класу в реальному часі у відеопотоці з високою достовірністю розпізнавання за рахунок використання методу Віюлі-Джонса та побудови каскаду класифікаторів, навчених з допомогою алгоритму AdaBoost.

Практична цінність результатів роботи полягає в тому, що розроблене програмне забезпечення може бути використано як модуль для розробки автоматизованої системи контролю заходів безпеки на промислових підприємствах та виробництві.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Постановка задачі розпізнавання об'єктів у відеопотоці

На цей час в галузі машинного зору існують дві окремі задачі: виявлення об'єктів заданого класу на зображенні та розпізнавання певних об'єктів на послідовності зображень (або відеозаписі) [1].

Виявлення об'єктів на відеозаписі можна звести до задачі виявлення об'єктів певного класу на статичному зображенні. При цьому, як правило, завдання виявлення об'єктів розглядають як задачу бінарної класифікації: навчений на тестових даних класифікатор приймає рішення про належність зображення до шуканого класу [1].

Часто використовується підхід вікна ковзання або підхід скануючого вікна. При використанні такого підходу, класифікатор аналізує не все зображення, а різні його частини, як правило прямокутної чи квадратної форми. Послідовно обираються різні частини зображення різного розміру та з різним положенням на вихідному зображенні. Таким чином можливо визначати об'єкти, що мають різний розмір та можуть розміщуватися у довільній частині кадру [1].

При цьому, алгоритм виявлення повинен мати достатню швидкодію, щоб мати змогу виявляти шукані об'єкти за короткий проміжок часу між послідовними кадрами (приблизно 1/30 секунди або менше).

Задача розпізнавання об'єктів на відеозаписі невідривна від задачі виявлення об'єктів, бо для розпізнавання об'єкту треба вміти відрізнити його від фону та знаходити певні характеристичні точки, за якими потім проводиться розпізнавання [1].

Задача розпізнавання об'єктів на відеозаписі може бути розділена на дві підзадачі :

- розпізнавання довільних об'єктів, що задаються “онлайн”, тобто у процесі роботи алгоритму;
- виявлення об'єктів, що належать до заздалегідь відомого класу [1].

У першому випадку, алгоритми розпізнавання мають більшу гнучкість у роботі, бо не обмежуються певним набором класів, а можуть визначати довільні об'єкти без перенавчання. Недоліком такого підходу є відносно невисока точність відстеження. Навчання трекеру (від англ. track - відстежувати) відбувається безпосередньо у процесі обробки відеозапису, й тому відсутні якісні матеріали для навчання алгоритму визначення об'єктів. Алгоритми цього класу можуть часто "втрачати" об'єкти, що відстежуються, та давати велику кількість хибнопозитивних результатів [1].

Для виявлення об'єктів певного заздалегідь відомого класу застосовується попередньо навчений класифікатор. Такі алгоритми мають більш високий показник точності та достовірності визначення шуканих об'єктів [1].

1.2 Дослідження підходу скануючих вікон для розпізнавання об'єктів у відеопотоці

Випадки, коли шуканий об'єкт займає усю площу зображення, на якому проводиться пошук, є досить нечастими. Навпаки, у більшості випадків шуканий об'єкт може бути будь-якого довільного розміру та бути розташованим у різних частинах зображень [1-2].

Для того, щоб мати змогу виявляти об'єкти у таких умовах, найчастіше застосовують підход скануючого вікна.

Вікном у такому випадку є ділянка вихідного зображення, яка передається на вхід класифікатора. Після цього кроку можна отримати відповідь, чи знаходиться шуканий об'єкт в цій частині зображення [1-2].

Для найбільш ефективного пошуку скануюче вікно варіюють у розмірах у процесі пошуку. Також обирається оптимальний крок для зміщення вікна, який дозволяє послідовно проаналізувати усю площу зображення. При цьому довжину кроку обирають таку, щоб кожне наступне вікно перекривало частину попереднього (як правило, не менше половини площі) [1-2].

Цей метод є досить простим, але добре підходить для локалізації та виявлення об'єктів незалежно від їх масштабу та положення на вихідному зображенні.

Проте, можуть виникати проблеми з пошуком об'єктів, пропорції обмежуючих прямокутників яких сильно відрізняються від пропорцій прямокутника скануючого вікна. Подолати цей недолік можна налаштуванням пропорцій вікна для пошуку або змінюючи його пропорції безпосередньо під час процесу пошуку [1-2].

Також, при використанні скануючого вікна можлива ситуація, коли один і той самий об'єкт буде захоплений у декілька різних близько розташованих вікон і, таким чином, порахований декілька разів. Цей недолік може бути усунутий “пригніченням немаксимальних вікон”: серед усіх перекриваючих одне одного вікон обираються вікна з локальним максимумом класифікатора [1-2].

1.3 Дослідження методів розпізнавання об'єктів у відеопотоці

На цей час існує багато методів розпізнавання об'єктів у відеопотоці, що мають свої переваги та недоліки [3-8]. Наприклад:

- алгоритм адаптивного бустингу AdaBoost;
- алгоритм градієнтного бустингу AnyBoost;
- метод Віоли-Джонса;
- методи розпізнавання на основі штучних нейронних мереж.

Розглянемо більш детально їхні характеристики.

1.3.1 Алгоритм адаптивного бустингу AdaBoost

Алгоритм адаптивного бустингу AdaBoost [3-4] – алгоритм посилення класифікаторів, шляхом об'єднання їх в множину, запропонований Йоавом Фройндом і Робертом Шапіре. Алгоритм AdaBoost є адаптивним в тому сенсі, що кожна наступна множина класифікаторів будується по об'єктах, невірно класифікованих попередньою множиною. Алгоритм AdaBoost чутливий до шуму в даних і викидів. Однак він менш схильний до перенавчання, в порівнянні з іншими алгоритмами машинного навчання.

Алгоритм AdaBoost викликає слабкі класифікатори в циклі. Після кожного виклику оновлюється розподіл ваг, які відповідають важливості кожного з об'єктів навчальної множини зразків для класифікації. На кожній ітерації ваги кожного невірно класифікованого об'єкта зростають, таким чином нова множина класифікаторів «фокусує свою увагу» на цих об'єктах.

Наведемо детальний опис фази навчання алгоритму AdaBoost [3-4].

Дано: навчаючі зразки $(x_1, y_1), \dots, (x_m, y_m)$, де x_i – зображення, y_i – клас зображення.

Крок 1. Ініціалізуємо масив ваг: $w_i = \frac{0.5}{m_{y_i}}$, де m_0 та m_1 – кількість зразків першого та другого типу відповідно.

Крок 2. Для кожного ще не обраного слабкого класифікатора C_j знаходимо помилку класифікації за формулою: $err_j = \sum_i w_i \cdot |C_j(x_i) - y_i|$, тобто помилка класифікації дорівнює сумі ваг неправильно класифікованих зразків.

Крок 3. Серед усіх слабких класифікаторів оберемо класифікатор C_t із найменшою сумою помилки та додамо його до шуканої множини класифікаторів з вагою $\text{cweight}_t = \log\left(\frac{1}{B_t}\right)$, де $B_t = \frac{err_t}{1 - err_t}$.

Крок 4. Оновлюємо вагові коефіцієнти тих зразків, що неправильно класифікуються класифікатором C_t за формулою $w_i = w_i \cdot B_t$.

Крок 5. Нормуємо вагові коефіцієнти так, щоб $\sum w_i = 1.0$.

Крок 6. Якщо обраних класифікаторів недостатньо, повертаємось до кроку 2.

Крок 7. Повертаємо множину обраних класифікаторів разом.

Результуючий класифікатор представлений формулою (1.1):

$$S(x) = \begin{cases} 1: & \sum_{t=1}^T cweight_t \cdot C_t(x) \geq 0.5 \cdot \sum_{t=1}^T cweight_t \\ 0: & otherwise \end{cases} \quad (1.1)$$

1.3.2 Алгоритм градієнтного бустингу AnyBoost

Алгоритм градієнтного бустингу AnyBoost [5] – алгоритм подібний до алгоритму AdaBoost, але на відміну від AdaBoost, що змінює ваги на кожній ітерації, алгоритм AnyBoost дозволяє або намагається навчати нові моделі за залишковою помилкою попередніх ітерацій (рухаючись до мінімуму функції втрат). Алгоритм AnyBoost оснований на роботі градієнтного спуску, принцип якого полягає у тому, щоб підібрати коефіцієнт, при якому функція втрат або різниця між істинною відповіддю та прогнозом моделі стала мінімально, тобто прогноз максимально наблизився до істинного значення [5].

Наведемо опис кроків алгоритму AnyBoost [5].

Крок 1. Модель будується по збірці даних.

Крок 2. Ця модель робить передбачення для всього набору даних.

Крок 3. За прогнозами та істинним значенням обчислюються помилки.

Крок 4. Нова модель будується з урахуванням помилок як цільових змінних. При цьому потрібно знайти краще розділення для мінімізації помилки.

Крок 5. Прогнози, зроблені за допомогою цієї нової моделі, поєднуються з передбаченнями попередніх.

Крок 6. Знову обчислюються помилки з використанням цих передбачених значень та істинних значень.

Крок 7. Цей процес повторюється, поки функція помилки не перестане змінюватися або поки не буде досягнуто максимальної кількості предикторів.

1.3.3 Метод Віоли-Джонса

Метод Віоли-Джонса – це один з традиційних методів машинного навчання для пошуку об'єктів на зображеннях. Цей метод може бути застосований для виявлення будь-яких класів об'єктів, але при створенні методу автори орієнтувалися в першу чергу на задачу виявлення обличч

Алгоритм, запропонований авторами цього методу, використовує поєднання простих характеристичних рис дерева рішень за допомогою методу AdaBoost та побудову каскаду із отриманих таким чином класифікаторів для досягнення високої швидкості та точності виявлення [6-7].

Важливо зауважити, що метод Віоли-Джонса може бути використаний лише для виявлення об'єктів (тобто, бінарної класифікації зображень), але не для їх розпізнавання.

Алгоритм роботи методу складається з наступних кроків [6-7].

Крок 1. Вибір характеристичних рис Хаара.

Крок 2. Побудова інтегральної форми зображення.

Крок 3. Тренування класифікаторів за допомогою алгоритму AdaBoost.

Крок 4. Побудова каскаду класифікаторів.

Основними перевагами методу Віоли-Джонса є:

- висока швидкодія (до 15 кадрів на секунду);
- низький рівень хибнопозитивних результатів;
- стійкість до похибок.

До недоліків методу можна віднести:

- достатньо довгий процес навчання класифікатору;
- невисока точність у деяких випадках (візуальні перешкоди, значна зміна форми або положення об'єктів відносно представлених у навчальній вибірці);

– застосування безпосередньо методу Віоли-Джонса дозволяє лише виявляти об'єкти, але не відстежувати їх.

1.3.4 Методи розпізнавання на основі штучних нейронних мереж

На сьогоднішній день, великої популярності набувають алгоритми обробки відео, які базуються на використанні різних типів штучних нейронних мереж для вирішення задач розпізнавання [1,8].

Штучні нейронні мережі застосовуються також і для виявлення та класифікації об'єктів на зображеннях.

Для задачі виявлення об'єктів найбільш широко застосовуються згорткові нейронні мережі. Це один з видів багат шарових перцептронів, який добре підходить для виявлення об'єктів на зображеннях, бо потребує невеликої попередньої обробки даних, а результат роботи не залежить від зсуву або повороту шуканого об'єкту на зображенні.

Методи, засновані на використанні штучних нейронних мереж, мають такі переваги:

- можливе виявлення об'єктів декількох класів одним класифікатором;
- в цілому дають більшу точність та надійність результатів.

До недоліків цієї групи методів можна віднести:

- високу обчислювальну складність: для використання класифікаторів, побудованих на основі штучних нейронних мереж, як правило, потрібна потужніша техніка та більша кількість обчислювальних ресурсів, що обмежує спектр використання таких класифікаторів;
- недостатньо висока швидкодія: класифікатори, побудовані на основі штучних нейронних мереж дають досить високу якість результатів, але, як правило, не дають достатньої швидкості виявлення для використання їх в реальному часі (не менше 30 кадрів/сек.);
- важко формалізується етап вибору архітектури мережі (кількість нейронів, верств, характер зв'язків).

1.4 Дослідження програмних засобів для розпізнавання об'єктів у відеопотоці

Насьогодні можна виділити такі (системи машинного зору) програмне забезпечення, що дозволяє розпізнавати об'єкти (медичні маски) на підприємстві [9-11]:

- Xeoma Mask Detector PRO;
- Aerialtronics Face Mask Detection Software;
- Hikvision Mask Detection.

1.4.1 Xeoma Mask Detector PRO

Xeoma Mask Detector PRO – програмне забезпечення, що дозволяє розпізнавати наявність маски у людини з використанням алгоритмів штучного інтелекту [9]. Система може виявляти людей, які носять маску, або тих, хто цього не робить. Носіння маски для обличчя під час відвідування громадських місць - одне з суворих і простих правил. Тому зараз багато урядів офіційно оголошують це під час пандемії COVID-19: не можна виходити у громадські місця, не маючи маски. Щоб попередити тих, хто не відповідає вимогам, можна використовувати систему Xeoma Mask Detector PRO, яка контролює зображення камери для людей, які носять або не носять захисні маски для обличчя, а побачивши «злочинця», може активувати різні реакції – наприклад, надіслати сигнал тривоги до служби безпеки. Приклади використання програмного забезпечення Xeoma Mask Detector PRO – магазини, супермаркети, аеропорти, вулиці, залізничні та автобусні станції та публичний транспорт.

До переваг використання Xeoma Mask Detector PRO можна віднести зберігання пропускнуї здатності та обробку зображень в режимі реального часу, до недоліків – вартість.

1.4.2 Aerialtronics Face Mask Detection Software

Aerialtronics Face Mask Detection Software – програмне забезпечення [10], також використовує нейронні мережі та дозволяє:

- виявити маску для обличчя в існуючих мережах IP-камер(підтримка більшості типів IP-камер);
- виконати просту інтеграцію в сторонні інформаційні системи;
- захистити конфіденційності за допомогою «функції маски конфіденційності», щоб гарантувати анонімність;
- виконати розгортання або встановлення системи на комп'ютерах замовника або використовувати облако;
- надсилати сповіщення службі безпеки, статистику або звіти про наявність або відсутність маски для обличчя.

Приклади використання програмного забезпечення Aerialtronics Face Mask Detection Software – публичні місця та офіси.

До переваг використання Aerialtronics Face Mask Detection Software можна віднести обробку зображень в режимі реального часу та захист приватних даних та конфіденційності, до недоліків – необхідність потужних апаратних ресурсів та вартість.

1.4.3 Hikvision Mask Detection

Hikvision Mask Detection – програмне забезпечення [11], що надає технологію виявлення маски, використовуючи алгоритми штучного інтелекту для аналізу даних форми обличчя, щоб виявити, чи носить людина маску для обличчя. На основі результату виявлення можуть бути визначені дії, наприклад, голосове сповіщення або посилання на систему доступу для заборони доступу. Це означає, що швидше, простіше та безпечніше можна перевіряти наявність маски у людей, які проходять через публичні місця, наприклад, транспортні центри, офіси, склади та великі магазини.

Система Hikvision Mask Detection має такі технічні рішення:

- виявлення маски за допомогою камер: AcuSense або термальної камери для виявлення також підвищеної температури у людини;
- виявлення маски за допомогою спеціального DeepinMind мережевого відеореєстратору;
- виявлення маски за допомогою спеціального I Series мережевого відеореєстратору;
- виявлення маски за допомогою HikCentral – потужний інструмент централізації, що поєднує переваги камер DeepinView, AcuSense та Thermal в одному місці; він гнучкий, масштабований, надійний та потужний, забезпечуючи безперебійний моніторинг для ефективного управління ситуацією, включаючи виявлення маски.

Приклади використання програмного забезпечення Hikvision Mask Detection – публичні місця, транспортні центри, офіси, склади та великі магазини.

До переваг використання Hikvision Mask Detection можна віднести наявність різноманітних технічних рішень, до недоліків – необхідність потужних апаратних ресурсів та вартість.

1.5 Висновки з розділу 1

Дослідження існуючих методів розпізнавання об'єктів у відеопотоці в режимі реального часу дозволило сформулювати завдання дослідження. Необхідно розробити систему машинного зору, що дозволяє вирішити задачу розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках.

На основі проведеного дослідження методів в даному розділі, було вирішено реалізувати технологію розпізнавання об'єктів заданого класу на основі методу Віюли-Джонса з використанням характеристичних рис Хаара та

з побудовою каскаду класифікаторів, навчених з допомогою алгоритму AdaBoost.

Розглянуті системи розпізнавання: Xeoma Mask Detector PRO, Aerialtronics Face Mask Detection Software, Hikvision Mask Detection мають недоліки, а саме: високу ціну, складність в використанні, неможливість використання цих систем в умовах виробництва, необхідність потужних апаратних ресурсів, недостатню продуктивність та достовірність результатів в умовах браку освітленості. Тому виникла необхідність розробити програмне забезпечення, що дозволяє розпізнавати об'єкти заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках.

2 ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ВИРОБНИЦТВІ

2.1 Опис технології розпізнавання об'єктів у відеопотоці

Інформаційна технологія у реальному часі має вирішувати такі задачі:

- визначення об'єктів заданих класів у кадрі (наявність медичної маски, каски та захисних окулярів одночасно);
- визначення відсутності одного, декілька або всіх елементів обладнання.

Для вирішення визначення об'єктів планується використовувати підхід на основі методу Віоли-Джонса з використанням характеристичних рис Хаара та навчанням класифікаторів за допомогою алгоритму AdaBoost.

Запропонована у даній роботі технологія розпізнавання об'єктів заданого класу на виробництві складається з наступних етапів (рис.2.1).

Етап 1. Отримання вхідних даних.

Отримання кадру з відеопотоку або зчитування зображення.

Етап 2. Валідація отриманого кадру.

Якщо кадр було отримано з відеопотоку, то встановлюємо флаг рівним true, що сигналізує про успішне зчитування кадру, в протилежному випадку – false.

Етап 3. Конвертація отриманого кадру у потрібний формат.

Виконання конвертації отриманого кадру у потрібний формат файлу зображення з використанням відповідного типу конвертації.

Етап 4. Класифікація об'єктів на зчитаному кадрі (застосування методу Віоли-Джонса з використанням характеристичних рис Хаара та навчанням класифікаторів за допомогою алгоритму AdaBoost).

Етап 5. Відображення результатів розпізнавання. Опрацьовуємо результати роботи класифікатора: малюємо прямокутник зеленого (для правильних об'єктів) або червоного (для неправильних) на кадрі, та пишемо текст відповідний текст.

Етап 6. Відображення результуючого зображення на екрані у окремому вікні.

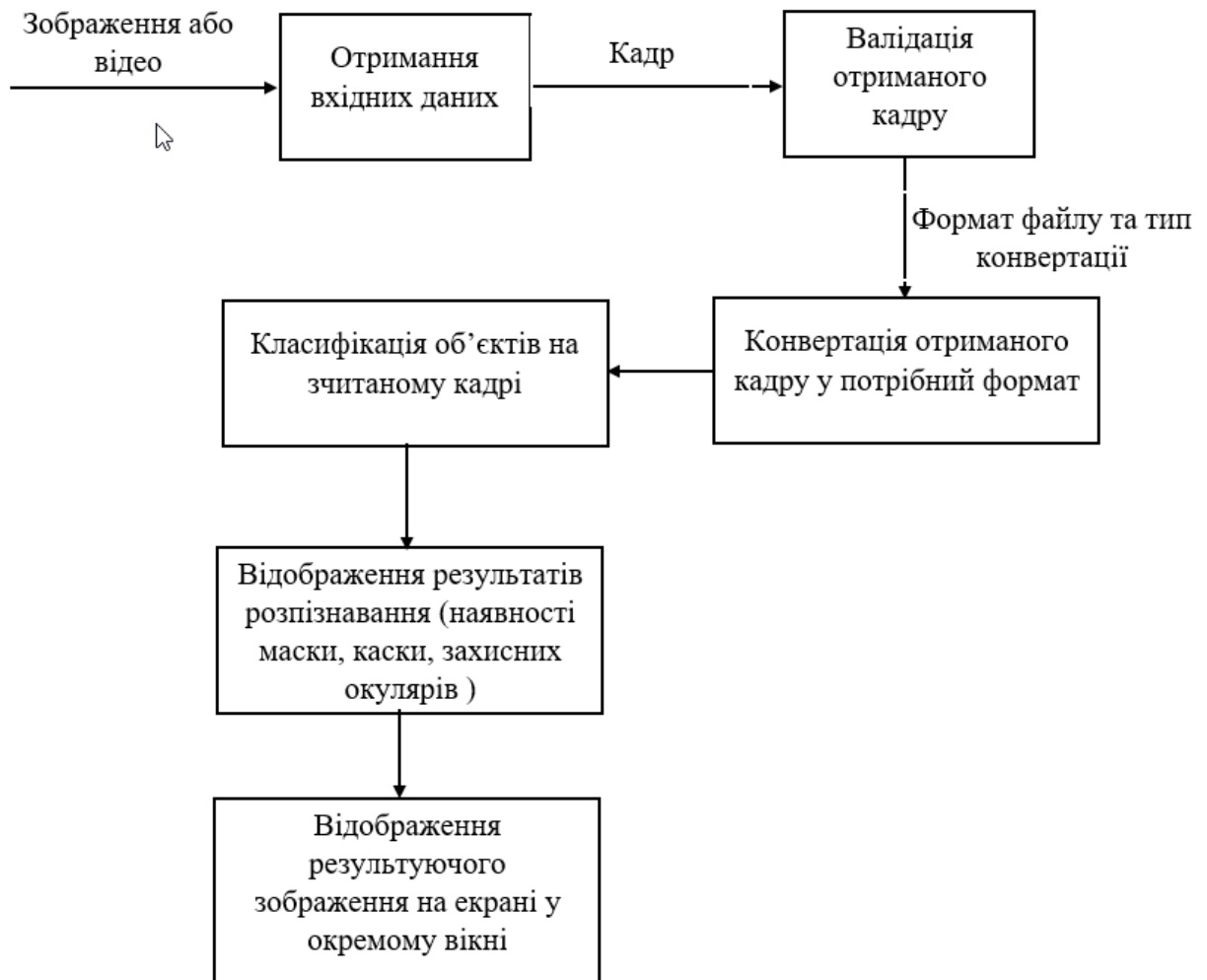


Рисунок 2.1 – Технологія розпізнавання об'єктів заданого класу на виробництві

Наведено опис технології що, дозволяє розпізнавати об'єкти на виробництві. Для реалізації технології у вигляді системи машинного зору необхідно реалізувати та дослідити каскадні класифікатори, що дозволять розпізнавати “коректні” об'єкти: у даному контексті задачі – це людина, у якої фізично наявні маска, каска та захисні окуляри, або розпізнавати “некоректні” об'єкти: це, навпаки, людина, у якої відсутній будь-який з елементів. Для

програмної реалізації системи потрібно обрати мову програмування та інструментарій розробки алгоритмів.

2.2 Дослідження програмних засобів розробки системи

2.2.1 Вибір мови програмування

C++ – це мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Вона вже протягом багатьох років є потужним інструментом програмування, що дозволяє швидко та зручно оброблювати дані, і водночас, будувати об'єктно-орієнтовані системи через класи, має доповнення до стандартної бібліотеки порівняно з C і підтримує обробку винятків, простори імен, вбудовані функції та перевантаження імен і операторів [12].

Python є стандартом де-факто в розробці інтелектуальних систем, саме тому що більшість інструментів машинного навчання розроблена для цієї мови. Виходячи з цього було вирішено використовувати саме мову Python.

Python – інтерпретована об'єкто-орієнтована мова програмування з високим рівнем абстракції з сильною але динамічною типізацією. Розроблена в 1990 році Гвідо ван Россумом. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання існуючих компонентів. Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується декілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована [13-14].

До переваг мови програмування Python можна віднести:

- простий та елегантний синтаксис, який дозволяє писати програми, які легко читати;

- велика стандартна бібліотека, яка підтримує рішення більшості задач програмування;
- зручність вирішення математичних проблем;
- наявність інтерактивного режиму;
- крос-платформність – можливість роботи в середовищі будь якої операційні системи;
- відкритий вихідний код;
- висока швидкість роботи, порівняно з іншими інтерпретованими мовами програмування;
- автоматичне керування пам'яттю [14].

Scala – мультипарадигмова мова програмування, що поєднує властивості об'єктно-орієнтованого та функційного програмування [15-16]. Назва Scala утворена зі слів «scalable» (масштабовна) та «language» (мова), для того щоб задекларувати, що мова може рости разом з вимогами користувачів.

Scala-програми багато в чому схожі на Java-програми, та можуть вільно взаємодіяти з Java-кодом. Мова включає одноманітну об'єктну модель – в тому сенсі, що будь-яке значення є об'єктом, а будь-яка операція – викликом методу. При цьому є також функціональним мовою в тому сенсі, що функції - це повноправні значення.

В Scala включені потужні та одноманітні концепції абстракцій як для типів, так і для значень. Зокрема, мова містить гнучкі симетричні конструкції домішок для композиції класів і типажів. Дозволяє виробляти декомпозицію об'єктів шляхом порівняння зі зразком; зразки і вирази при цьому узагальнені для підтримки природної обробки XML-документів. В цілому, ці конструкції дозволяють легко висловлювати самостійні компоненти, які використовують бібліотеки Scala, не користуючись спеціальними мовними конструкціями.

Мова допускає зовнішні розширення компонентів з використанням уявлень (views). Можливості узагальненого програмування реалізуються за рахунок підтримки узагальнених функцій (generics), в тому числі вищого

типажу (generics of a higher kind). Крім різних класичних структурних типів даних, в мову включена підтримка екзистенціальних типів.

Порівняльна характеристика мов програмування наведена у табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика мов програмування

Мова програмування / Критерій	C++	Python	Scala
Підтримка парадигм програмування	мова програмування високого рівня з підтримкою об'єктно-орієнтованої, узагальненої та процедурної парадигм програмування	інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією	мультипарадигмова мова програмування, що поєднує властивості об'єктно-орієнтованого та функційного програмування
Наявність бібліотек машинного навчання, комп'ютерного зору	+	+	—
Область застосування	Оптимізація програмного коду, розробка ігор (Unreal Engine), GPU-обчислення, наукові дослідження	Веб розробка, Data Science: машинне навчання, аналіз даних та візуалізація, автоматизація процесів	Data Science: машинне навчання, аналіз даних та візуалізація, Big Data

2.2.2 Вибір бібліотеки для розробки системи з використанням алгоритмів комп'ютерного зору

OpenCV [17] (Open Source Computer Vision Library, бібліотека комп'ютерного зору з відкритим вихідним кодом) – бібліотека функцій та

алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека розроблена Intel і на тепер підтримується Willow Garage та Itseez. Реалізована на C / C++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних та комерційних цілях – поширюється на умовах ліцензії BSD. OpenCV – найбільш популярна та різноманітна за функціоналом бібліотека у цій сфері. Завдяки відкритості та невеликій кількості ліцензійних обмежень OpenCV має велике співтовариство розробників та швидко розвивається. Бібліотека містить понад 2500 оптимізованих алгоритмів, серед яких повний набір як класичних так і практичних алгоритмів машинного навчання і комп'ютерного зору.

Проте великий розмір цієї бібліотеки є також і наслідком декількох її недоліків: не дотримано єдиного стилю написання коду у різних модулях, деякі алгоритми є недостатньо оптимізованими та протестованими.

Google Vision Application Programming Interface (API) – платформа Google, що надає технології комп'ютерного зору, що можуть бути використані для виявлення об'єктів на зображеннях [18]. Платформа Google, що надає технології комп'ютерного зору, а саме бібліотеку Faces API – для пошуку та відстеження облич на зображеннях. Засоби бібліотеки дозволяють визначати орієнтацію обличчя в просторі, розраховувати положення основних точок обличчя (ніс, очі, кути рота) і виробляти найпростіше розпізнавання: визначати ймовірність того, що очі людини відкриті, і ймовірність посмішки. Faces API надає можливість визначення людських облич на готових зображеннях та потоках, що йдуть з камери мобільних пристроїв. Але ця платформа основана на мережевому інтерфейсі та надає лише публічний API, доступний через Інтернет, що повністю виключає можливість використання її функціоналу в реальному часі.

SimpleCV – фреймворк з відкритим кодом для створення застосунків для комп'ютерного зору [19]. За допомогою нього можна отримати доступ до декількох потужних бібліотек комп'ютерного зору, таких як OpenCV – без

необхідності попередньо дізнаватися про глибину бітів, формати файлів, кольорові простори, керування буфером, власні значення або матрицю в порівнянні із сховищем растрових зображень. Фреймворк додатково надає інтерфейс роботи з захопленням відеопотоку з різних джерел.

Imutils – бібліотека, що використовує методи OpenCV для розпізнавання об'єктів [20]. Бібліотека містить набір зручних функцій для полегшення основних функцій обробки зображень, таких як переклад, обертання, зміна розміру, скелетування, відображення зображень Matplotlib, сортування контурів, виявлення країв, тобто містить методи різного роду перетворень і обробки вихідного зображення.

TensorFlow – відкрита програмна бібліотека для машинного навчання для низки задач [21]. Розроблена компанією Google для систем, здатних будувати та тренувати нейронні мережі на виявлення та розшифровування образів та кореляцій, аналогічно до навчання й розуміння, які застосовують люди. На цей час бібліотеку застосовують як для досліджень, так і для розробки продуктів Google.

2.3 Висновки з розділу 2

В розділі було розроблено технологію розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках, яка містить шість етапів.

Також було виконано дослідження програмних засобів розробки системи: була обрана мова програмування Python та бібліотека комп'ютерного зору з відкритим вихідним кодом OpenCV.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ВИРОБНИЦТВІ

3.1 Опис основних рішень щодо реалізації системи

Для реалізації системи було обрано мову програмування Python та бібліотеку OpenCV.

OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення з відкритим кодом. Бібліотека надає засоби для обробки і аналізу вмісту зображень, у тому числі розпізнавання об'єктів на фотографіях (наприклад, осіб і фігур людей, тексту тощо), відстежування руху об'єктів, перетворення зображень, застосування методів машинного навчання і виявлення загальних елементів на різних зображеннях [17].

Для цієї системи немає необхідності розроблювати користувацький візуальний інтерфейс, так як програма є лише реалізацією технології, що вирішує окремі задачі комп'ютерного зору і може бути використана в інших програмних системах. Тому для демонстрації можливостей розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині було вирішено використовувати вбудовані функції OpenCV, які надають можливість отримувати зображення з веб-камери чи файлу та виводити його на екран.

3.2 Основні рішення щодо модульного представлення системи

В роботі необхідно вирішити задачу розпізнавання: позитивний (правильний) об'єкт та негативний (неправильний). Для цього слід виконати тренування 2 окремих класифікаторів. Один буде розпізнавати правильний об'єкт (наявність медичної маски, каски та захисних окулярів одночасно), інший – неправильний (відсутній один, декілька або всі елементи обладнання).

Для тренування обох класифікаторів, виконуємо один і той же набір операцій, але використовуємо різні дані.

У кореновому каталозі проєкту знаходяться дві готові моделі: cascade_pos.xml та cascade_neg.xml (рис. 3.1).

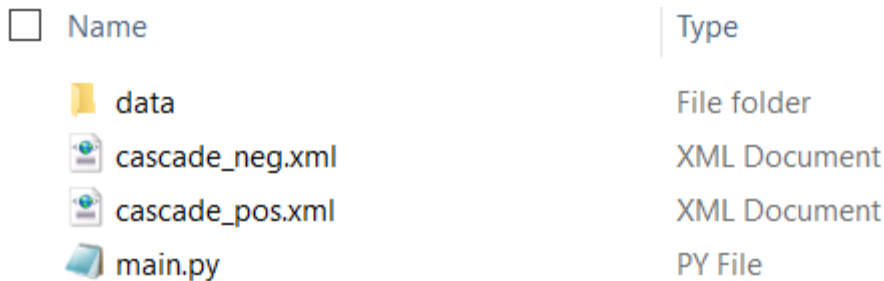


Рисунок 3.1 – Структура проєкту

Опис змінних файлу main.py наведений у таблиці 3.1.

Таблиця 3.1 – Опис змінних файлу main.py

Змінна	Тип	Спосіб задання значення	Опис
1	2	3	4
posClassifier	клас CascadeClassifier	конструктор класу, що приймає шлях до файлу з моделлю, як параметр	класифікатор, що розпізнає “коректні” об’єкти
negClassifier	клас CascadeClassifier	конструктор класу, що приймає шлях до файлу з моделлю, як параметр	класифікатор, що розпізнає “некоректні” об’єкти
videoCapture	клас VideoCapture	конструктор класу, що приймає ідентифікатор пристрою, як параметр	об’єкт, що описує відео-потік та методи роботи з ним

Продовження табл.3.1

1	2	3	4
successFlag	Логічний тип (Boolean)	повертається методом read() класу VideoCapture	Прапорець, що сигналізує про коректне зчитування поточного кадру з потоку
videoFrame	Кадр у вигляді масиву	повертається методом read() класу VideoCapture	зашифровані у масив чисел дані, що описують поточний кадр з потоку
grayFrame	Кадр у вигляді масиву	повертається функцією cvtColor()	зашифровані у масив чисел дані, що описують поточний кадр у відтінках сірого
positives	Масив кортежів з координатами	повертається методом detectMultiscale() класу CascadeClassifier	масив об'єктів, які описують прямокутник, у площі якого знаходиться “коректний” об'єкт
negatives	Масив кортежів з координатами	повертається методом detectMultiscale() класу CascadeClassifier	масив об'єктів, які описують прямокутник, у площі якого знаходиться “некоректний” об'єкт

Дані, що використовувались для тренування моделей знаходяться в папці data (рис. 3.2).

☐ Name	Type
 negative_detector	File folder
 positive_detector	File folder
 fix.py	PY File
 opencv_createsamples	File
 opencv_traincascade	File
 trainer.py	PY File

Рисунок 3.2 – Структура папки data

Структура папки data така:

- positive_detector – дані для навчання класифікатора правильних об'єктів;
- negative_detector – дані для навчання класифікатора неправильних об'єктів;
- fix.py – скрипт для корекції колекції вибірки або файлу опису вибірки;
- trainer.py – скрипт для генерації вибірки;
- opencv_createsamples – текстовий файл з прикладом запуску утіліти OpenCV для генерації вектору позитивних (правильних ?) зображень;
- opencv_traincascade – текстовий файл з прикладом запуску утіліти OpenCV для запуску процесу навчання.

Приклад структури папок negative_detector та positive_detector наведений на рис. 3.3.

☐ Name	Type
 bad	File folder
 good	File folder
 haarcascade	File folder
 bad.dat	DAT File
 good.dat	DAT File
 samples.vec	VEC File
 trainer.conf	CONF File

Рисунок 3.3 – Структура папки negative_detector

Структура папок negative_detector та positive_detector наступна:

- bad – папка з негативними зображеннями; містить приклади зображень, на яких немає шуканого об’єкта;
- good – папка з позитивними зображеннями; містить зображення, на яких присутній шуканий об’єкт;
- good.dat – файл опису позитивної вибірки; має наступний формат: good/1.bmp 1 0 0 194 223; спочатку зазначається відносний шлях до зображення, далі – кількість позитивних об’єктів на зображенні, далі описуються координати прямокутників, у яких наявний шуканий об’єкт; для зручності було вирішено використовувати зображення, на яких тільки один позитивний об’єкт і координати його прямокутника – це (0,0) – (<ширина зображення>, <висота зображення>).
- bad.dat – файл опису негативної вибірки; містить лише список відносних шляхів до зображень;
- samples.vec – вектор позитивних зображень, генерується запуском утиліти opencv_createsamples; використовується для подальшої обробки утилітою opencv_traincascade;
- trainer.conf – конфігураційний файл, використовується скриптом для тренування trainer.py; містить інформацію про актуальну кількість позитивних та негативних зображень вибірки.

3.3 Опис особливостей реалізованих алгоритмів

3.3.1 Опис процесу тренування моделі (навчання)

Розглянемо алгоритм тренування моделі (навчання) детальніше.

Етап 1. Генерація вибірки за допомогою скрипту `trainer.py`.

Етап 2. Форматування та конвертація набору позитивних зображень у спеціальний вектор (`samples.vec`) за допомогою утиліти `opencv_createsamples`.

Етап 3. Тренування каскаду за допомогою утиліти `opencv_traincascade`.

Опис принципу роботи скрипту `trainer.py`. Скрипт працює у двох режимах – генерація позитивної та негативної вибірки.

Для розділення логіки роботи було розроблено 3 класи: `FrameProvider`, `DataController` та `Selector`. Взаємодію між ними описано у функції `main()`. Узагальнений алгоритм складається з наступних дій: Ініціалізація -> Основний цикл -> Звільнення ресурсів.

На етапі ініціалізації виконується зчитування аргументів та конструювання основних об'єктів. Основний цикл виглядає наступним чином:

- отримання кадру від `FrameProvider` та його передача до `Selector`;
- `Selector` отримує кадр та повертає обраний користувачем регіон;
- `DataController` оновлює вибірку та файли конфігурацій у відповідності до заданого режиму роботи скрипта.

Клас `FrameProvider` – виконує захоплення кадрів з відеопотоку, очікуючи натискання на клавішу 'с'. Так користувач обирає кадр з потоку, який частково чи повністю буде включений до вибірки.

Клас `Selector` – призначений для відображення обраного користувачем кадру та виділенням користувачем регіону.

Клас `DataController` – підтримує синхронізацію робочого стану програми та конфігураційних файлів.

Запуск утиліти `opencv_createsamples`: використовується для генерації спеціального вектор-файлу, який потрібен для процесу тренування каскаду. У якості параметрів приймає шлях до вектор-файлу (`samples.vec`), шлях до файлу

з описом позитивних зображень (good.dat), ширину та висоту, до яких має бути стиснене кожне зображення із вибірки.

Запуск утіліти `opencv_traincascade`: безпосередній процес тренування. Параметри, що передаються: назва папки з результатами тренування (haarcascade), вектор-файл з позитивними зображеннями (samples.vec), шлях до файлу з описом негативної вибірки (bad.dat), ширина та висота “зразкового” позитивного зображення (мають бути рівними тим, що використовувались при запуску `opencv_createsamples`), кількість негативних та позитивних зображень.

В результаті запуску, через деякий час у папці haarcascade знайдемо натреновану модель під назвою cascade.xml. Окрім того у цій папці зберігається покроковий стан тренування, для можливості переривання та поновлення процесу (часто займає досить довгий проміжок часу). Файл cascade.xml буде використаний для ініціалізації об’єкта класу CascadeClassifier, що використовується для динамічного розпізнання об’єктів.

3.3.2 Опис процесу розпізнавання

Розглянемо процес розпізнавання детальніше.

Етап 1. Ініціалізація:

- імпорт бібліотеки OpenCV до виконуваного скрипт-файлу;
- завантаження класифікаторів з файлу за допомогою засобів OpenCV (CascadeClassifier);
- захоплення та валідація відео-потoku. (VideoCapture, VideoCapture.isOpened()).

Етап 2. Розпізнавання складається з наступних кроків:

- а) зчитування кадру з відеопотоку (VideoCapture.read());
- б) валідація отриманого кадру: перша змінна з кортежу, що повертає VideoCapture.read() – це флаг, що сигналізує про успішне зчитування кадру;
- в) конвертація отриманого кадру у форматі BGR (blue, green, red) у зображення у відтінках сірого (gray) за допомогою функції cvtColor(), у яку

передаємо кадр та бажаний тип конвертації, у даному випадку `COLOR_BGR2GRAY`;

г) класифікація об'єктів на зчитаному кадрі. За це відповідає метод `CascadeClassifier.detectMultiscale()`. Викликається для кожного класифікатора. Приймає в якості параметрів:

- 1) зображення, на якому слід виконати класифікацію об'єктів;
- 2) коефіцієнт масштабування зображення. Підбирається емпіричним методом. Чим нижче значення, тим менший крок масштабування (підвищує вірогідність правильного розпізнавання, але суттєво впливає на швидкодію). Не може бути меншим за 1;

- 3) мінімальне число сусідніх об'єктів в певному регіоні на зображенні для формування позитивного результату. (Наприклад, в певній області має бути 3 об'єкта, які видаються класифікатором за єдиний. Своєрідний “поріг” розпізнавання). Підбирається емпіричним методом для кожного класифікатора;

- 4) повертає масив об'єктів, що описують прямокутник, у якому знаходиться розпізнаний на кадрі об'єкт;

д) відображення результатів розпізнавання. Опрацьовуємо результати роботи класифікатора: малюємо прямокутник зеленого (для правильних об'єктів) або червоного (для неправильних) на кадрі (функція `rectangle()`), та пишемо текст відповідний текст (функція `putText()`);

е) відображаємо результуюче зображення на екрані у окремому вікні (`imshow()`).

Етап 3. Звільнення ресурсів:

– `videoCapture.release()` – закриття потоку відео.

– `destroyAllWindows()` – знищення створених за допомогою OpenCV вікон.

3.4 Висновки з розділу 3

В розділі наведено опис файлів розробленого проекту, також процесу тренування або навчання моделі та розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1 Функціональні можливості системи

Розроблена система вирішує задачі розпізнавання об'єктів заданого класу на виробництві в реальному часі у відеопотоках.

Розроблена система має наступні функціональні можливості:

- пошук та виявлення наявності медичної маски синього коліру у людини на кадрах відео;
- відстеження білої каски та жовтих окулярів у працівника промислового підприємства на послідовних кадрах відеозапису;
- виконання розпізнавання об'єктів заданого класу з будь-якого зручного ракурсу камери зліва або права в проході.

Розроблена система здатна ефективно вирішувати описані задачі у режимі реального часу на потоковому відео зі швидкістю роботи не меншою за 30 кадрів на секунду.

Приклади результатів роботи розробленої системи наведено далі. Результат при наявності медичної маски синього коліру у людини зображено на рис 4.1.

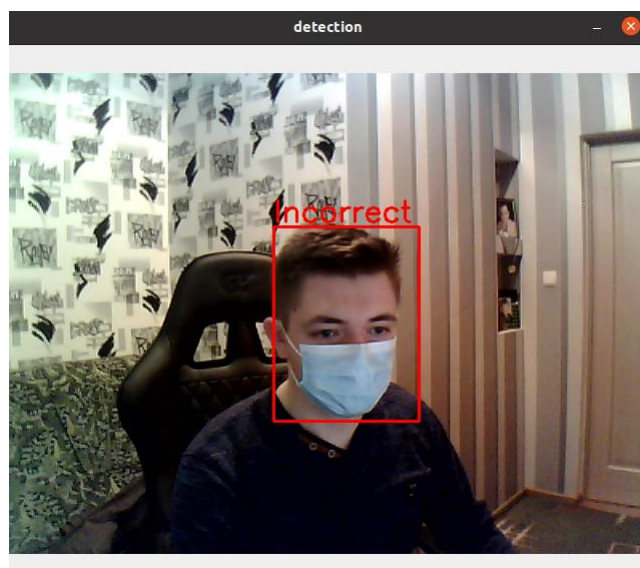


Рисунок 4.1 – Результат при наявності медичної маски синього коліру у людини

Відстеження медичної маски синього коліру та жовтих захисних окулярів у працівника наведено на рис.4.2.



Рисунок 4.2 – Відстеження медичної маски синього коліру та жовтих захисних окулярів у працівника

Результат при наявності білої каски у працівника наведено на рис.4.3.

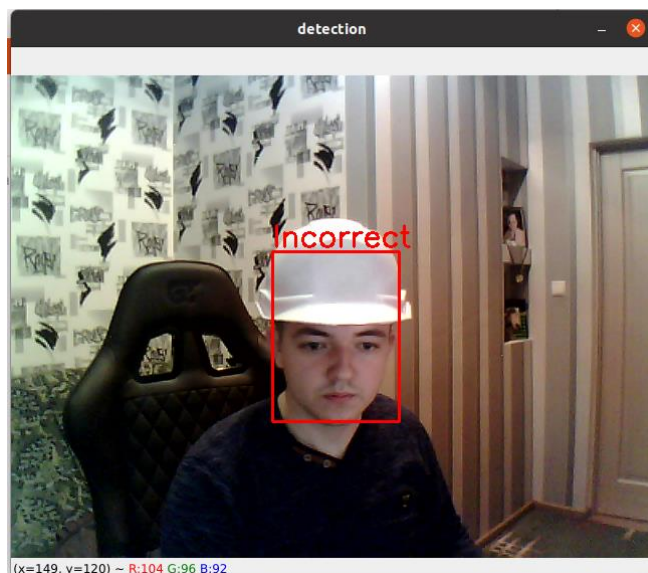


Рисунок 4.3 – Результат при наявності білої каски у працівника

На рисунку 4.4 можна побачити результат роботи програми при наявності білої каски, медичної маски синього коліру та жовтих захисних окулярів у працівника.

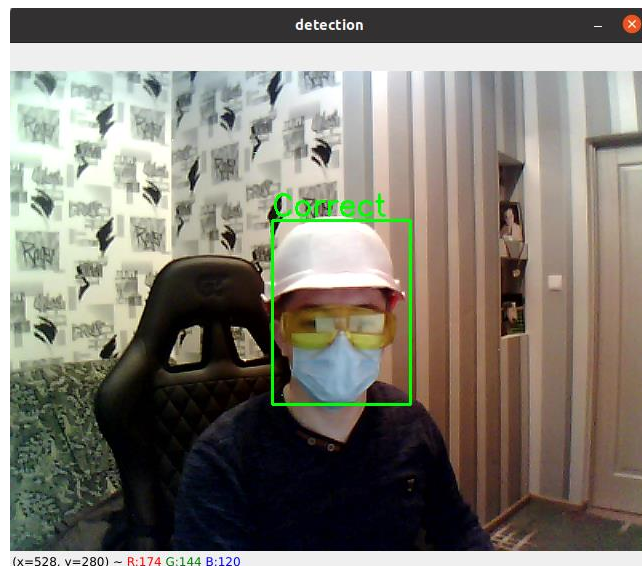


Рисунок 4.4 – Результат при наявності білої каски, медичної маски синього коліру та жовтих захисних окулярів у працівника

4.2 Порівняльний аналіз функціональних можливостей системи з відомими аналогами

З відомих аналогів, що реалізують технологію, подібну розробленої у даній роботі, можна відзначити такі системи:

- Xeoma Mask Detector PRO;
- Aerialtronics Face Mask Detection Software;
- Hikvision Mask Detection.

Порівняння функціональних можливостей існуючих аналогів систем для розпізнавання об'єктів заданого класу на виробництві та розробленої системи наведено в таблиці 4.1.

Таблиця 4.1 – Порівняльний аналіз систем для розпізнавання об’єктів заданого класу на виробництві

Системи\ Параметри	Xeoma Mask Detector PRO	Aerialtronics Face Mask Detection Software	Hikvision Mask Detection	Розроблена система
Пошук наявності медичної маски	+	+	+	+
Відстеження каска	—	—	—	+
Відстеження захисних окулярів	—	—	—	+
Робота в режимі реального часу	+	+	+	+
Портативність	—	—	—	+

На основі проведеного аналізу можна зробити висновок, що розроблена система, на відміну від існуючих аналогів, дозволяє визначати наявність медичної маски синього коліру у людини, а також вирішує задачу визначення білої каски та жовтих окулярів у працівника промислового підприємства в режимі реального часу на відео.

4.3 Оцінка показника достовірності розпізнавання

Для оцінки показника достовірності розпізнавання об’єктів заданого класу на виробництві за розробленою технологією було проведено низку експериментів. В ході яких було встановлено залежність показника достовірності від умов, у яких цей показник розраховується. На показник достовірності впливають такі параметри: якість освітленості робочої зони, кут ведення відеофіксації та фізична відстань від відеопристрою до потенційного об’єкта розпізнавання.

Для точного опису вищеописаної залежності було досліджено результати, згенеровані системою під час обробки 1000 статичних зображень у різних умовах. Результати експериментів наведені у таблиці 4.2.

Таблиця 4.2 – Результати експериментів

№	Освітленість, лк (люкс)	Кут, градус	Відстань, м	Кількість правильно розпізнаних об'єктів	Показник достовірності, %
1	1600	90	5	153	15,3
2	5800	135	2	396	39,6
3	19000	180	1	768	76,8

4.4 Напрямки подальшого розвитку технології

Розроблена система в результаті виконання роботи вирішує поставлені задачі та показує достатньо добрі результати.

Проведений аналіз результатів та отримана оцінка показника достовірності розпізнавання об'єктів заданого класу на виробництві розробленої системи показують, що можливе поліпшення технології за рахунок додавання протоколу Modbus TCP/IP для сигналізації результатів розпізнавання на сервер.

4.5 Висновки з розділу 4

В цьому розділі було описано та досліджено функціональні можливості розробленої системи. Ці функціональні можливості були порівняні з іншими відомими системами, що працюють у сферах визначення об'єктів заданого класу на виробництві або підприємстві за допомогою алгоритмів комп'ютерного зору або машинного навчання. Для оцінки показника достовірності розпізнавання об'єктів заданого класу на виробництві було

проведено тестування, що показало високий показник достовірності розпізнавання у розробленій технології.

Також було запропоновано можливе поліпшення технології за рахунок додавання протоколу Modbus TCP/IP для сигналізації результатів розпізнавання на сервер.

5 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

В ході виконання дипломної кваліфікаційної роботи магістра було досліджено сучасні підходи щодо розпізнавання об'єктів на виробництві та розроблено програмне забезпечення яке дозволяє розпізнавати наявність маски, каски та захисних окулярів у людині, що працює на виробництві.

Наявні аналоги таких систем розпізнавання (Xeoma Mask Detector PRO, Aerialtronics Face Mask Detection Software, Hikvision Mask Detection) мають недоліки, а саме: високу ціну, складність в використанні, недостатню продуктивність та недостатню достовірність результатів.

Описаний вище метод виконує ту ж функцію але дає вищий показник достовірності результатів та використовує менше апаратних ресурсів. На існуючий час ринок програмного забезпечення такого типу представлений імпортованими програмними продуктами з мінімальною ціною близько 4900 гривень (Xeoma Mask Detector PRO).

В результаті проведеного дослідження було розроблено програмне забезпечення, яке дозволяє розпізнавати наявність маски, каски та захисних окулярів у людині, що працює на виробництві з високим ступенем достовірності. Планується продати створене програмне забезпечення в кількості 100 копій по 2850 гривень за копію.

5.1 Планування розробки програмного виробу

Весь комплекс розробки програмного продукту можна розділити на етапи. Для кожного етапу вказуються трудомісткість, кількість виконавців та тривалість робіт. В розробці приймають участь програміст протягом 1,5 місяця та консультант протягом 0,2 місяця. Розробка ПЗ починається з 5 жовтня і має бути виконана до 17 листопада 2020 року.

Тривалість робіт визначають за формулою (5.1):

$$T_{\text{ц}} = \frac{R}{Q}, \quad (5.1)$$

де $T_{\text{ц}}$ – тривалість циклу, днів;

R – кількість виконавців, чол;

Q – трудомісткість, людино-днів.

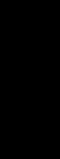
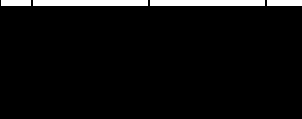
Результати планування заносяться до таблиці 5.1.

Таблиця 5.1 – Характеристика запланованих робіт

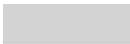
Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд.- днів	% до підсумку		
1 Аналіз методів розпізнавання об'єктів	2	5,5	програміст	2
2 Визначення вимог до програмного продукту	2	5,5	програміст	2
3 Проектування структури програми	8	22	програміст, аналітик	6 2
4 Розробка схеми функціонування програми	4	11,1	програміст	4
5 Створення програмного коду	12	33,3	програміст	12
6 Тестування та налагодження програмного продукту	5	13,8	програміст, аналітик	4 1
7 Складання програмної документації	3	8,3	програміст, аналітик	2 1
Разом	36	100		32

За даними таблиці 5.1 складаємо стрічковий графік планування розробки продукту (табл. 5.2).

Таблиця 5.2 – Стрічковий графік планування розробки продукту

Найменування робіт	Календарний період, дні						
	05.10-09.10	12.09-16.10	19.10-23.10	26.10-30.10	02.11-06.11	09.11-13.11	16.11-20.11
1 Аналіз методів розпізнавання об'єктів							
2 Визначення вимог до програмного продукту							
3 Проектування структури програми		 					
4 Розробка схеми функціонування програми							
5 Створення програмного коду							
6 Тестування та налагодження програмного продукту						 	
7 Складання програмної документації							 

Позначки:

 - консультант  - програміст

5.2 Визначення витрат на розробку

Для визначення витрат на аналіз методологій та інструментів керування програмними проектами та розробку програмного продукту складається калькуляція кошторисної вартості робіт, яка включає наступні пункти:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- витрати на спеціальне обладнання;
- матеріали і комплектуючі вироби;
- накладні витрати;
- податки.

5.2.1 Розрахунок основної та додаткової заробітної плати

Необхідні початкові дані:

- час на розробку – до 1,5 міс.;
- робота спеціаліста – 1,5 міс. (32 дні) з заробітною платою (ЗП) 9140 грн/міс;
- робота консультанта – 0,2 міс. (5 днів) з ЗП 8700 грн/міс.

Додаткову заробітну плату приймають рівною 10% від основної заробітної плати працівників і розраховують за формулою (5.2):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 0,1, \quad (5.2)$$

Результати розрахунків основної заробітної плати наведено в таблиці 5.3.

Таблиця 5.3 – Розрахунок основної та додаткової заробітної плати

Посада виконавця	Чисельність фахівців, осіб	Місячний оклад, грн	Кількість робочих міс.	Сума основної ЗП, грн	Сума додаткової ЗП, грн
Програміст	1	9140	1,5	13710	1371
Аналітик	1	8700	0,2	1740	174
Разом	2			15450	1545

5.2.2 Відрахування на єдиний соціальний внесок

На даному етапі розраховуються відрахування на соціальні заходи за формулою (5.3) – єдиний соціальний внесок, що включають у себе відрахування до Пенсійного фонду, фонду страхування на випадок тимчасової непрацездатності, фонду страхування по безробіттю та фонду страхування від нещасних випадків на виробництві:

$$B_{\text{с.з.}} = (ЗП_{\text{осн}} + ЗП_{\text{дод}}) \cdot K_{\text{с.з.}}, \quad (5.3)$$

де $K_{\text{с.з.}}$ – коефіцієнт відрахувань на соціальні заходи (єдиний соціальний внесок), згідно діючого законодавства становить 0,22.

$ЗП_{\text{осн}}$ – основна заробітна плата, грн.

$ЗП_{\text{дод}}$ – додаткова заробітна плата, грн.

Отримаємо $B_{\text{с.з.}} = 5251,46$ грн.

5.2.3 Визначення витрат на матеріали

Матеріали, які знадобилися для розробки даної програми представлені в табл. 5.4.

Таблиця 5.4 – Витратні матеріали

Матеріал	Кількість	Вартість, грн
Папір формату А4	500 листів	90
CD-RW	1 шт.	6
Картридж для принтера	1 шт.	400

Витрати на матеріали розраховуються за формулою (5.4):

$$M = \sum_{i=1}^n (C_i \cdot N_i (1 + K_{\text{т.з.}}) - C_{io} \cdot C_{io}), \quad (5.4)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати (0.03 – 0.1);

C_i – ціна i -го найменування матеріалу, напівфабрикату, комплектуючого, грн.;

N_i – ціна i -го матеріалу, напівфабрикату, комплектуючого;

C_{io} – ціна поворотних відходів i -го найменування матеріалу, грн.;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів, комплектуючих.

Необхідні вхідні дані: $C_i = 0$; $N_{io} = 0$; $K_{\text{т.з.}} = 0.05$.

Розрахуємо витрати на матеріали. Разом витрати на матеріали складають 520,80 грн.

5.2.4 Витрати на спеціальне обладнання

Для розрахунку експлуатаційних витрат на одну годину машинного часу складемо кошторис витрат на зміст і експлуатацію цього устаткування.

Амортизаційні відрахування визначають за формулою (5.5):

$$A = \frac{\Phi_6 \cdot N_a}{100}, \quad (5.5)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн;

N_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки.

Балансова вартість обчислювальної техніки становить 8000 грн. Одержуємо $A = 2000$ грн.

Для розрахунку вартості експлуатації обладнання використаємо формулу (5.6):

$$C_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_{\text{ел.е}}, \quad (5.6)$$

де N_n – номінальна потужність комп'ютера, кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи комп'ютера, год.;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

$C_{\text{ел.е}}$ – ціна одного кВт·ч електроенергії, грн.

Номінальна потужність комп'ютера – 0,2 кВт. Річний ефективний фонд часу роботи комп'ютера становить 1800 годин. Середні коефіцієнти завантаження за часом і за потужністю рівні відповідно 0,9 і 0,6. Ціна однієї кіловат-години електроенергії становить 2,11 грн. Отримаємо: $C_e = 410,184$ грн.

Зарплати обслуговуючого персоналу розраховуються за формулою (5.7):

$$ЗП_{\text{обсл}} = ФЗП_{\text{г}} \cdot (1 + K_{\text{отч}}) \cdot \frac{t_{\text{обсл}}}{\Phi_{\text{еф.обсл}}}, \quad (5.7)$$

де $ФЗП_{\text{г}}$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн;

$K_{\text{відрах}}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{\text{обсл}}$ – час протягом року, необхідний на технічне обслуговування комп'ютера, год/рік;

$\Phi_{\text{еф.обсл}}$ – річний ефективний фонд часу обслуговуючого персоналу, год/рік.

Місячна заробітна плата обслуговуючого персоналу становить 6230 грн., а річний фонд заробітної плати відповідно дорівнює 74760 грн.

Річний ефективний фонд робочого часу обслуговуючого комп'ютери працівника дорівнює 1750 г/рік.

На обслуговування одного комп'ютера по 0.5 годині на місяць, що на рік становить 6 годин.

$$\text{Отримаємо: } ЗП_{\text{обсл}} = 74760 \cdot (1 + 0,22) \cdot \frac{6}{1750} = 312,71 \text{ грн.}$$

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання і складає 240 грн.

Стаття «Інші витрати» приймається рівною 5% від суми всіх попередніх статей витрат на утримання та експлуатацію устаткування. Сума всіх попередніх статей дорівнює 2962,89 грн., 5% від суми становлять 148,14 грн.

Розраховані статті витрат на утримання та експлуатацію устаткування внесені в табл. 5.5.

Таблиця 5.5 – Кошторис витрат на утримання та експлуатацію обладнання

Найменування статей витрат	Сума, грн
1	2
Амортизація обладнання	2000
Експлуатація обладнання (крім витрат на поточний ремонт)	410,18

Продовження табл. 5.5

1	2
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	312,71
Поточний ремонт обладнання	240
Інші витрати	148,14
Підсумок	3111,03

Витрати на оплату машинного часу комп'ютера для написання і налагодження даної програми визначаються за формулою (5.8):

$$C_{\text{мо}} = P_{\text{екс}} \cdot t_{\text{мо}}, \quad (5.8)$$

де $C_{\text{мо}}$ – витрати на оплату машинного часу, грн.;

$P_{\text{екс}}$ – експлуатаційні витрати на одну годину машинного часу комп'ютера, грн. / машино-год.;

$t_{\text{мо}}$ – машинний час комп'ютера для написання і налагодження даного програмного продукту, машино-год.

Експлуатаційні витрати на одну годину машинного часу використовуваного комп'ютеру розраховують діленням суми витрат за кошторисом «Витрати на утримання та експлуатацію обладнання» (таблиця 5.5) на річний ефективний фонд часу роботи комп'ютера. Річний ефективний фонд часу роботи комп'ютера дорівнює 1800 годин. В результаті експлуатаційні витрати на одну годину машинного часу рівні $P_{\text{екс}} = 1,73$ грн./машино-год.

Комп'ютер експлуатується 32 днів в одну зміну, що становить в сумі 256 годин. Таким чином, витрати на оплату машинного часу складуть 442,88 грн.

5.2.5 Інші прямі витрати

В інші прямі витрати входять витрати на комерційне програмне забезпечення, яке використовується при розробці, а саме загальне ПЗ, що використовується постійно при роботі з комп'ютером – 5699 грн, без ПДВ.

Розрахунок інших прямих витрат виконується за формулою 5.9:

$$S = \frac{Ц_{ПО} \cdot T_{КТС}}{\Phi_{\text{еф.КТС}} \cdot T_{\text{с.ПО}}} \quad (5.9)$$

де $Ц_{ПО}$ – ціна програмного забезпечення;

$T_{КТС}$ – машинний час КТС, необхідне користувачу для розробки системи, машино-год/рік;

$\Phi_{\text{еф.КТС}}$ – річний ефективний фонд часу роботи КТС, машино-год/рік;

$T_{\text{с.ПО}}$ – строк служби програмного забезпечення, років.

Таким чином, інші прямі витрати дорівнюють $S = 405,26$ грн.

5.2.6 Розрахунок накладних витрат

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати та ін.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки непрямым шляхом – у відсотках до основної заробітної плати розробників. В даному випадку накладні витрати становлять 40% до основної заробітної плати розробників, що складає 15450 грн, що становить 6180 гривень.

Результати визначення витрат на розробку у вигляді калькуляції кошторисної вартості робіт наведені в таблиці 5.6.

Таблиця 5.6 – Калькуляція кошторисної вартості робіт з розробки та її ціни

Стаття витрат	Сума, грн	Питома вага до підсумку, %
Основна заробітна плата	15450	51,8
Додаткова заробітна плата	1545	5,2
Відрахування на ЕСВ	5251,46	17,6
Вартість матеріалів	520,8	1,7
Витрати на обладнання	442,88	1,5
Інші прямі витрати	405,26	1,4
Накладні витрати	6180	20,8
Разом витрати	29798,4	100
Прибуток (30%)	8939,5	
Разом	38737,92	
ПДВ (20%)	7747,58	
Разом	46485,5	

5.3 Оцінка економічної ефективності

Створений програмний продукт дозволяє отримати відео з високою роздільною здатністю за меншу ціну. Так, існуючі програмні рішення коштують мінімум 4900 гривень, в той же час було вирішено продавати створене ПЗ за ціною 2850 гривень в кількості 100 екземплярів.

Надходження від продажу повинні складати $100 \cdot 2850 = 285000$ гривень, з урахування ризику 50% надходження складатиме 142500 гривень. З огляду на те, що очікуване надходження від продажів вище, ніж ціна розробки, в якій врахована рентабельність 30%, можна зробити висновок про економічну доцільність розробки.

6 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

6.1 Аналіз потенційних небезпек

При дослідженні та програмній реалізації методів розпізнавання об'єктів на виробництві програмістами були використані комп'ютери, камери, які знаходяться в обчислювальному центрі розміром 15 на 10 метрів. Під час роботи на інженерів постійно або періодично впливають такі небезпечні та шкідливі фактори:

- широкий спектр випромінювання від дисплея;
- невідповідність нормам параметрів мікроклімату;
- підвищений рівень електромагнітних випромінювань;
- незадовільне освітлення робочих зон, внаслідок виходу з ладу або забрудненості освітлювальних приладів, що може призвести до погіршення зору або механічних травм;
- підвищена яскравість монітора (недостатня освітленість);
- знижена контрастність;
- тривале перебування в тому ж положенні, і повторення тих самих рухів призводить до синдрому тривалих статичних навантажень (СТСН);
- невідповідність ергономічних характеристик обладнання нормованим величинам;
- забруднення повітря шкідливими речовинами і пилом;
- монотонність праці;
- небезпека виникнення пожежі.

Аналізуючи ризик потенційних небезпек, можна додати такі:

- недостатня професійна підготовка працівників, причина – низький рівень професійних знань та навичок та відсутність проведення усіх необхідних інструктажів з охорони праці, що може призвести до аварійних ситуацій;
- можливість ураження електричним струмом при виконанні частини дослідних робіт на робочій зоні, що може бути пов'язано з порушенням правил

електробезпеки, зокрема, відсутність захисного заземлення, частково оголені дроти, що можуть привести до електричних травм або летальних наслідків.

- нервово-психічні навантаження в наслідок специфіки виконуваних робіт, що призводить до захворювань загального характеру;
- механічне травмування в наслідок не раціонального розташування робочих місць;
- небезпеки, які пов'язані з умовами праці в надзвичайних ситуаціях.

6.2 Заходи із забезпечення безпеки

Правила електробезпеки регламентуються правовими і технічними документами, нормативно-технічною базою. Знання основ електробезпеки обов'язково для персоналу, який обслуговує комп'ютери та електроустаткування.

Небезпека електричного струму на відміну від інших посилюється тим, що людина не в змозі без спеціальних пристроїв виявити напругу на відстані, а також швидкоплинністю ураження – небезпека з'ясовується тільки, коли людина вже уражена.

Для фахівців, які постійно мають відношення до використання і обслуговування комп'ютерів, так і для працівників, які тільки їх використовують, встановлені п'ять кваліфікаційних груп з електробезпеки.

Згідно класифікації приміщень за ступенем безпеки ураження електричним струмом визначених у Правилах улаштування електроустановок, обчислювальний центр відноситься до категорії приміщень без підвищеної небезпеки, тому що відносна вологість повітря не перевищує 75%, температура не більше 27°C, відсутня хімічно агресивне середовище [22].

Виділяють три системи засобів і заходів забезпечення електробезпеки:

- система технічних засобів і заходів;
- система електрозахисних засобів;
- система організаційно-технічних заходів і засобів.

Технічні засоби і заходи з електробезпеки реалізуються в конструкції комп'ютерів при їх розробці, виготовленні і монтажі відповідно до чинних нормативів.

За своїми функціями технічні засоби і заходи забезпечення електробезпеки поділяються на дві групи:

- технічні заходи і засоби забезпечення електробезпеки при нормальному режимі роботи комп'ютерів;
- технічні заходи і засоби забезпечення електробезпеки при аварійних режимах роботи комп'ютерів.

Одним з найбільш небезпечних виробничих факторів при роботі з комп'ютерами, є електричний струм. Для попередження ураження електричним струмом проводяться організаційні та технічні заходи.

Заходи щодо забезпечення електробезпеки розроблено відповідно до вимог [23].

Електрична мережа вмикається і вимикається за допомогою пускової апаратури (рубильником).

До роботи з блоком допускаються особи, які мають посвідчення на право роботи з електроустановками, що вивчили матеріальну частину блоку.

З метою підвищення рівня безпеки, в залежності від призначення, умов експлуатації і конструкції, в електроприладах застосовується одночасно більшість з перерахованих технічних засобів і заходів.

У відповідність з главою 1.7 [22] існують такі загальні вимоги до електроустановок:

- струмопровідні частини електроустановки не повинні бути доступними для випадкового прямого дотику до них, а доступні для дотику відкриті і сторонні провідні частини не повинні перебувати під напругою, що становить небезпеку ураження струмом, як у нормальному режимі роботи електроустановки, так і в разі пошкодження ізоляції;
- для запобігання ураження електричним струмом у нормальному режимі слід застосовувати окремо або в поєднанні такі заходи захисту від

прямого дотику: основну ізоляцію струмопровідних частин; огорожі та оболонки; бар'єри; розміщення поза зоною досяжності.

Для додаткового захисту від ураження електричним струмом в разі прямого дотику в електроустановках напругою до 1 кВ можна застосовувати пристрій захисного відключення (ПЗВ).

Дисплей комп'ютера здатний накопичувати статичний заряд. Небезпека зарядів статичного електрики виявляється у можливості виникнення електричних зарядів у просторі, впливі на обслуговуючий персонал та порушенні нормального ходу робочого процесу. Вплив статичної електрики на обслуговуючий персонал позначається у розрядах статичних зарядів через людину та вплив електростатичного поля. Безпосередньо струм розряду не є небезпечним, оскільки час його протікання через тіло людини малий (дорівнює декільком мілісекундам). Такі короткочасні імпульси можуть викликати електричні удари.

Основні організаційно-технічні заходи і засоби щодо попередження електротравм регламентуються в [23], за якими відповідальність за організацію безпечної експлуатації комп'ютерів покладається на власника.

Згідно з чинними вимогами власник повинен:

- призначити відповідального за справний стан і безпечну експлуатацію комп'ютерів (далі – відповідальний за електрогосподарство);
- створити і укомплектувати відповідно до потреб електротехнічну службу;
- розробити і затвердити посадові інструкції працівників;
- створити на підприємстві такі умови, щоб працівники, на яких покладено обов'язки з обслуговування комп'ютерів відповідно до діючих вимог, своєчасно здійснювали їх огляд, профілактичні заходи.

Для зниження рівня статичної електрики передбачено екранування джерела статичної електрики (монітора) захисним екраном і заземлення екрану.

У електричних схемах передбачено пристрій, який централізовано відключає від джерела живлення все електричне обладнання. У конструкторському бюро всі мережеві розетки мають надписи зі значенням напруги і застережливий знак.

6.3.Заходи з виробничої санітарії та гігієни праці

До основних завдань ергономічної організації робочого місця згідно з вимогами [24] відносяться:

- визначення просторових параметрів робочого місця і його елементів, які відповідають антропометричним характеристикам контингенту працюючих;
- оптимальне розміщення елементів робочого місця відносно користувача на основі аналізу його діяльності [24].

Обладнання і організація робочого місця працюючих з комп'ютерами забезпечує відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності відповідно до вимог [25].

При розміщенні обладнання витримуються необхідні проміжки між обладнанням, відстані від стін, які забезпечують свободу пересування людей, зручність виконання робіт і безпеку працюючих. Робочі місця операторів комп'ютерів, а так само ділянки підготовки технічних носіїв інформації розташовані в ряди. Розташування місць може бути дворядним, трирядним, чотирирядним, а також прямим і поперечним.

На організацію роботи впливає конструкція і параметри основного і допоміжного обладнання, які відповідають вимогам ергономіки: оптимальному розподілу функцій у системі людина – машина. Конструкція обладнання відповідає антропометричним і психофізіологічним даним організму працюючого. Витримані допустимі параметри санітарно-гігієнічних умов праці, а також безпеки експлуатаційного обладнання.

Конструкція робочого столу забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання, враховуючи конструктивні особливості. Висота робочої поверхні столу регулюється в межах 680-800 мм, при відсутності такої можливості висота робочої поверхні столу становить 725 мм. Модульними розмірами робочої поверхні столу для ПК слід вважати: ширину 800, 1000, 1200, 1400 мм, глибину 800 і 1000 мм, при нерегульованій його висоті, рівної 725 мм.

Робоче місце обладнане підставкою для ніг, яка має ширину не менше 300 мм, глибину не менше 400 мм, регулювання по висоті в межах до 150 мм і по куту нахилу опорної поверхні підставки до 20 град. Підставка має рифлену поверхню.

Конструкція робочого стільця (крісла) підтримує раціональну робочу позу при роботі з комп'ютером, дозволяє змінювати позу з метою зниження статичного напруження м'язів шийно-плечової області і спини для попередження стомлення. Робочий стілець (крісло) є підйомно – поворотним і регульованим по висоті і кутам нахилу сидіння і спинки, а також відстані від переднього краю сидіння.

Конструкція стільця повинна забезпечувати максимум результатів праці. Досягти цього можна, лише освоївши раціональні методи і прийоми праці на робочому місці. Тільки вони дозволяють виконати задану роботу якісно, у мінімальний термін і без зайвої напруги.

Специфіка роботи таких працівників полягає у великих зорових навантаженнях разом з маленькою активністю, монотонністю виконаних операцій, вимушеною робочою позою. Ці фактори негативно позначаються на самопочутті працюючого.

При організації робочого місця дуже важливим фактором є робоча поза працівника, тобто положення його корпусу, голови, рук і ніг по відношенню до знарядь праці. Якщо працівник працює сидячи, йому потрібно забезпечити правильну і зручну посадку, яка досягається пристроєм опори для спини, рук,

ніг, правильною конструкцією сидіння, яке сприяє рівномірному розподілу маси тіла.

Усі матеріальні елементи робочого столу ділять на предмети постійного, тимчасового користування з урахуванням цього розташовують у видатного порядку на місцях постійного зберігання, що це економить трудові руху і сили працюючого.

Клавіатуру комп'ютера краще розташовувати на відстані 10-15 мм від краю столу, тоді зап'ястя рук будуть спиратися на стіл. Бажано придбати спеціальну підставку під зап'ястя, що, як стверджують медики, допоможе уникнути хвороби кистей.

Для ефективного використання маніпулятора типу «миша» застосовується спеціальний «килимок» – планшет. Килимок – планшет задовольняє основним критеріям: по-перше, добре тримається на поверхні столу, по-друге, матеріал верхньої поверхні планшета забезпечує вільне ковзання миші.

Введення текстової інформації з клавіатури полегшують підставки для документів. Вони можуть або кріпитися до монітора, або встановлюватися безпосередньо на столі. Багато підставки оснащені лінійками для виділення рядка.

Інструмент, обладнання та предмети праці знаходяться на відстані 560-750 мм на рівні рук працівника, оскільки їх використання не призводить до зайвих рухомим нахилам. Важливим елементом раціонального планування робочого місця є облік індивідуальних антропометричних психофізіологічних даних працюючого.

Робочі місця обладнані відповідними меблями та інвентарем, які відповідають найбільш комфортабельним умов роботи і вимог фізіології, психології та естетики [26].

Для нормальної експлуатації машин електронної та обчислювальної техніки і високопродуктивної праці робітників та інженерно-технічного

персоналу в процесі проектування пристрою передбачений наступний комплекс заходів:

- забезпечення оптимального мікроклімату;
- застосування оптимального освітлення;
- створення оптимальних умов праці на робочому місці;
- зниження психоемоційного і фізичного навантаження.

Оптимальні норми температури повітря в робочій зоні обчислювального центру в холодний і перехідний періоди року при I категорії робіт (легкої) згідно [27] мату виробничих приміщень – 22 ... 24°C, відносна вологість – 60 ... 40%, швидкість руху повітря – 0.2 м / сек. У теплий період року температура м 23 ... 25 °C, відносна вологість – 60 ... 40%, швидкість руху повітря – 0,2 м / сек.

Для забезпечення вищевказаних параметрів передбачено застосування вентиляції – природної (аерації і провітрювання), механічної (загальнообмінної, місцевої припливної та витяжної), кондиціювання повітря.

У приміщенні застосована система кондиціювання згідно з [28], а також регулярно проводиться провітрювання. В приміщенні регулярно здійснюється вологе прибирання.

Для комфортної роботи співробітників кожне приміщення забезпечено елементами природного та штучного освітлення, відповідно до [29]. Оскільки тривала робота за комп'ютером чи документами при поганому рівні освітленості може призвести до перенапруження органів зору.

Зазвичай використовують природне бокове одностороннє освітлення в день, а у ввечері – загальне рівномірне штучне освітлення. Робоче місце розміщене так, щоб природне світло було з лівого боку та уникати потрапляння прямого світла в очі. Також у працівників є можливість для корегування рівня природнього освітлення за допомогою жалюзі або штор. Оскільки яскраве світло в районі периферійного зору може збільшити навантаження на органи зору та призвести до швидкої втоми, освітлення є

рівномірним, тобто ступінь освітленості приміщення та яскравість монітору комп'ютера повинні бути приблизно однаковими [30].

Коефіцієнт пульсації не перевищує 5 %. Рівень освітлення робочого столу в зоні розташування документів знаходиться в межах 300 – 500 лк. Освітлювальні прилади приміщення не повинні створювати відблисків на поверхні екрана, а освітлення екрана має не перевищувати 300 лк.

Правильно спроектоване та встановлене освітлення здатне покращити умови праці працівників, полегшити роботу органів зору, знизити втому, підвищити продуктивності та швидкості роботи, а недостатнє освітлення приміщення призводить до настання передчасної втоми, послабленню уваги та напруженню зору.

Підвищення рівню шуму погіршує умови праці та негативно впливає на організм людини. В результаті у робітників спостерігаються підвищений рівень стомленості, роздратованість, сильні головні болі, зниження концентрації уваги, нервово-психічне навантаження. Тривала дія інтенсивного шуму на слух людини призводить до його часткової або повної втрати [31].

Основними джерелами шуму на робочих місцях в конструкторських бюро є принтери, кондиціонери, системи вентиляції та комп'ютери, в яких джерелами шуму є вентилятори, що охолоджують систему. Рівень шуму та вібрацій на робочих місцях осіб, що працюють за персональним комп'ютером, визначають відповідно до [31]. Для забезпечення дотримання допустимих рівнів шуму на робочих місцях застосовуються засоби звукопоглинання, вибір яких обґрунтовується спеціальними інженерно-акустичними розрахунками.

6.4 Заходи безпеки у надзвичайних ситуаціях

6.4.1 Заходи з пожежної безпеки

Заходи з протипожежної безпеки для обчислювального центру обладнаного комп'ютерами розроблені згідно вимог [32].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному комп'ютерами:

- згідно вимог [32] у приміщенні, обладнаному комп'ютерами можлива пожежа класу – А та Е;

- відповідно до вимог [33], воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали.

Оскільки приміщення належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог [32] воно має II ступінь вогнестійкості.

У разі виникнення пожежі у приміщенні обладнаному комп'ютерами для евакуації персоналу відповідно до вимог [32] передбачені виходи, по обидві сторони приміщення, з одного боку вікно (на пожежні сходи), а з іншого – вхідні двері. Відстань від найбільш віддаленого робочого місця до найближчого евакуаційного виходу не обмежується.

Обладнання, силові та освітленні мережі приміщення, обладнаного комп'ютерами відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог [23]. Вимоги до захисних заходів від ураження електричним струмом, та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

Для виявлення початкової стадії загоряння, швидкого й точного оповіщення служби пожежної охорони про час і місце виникнення пожежі встановлені системи автоматичної пожежної сигналізації (АПС). Згідно вимог [33], в приміщенні обладнаному комп'ютерами встановлена система пожежної й охоронної сигналізації, яка забезпечує виявлення теплових і

димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика [30].

Оскільки приміщення, що обладнане комп'ютерами має площу 150 м^2 , тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» [30] для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 8 штук (з розрахунку один вогнегасник з величиною заряду вогнегасної речовини $3,5 \text{ кг}$ і більше, на 20 м^2 площі приміщення). Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м .

Розрахуємо стаціонарну установку для гасіння пожежі вуглекислотою у виробничому приміщенні, якщо:

- об'єм приміщення $W_{\text{п}} = 550 \text{ м}^3$;
- коефіцієнт $K_{\text{в}} = 1,5$, який враховує особливості процесу газообміну, витік вуглекислоти через нещільність та пройоми приміщення;
- довжина трубопроводу від установки до місця гасіння загорання $L = 80 \text{ м}$.

Необхідна маса вогнегасної газової суміші визначається за рівнянням (6.1):

$$G_{\text{г}} = G_{\text{в}} \cdot W_{\text{п}} \cdot K_{\text{в}} + G_0, \quad (6.1)$$

де $G_{\text{в}}$ – вогнегасна концентрація газового складу для вуглекислоти;

$$G_{\text{в}} = 0,7 \text{ кг / м}^3;$$

G_0 – кількість вуглекислоти, що залишається в установці після закінчення її роботи; $G_0 = 0,2 \cdot G_{\text{г}}$, кг

Далі виведемо рівняння (6.2).

$$G_{\text{г}} = G_{\text{в}} \cdot W_{\text{п}} \cdot K_{\text{в}} + 0,2 \cdot G_{\text{г}}, \quad (6.2)$$

$$0,8 \cdot G_r = G_b \cdot W_n \cdot K_b,$$

$$G_r = G_b \cdot W_n \cdot K_b / 0,8 = (0,7 + 550 + 1,5) / 0,8 = 722 \text{ кг}.$$

Розрахуємо необхідну кількість робочих балонів N_6 із вуглекислотою в установці за рівнянням (6.3):

$$N_6 = G_r / V_6 \cdot \rho \cdot \alpha_n, \text{ шт.}, \quad (6.3)$$

де V_6 – об'єм балону (л). Приймаємо $V_6 = 25$ л, при цьому в балоні міститься 15,6 кг вуглекислоти;

ρ – питома вага вуглекислоти, $\rho = 0,625$ кг/л;

α_n – коефіцієнт наповнення, $\alpha_n = 0,85$.

$$N_6 = 722 / (25 \cdot 0,625 \cdot 0,85) = 54,36 \approx 55 \text{ шт.}$$

Розрахуємо пропускну здатність трубопроводу.

$$F = 10^{-4} \cdot \sqrt{\frac{P \cdot \gamma}{2 \cdot A \cdot L}} \text{ (кг/с)}, \quad (6.4)$$

де P – питомий тиск вуглекислоти на початку трубопроводу (в балоні), Н/м²,

$$P = 49 \cdot 10^5 \text{ Н/м}^2;$$

A – питомий опір трубопроводу при діаметрі труби 40 мм, ($A = 0,035$);

γ – щільність вуглекислоти на початку трубопроводу (в балоні),

$$\gamma = 2900 \text{ кг/м}^3.$$

Виконаємо розрахунок пропускну здатності трубопроводу (6.5).

$$F = 10^{-4} \cdot \sqrt{\frac{P \cdot \gamma}{2 \cdot A \cdot L}} = 10^{-4} \cdot \sqrt{\frac{4,9 \cdot 10^6 \cdot 2900}{2 \cdot 0,035 \cdot 80}} = 5,03 \text{ кг/с}. \quad (6.5)$$

Таким чином, необхідна маса вуглекислоти $G_r = 722$ кг, кількість робочих балонів $N_6 = 55$ шт., пропускна здатність трубопроводу $F = 5,03$ кг / с.

6.4.2 Заходи з цивільного захисту

Організація навчання працюючого та непрацюючого населення діям у надзвичайних ситуаціях.

Надзвичайна ситуація – порушення нормальних умов життя і діяльності людей на об'єкті або території, спричинене аварією, катастрофою, стихійним лихом, епідемією, епізоотією, епіфітотією, великою пожежею, застосуванням засобів ураження, що призвели або можуть призвести до людських і матеріальних втрат (Закон України «Про цивільну оборону»).

Порядок здійснення навчання населення діям у надзвичайних ситуаціях затверджено відповідною постановою Кабінету Міністрів України від 26 червня 2013 р. № 444. Цей Порядок визначає механізм організації навчання населення діям у надзвичайних ситуаціях (далі – навчання населення), його структуру, види та форми. Навчання населення здійснюється:

- за місцем роботи – працюючого населення;
- за місцем навчання – дітей дошкільного віку, учнів та студентів;
- за місцем проживання – непрацюючого населення.

Організація навчання населення покладається на працюючого та непрацюючого – на ДСНС, Раду міністрів Автономної Республіки Крим, місцеві державні адміністрації, органи місцевого самоврядування, а дітей дошкільного віку, учнів та студентів – на МОН.

Навчально-методичне забезпечення навчання населення здійснюється ДСНС разом з МОН. Навчання населення складається з:

- навчання безпосередньо на підприємствах, в установах та організаціях;

- навчання за межами підприємств, установ та організацій керівного складу і фахівців з питань цивільного захисту та пожежної безпеки;
- практичної підготовки під час проведення спеціальних об'єктових навчань і тренувань з питань цивільного захисту;
- навчання під час здобуття відповідного освітнього рівня у навчальних закладах системи освіти;
- самостійного вивчення інформації про дії в умовах надзвичайних ситуацій.

Навчання працюючого населення здійснюється безпосередньо на підприємстві, в установі та організації згідно з програмами підготовки працівників до дій у надзвичайних ситуаціях, а також під час проведення спеціальних об'єктових навчань і тренувань з питань цивільного захисту. Програми підготовки працівників до дій у надзвичайних ситуаціях розробляються і затверджуються підприємствами, установами, організаціями на підставі програм та організаційно-методичних вказівок з підготовки населення до дій у надзвичайних ситуаціях, що розробляються і затверджуються ДСНС, Радою міністрів Автономної Республіки Крим, місцевими державними адміністраціями, органами місцевого самоврядування. Програми навчання з питань пожежної безпеки погоджуються із ДСНС.

Програми підготовки працівників до дій у надзвичайних ситуаціях поділяються на:

- загальної підготовки працівників підприємств, установ та організацій;
- спеціальної підготовки працівників, що входять до складу спеціалізованих служб і формувань цивільного захисту;
- додаткової підготовки з техногенної безпеки працівників об'єктів підвищеної небезпеки;
- пожежно-технічного мінімуму для працівників, зайнятих на роботах з підвищеною пожежною небезпекою;
- прискореної підготовки працівників до дій в особливий період.

Підготовка працівників до дій у надзвичайних ситуаціях передбачає:

- за програмою загальної підготовки працівників підприємств, установ та організацій - вивчення інформації, що міститься у планах реагування на надзвичайні ситуації, про дії в умовах загрози і виникнення надзвичайної ситуації, а також оволодіння навичками надання першої допомоги потерпілим, користування засобами індивідуального і колективного захисту;

- за програмою спеціальної підготовки працівників, що входять до складу спеціалізованих служб і формувань цивільного захисту, ознайомлення з обов'язками, навичками користування та матеріальною частиною техніки, приладів і табельного майна таких служб і формувань, засобами захисту, вивчення порядку приведення їх у готовність, проведення рятувальних та інших невідкладних робіт;

- за програмою додаткової підготовки з техногенної безпеки працівників об'єктів підвищеної небезпеки - поглиблення знань з питань техногенної безпеки, джерел небезпеки, що за певних обставин можуть спричинити виникнення надзвичайної ситуації на об'єкті підвищеної небезпеки, та небезпечних речовин, що виготовляються, переробляються, зберігаються чи транспортуються на його території;

- за програмою пожежно-технічного мінімуму для працівників, зайнятих на роботах з підвищеною пожежною небезпекою, - підвищення рівня загальних пожежно-технічних знань, вивчення правил пожежної безпеки з урахуванням особливостей виробництва, ознайомлення з протипожежними заходами та діями у разі виникнення пожежі, оволодіння навичками використання наявних засобів пожежогасіння;

- за програмою прискореної підготовки працівників до дій в особливий період – навчання способам захисту від наслідків надзвичайних ситуацій, спричинених застосуванням засобів ураження в особливий період, що здійснюється підприємствами, установами та організаціями, які продовжують роботу у воєнний час, і розпочинається одночасно з уведенням в дію планів цивільного захисту на особливий період.

Навчання працівників на підприємстві, в установі та організації здійснюється шляхом:

- курсового навчання, що передбачає формування навчальних груп і здійснюється в навчальних класах або на об'єктах навчально-виробничої бази підприємства, установи та організації;
- індивідуального навчання, що передбачає вивчення теоретичного матеріалу самостійно та у формі консультацій з керівниками навчальних груп або іншими особами.

Навчальні групи комплектуються переважно з працівників, що входять до складу спеціалізованих служб і формувань цивільного захисту.

На підприємствах, в установах та організаціях із чисельністю працівників 50 і менше осіб навчання може здійснюватися шляхом проведення інструктажів за програмою загальної підготовки працівників, які проводяться особами з питань цивільного захисту, призначеними в межах штатної чисельності суб'єкта господарювання [35].

ВИСНОВКИ

В результаті виконання дипломної кваліфікаційної роботи магістра було досліджено методи розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках.

Було розглянуто наступні системи розпізнавання: Xeoma Mask Detector PRO, Aerialtronics Face Mask Detection Software, Hikvision Mask Detection. Розглянуті системи мають певні недоліки, а саме: високу ціну, складність в використанні, неможливість використання цих систем в умовах виробництва, необхідність потужних апаратних ресурсів, недостатню продуктивність та достовірність результатів в умовах браку освітленості. Тому виникла необхідність в розробці програмного забезпечення, що дозволяє розпізнавати об'єкти заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках.

В роботі було побудовано технологію розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках, яка складається з шістьох етапів.

Також було виконано дослідження програмних засобів розробки системи: була обрана мова програмування Python та бібліотека комп'ютерного зору з відкритим вихідним кодом OpenCV.

Наведено опис файлів розробленої системи, процесу тренування або навчання моделі та розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві у реальному часі у відеопотоках.

Було описано та досліджено функціональні можливості розробленої системи. Ці функціональні можливості були порівняні з іншими відомими системами, що працюють у сферах визначення об'єктів заданого класу на виробництві або підприємстві за допомогою алгоритмів комп'ютерного зору або машинного навчання. Для оцінки показника достовірності розпізнавання

об'єктів заданого класу на виробництві було проведено ряд експериментів. В ході цих експериментів встановлено залежність показника достовірності від умов, у яких цей показник розраховується. На нього впливають такі параметри як якість освітленість робочої зони, кут ведення відеофіксації та фізична відстань від відео пристрою до потенційного об'єкта розпізнавання. Результати експериментів показали достатньо добрий показник достовірності розпізнавання у розробленій технології (76,8 %).

Також було запропоновано можливе поліпшення технології за рахунок додавання протоколу Modbus TCP/IP для сигналізації результатів розпізнавання на сервер.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Магістерська робота «Дослідження та програмна реалізація методів виявлення та відстеження об'єктів у відеопотоці»/ Твердохліб Є.Р. – Запоріжжя: ЗНТУ, 2018. – 133 с.
2. David A. Forsyth. Computer Vision: A Modern Approach / David A. Forsyth, Jean Ponce // Prentice Hall. – 2002. – 693 pp.
3. Freund Yoav. A decision-theoretic generalization of on-line learning and an application to boosting / Yoav Freund, Robert E Schapire // Journal of Computer and System Sciences. – Vol. 55. – 1997. – PP. 119-139.
4. AdaBoost [Electronic resource]. – Access mode: <https://en.wikipedia.org/wiki/AdaBoost>
5. Boosting with AdaBoost and Gradient Boosting [Electronic resource]. – Access mode: <https://medium.com/diogo-menezes-borges/boosting-with-adaboost-and-gradient-boosting-9cbab2a1af81>
6. Viola P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proc. of CVPR. – 2001. – Vol.1. – PP. 511-518.
7. Viola P. Robust real-time face detection / P. Viola, M.J. Jones // International Journal of Computer Vision. – 2004. – V. 57, № 2. – PP. 137–154.
8. С. О. Субботін Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник. Запоріжжя: ЗНТУ, 2008. – 341 с.
9. Mask Detector: AI-based detection of people wearing a facial mask [Electronic resource]. – Access mode: <https://felenasoft.com/xeoma/en/articles/mask-detector/>
10. AI powered face mask detection software [Electronic resource]. – Access mode: <https://www.aerialtronics.com/en/products/face-mask-detection-software>
11. Mask Detection Solution Monitor the use of protective face masks [Electronic resource]. – Access mode:

<https://www.hikvision.com/europe/solutions/solutions-by-application/mask-detection/>

12. Bjarne Stroustrup. The C++ Programming Language (Third ed.) / Stroustrup, Bjarne // Addison-Wesley. – 1997. – 912 pp.

13. Python [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Python>.

14. Python overview [Electronic resource]. – Access mode: <https://wiki.python.org/moin/BeginnersGuide/Overview>

15. Scala [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Scala>

16. Scala (programming language) [Electronic resource]. – Access mode: [https://en.wikipedia.org/wiki/Scala_\(programming_language\)](https://en.wikipedia.org/wiki/Scala_(programming_language))

17. Документація фреймворку OpenCV [Електронний ресурс]. – Режим доступу: <http://docs.opencv.org/>.

18. Google Vision API [Electronic resource]. – Access mode: <https://cloud.google.com/vision/docs>

19. Computer Vision platform using Python [Electronic resource]. – Access mode: <http://simplecv.org/>

20. I just open sourced my personal imutils package: A series of OpenCV convenience functions [Electronic resource]. – Access mode: <https://www.pyimagesearch.com/2015/02/02/just-open-sourced-personal-imutils-package-series-opencv-convenience-functions/>

21. TensorFlow [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/TensorFlow>

22. Правила улаштування електроустановок [Текст] : ПУЕ-2017. – На заміну ПУЕ-86. – [Чинний від 2017-08-21]. – Київ: Міненерговугілля України, 2017. – 617 с.

23. ДСТУ Б В.2.5-82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. – [Чинний від 2017-04-01]. – Київ: ДП "УкрНДНЦ", 2016. – 109 с.

24. ДСТУ ISO 13406-2:2006 Ергономічні вимоги до роботи з відеотерміналами з плоским екраном. Частина 2. Ергономічні вимоги до дисплеїв з плоским екраном (ISO 13406-2:2001, IDT) [Текст]. – [Уведено вперше; чинний від 2007-10-01]. – Київ: ДП «УкрНДНЦ», 2007. – 117 с.

25. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. НПАОП 0.00-7.15-18. Державні санітарні правила і норми [Текст]. – [Чинний від 2018-02-14]. – Харків: Форт, 2018. – 48 с.

26. Курс лекцій по «Эргономике». Ю.М. Мадиевский. – Под ред. д-ра. техн. наук., проф. Березуцкого В.В. – Харків: НТУ «ХПИ», 2015. – 294 с.

27. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – [Чинний від 1999-12-01]. – Харків : Форт, 2000. – 24 с.

28. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування [Текст]. – [Чинний від 2014-01-01]. – Київ: Мінрегіон України, 2013. – 141 с.

29. ДБН В.2.5-28:2018. Природне та штучне освітлення [Текст]. – [Чинний від 2019-03-01]. – К. : Мінрегіон України, 2019. – 137 с.

30. Грибан В. Охорона праці / В. Грибан, А. Негодченко. – Київ: Центр навчальної літератури, 2017. – 280 с.

31. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Текст]. – [Чинний від 1999-12-01]. – К. : МОЗ України, 1999. – 24 с.

32. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – [Чинний від 2017-01-31]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

33. ДБН В.2.5-56:2014 Системи протипожежного захисту[Текст]. – [Чинний від 2015-07-01]. – Київ: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2015. – 127 с.

34. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою: ДСТУ Б В.1.1-36:2016. –

[Чинний від 2017-01-01]. – К.: Міненергобуд України, 2017. – 86 с. – (Національний стандарт України).

35. Основи цивільного захисту: навч. посіб. / В.О. Васійчук, В.Є. Гончарук, С.І. Качан, С.М. Мохняк. – Львів, 2010. – 384 с.

ДОДАТОК А
Текст програми

A.1 Текст файла main.py

```

import cv2

posClassifier = cv2.CascadeClassifier('cascade_pos.xml')
negClassifier = cv2.CascadeClassifier('cascade_neg.xml')

videoCapture = cv2.VideoCapture(0)
if (videoCapture.isOpened() == False):
    print('Error opening video file!')

while (videoCapture.isOpened()):

    successFlag, videoFrame = videoCapture.read()

    if successFlag:
        grayFrame = cv2.cvtColor(videoFrame, cv2.COLOR_BGR2GRAY)

        positives = posClassifier.detectMultiScale(grayFrame, 1.05, 60)
        negatives = negClassifier.detectMultiScale(grayFrame, 1.05, 12)

        for (x, y, w, h) in positives:
            cv2.rectangle(videoFrame, (x, y), (x + w, y + h), (0, 255, 0), 2)
            cv2.putText(videoFrame, 'Correct', (x, y - 5),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

            for (x, y, w, h) in negatives:
                cv2.rectangle(videoFrame, (x, y), (x + w, y + h), (0, 0, 255), 2)
                cv2.putText(videoFrame, 'Incorrect', (x, y - 5),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

            cv2.imshow('detection', videoFrame)

            if (cv2.waitKey(25) & 0xFF == ord('q')):
                break

videoCapture.release()
cv2.destroyAllWindows()

```

A.2 Текст файлу fix.py

```

from os import listdir, rename
from os.path import abspath, join
import cv2

def renameFiles(path):
    files = listdir(path)

    for count, f in enumerate(files):
        newFileName = abspath('{} /temp.{}.bmp'.format(path, count))
        currentFileName = abspath(join(path, f))
        rename(currentFileName, newFileName)
        print('{} -> {}'.format(currentFileName, newFileName))

    files = listdir(path)

    for count, f in enumerate(files):
        newFileName = abspath('{} /{}.bmp'.format(path, count))
        currentFileName = abspath(join(path, f))
        rename(currentFileName, newFileName)
        print('{} -> {}'.format(currentFileName, newFileName))

def fixGoodDatFile(datPath, filesPath):
    newContent = ""
    with open(datPath, 'r') as datFile:
        for line in datFile.readlines():
            imgPath = line.split(' ')[0]
            if (imgPath.strip() != ""):
                image = cv2.imread(imgPath)
                if image is None:
                    print('Error: ', imgPath)
                    exit()
                h, w, c = image.shape
                newContent += '{} 1 0 0 {} {} \n'.format(imgPath, w, h)

    with open(datPath, 'w') as datWriter:
        datWriter.write(newContent)

def fixBadDatFile(datPath, filesPath):
    newContent = ""
    files = listdir(filesPath)
    for count, f in enumerate(files):
        newContent += 'bad/{} \n'.format(f)

```

```

with open(datPath, 'w') as datWriter:
    datWriter.write(newContent)

```

```

goodDat = './good.dat'
badDat = './bad.dat'
pathGood = './good'
pathBad = './bad'

```

```

renameFiles(pathGood)
renameFiles(pathBad)

```

```

fixGoodDatFile(goodDat, pathGood)
fixBadDatFile(badDat, pathBad)

```

A.3 Текст файлу **trainer.py**

```

import cv2
import os
import sys

class FrameProvider:
    def __init__(self, source):
        self.equipClassifier = cv2.CascadeClassifier('cascade.xml')
        self.__source = source

    def capture(self):
        while (self.__source.isOpened()):
            successFlag, videoFrame = self.__source.read()
            if successFlag:
                grayFrame = cv2.cvtColor(videoFrame,
cv2.COLOR_BGR2GRAY)
                faces = self.equipClassifier.detectMultiScale(grayFrame, 1.05,
40)

                detectFrame = grayFrame.copy()
                for (x, y, w, h) in faces:
                    cv2.rectangle(detectFrame, (x, y), (x + w, y + h), (255, 0, 0), 2)

                cv2.imshow('capture', detectFrame)

                key = (cv2.waitKey(33) & 0xFF)
                if (key == ord('c')):

```

```

        return grayFrame

    elif (key == ord('q')):
        exit()

class DataController:
    def __init__(self, configFile, goodDataFile, badDataFile, mode):
        self.goodDataFile = goodDataFile
        self.badDataFile = badDataFile
        self.configFile = configFile
        self.mode = mode

    def readConfig(self):
        dataFile = self.goodDataFile if self.mode == '-g' else self.badDataFile
        if (not os.path.isfile(dataFile)):
            print('Cannot find data file')
            exit()

        try:
            with open(self.configFile) as configContent:
                for configLine in configContent:
                    parts = configLine.split();
                    type = parts[0]
                    if (type == 'Good'):
                        self.goodCount = int(parts[1])
                    elif (type == 'Bad'):
                        self.badCount = int(parts[1])
        except:
            print('Error while opening config file')
            exit()

        print('Success. Good/Bad', self.goodCount, self.badCount)

    def appendImage(self, image):
        if (self.mode == '-g'):
            try:
                filename = 'good/{}.bmp'.format(self.goodCount)
                cv2.imwrite('./data/' + filename, image)
                with open(self.goodDataFile, 'a') as writer:
                    writer.write('{} 1 0 0 {} {} \n'.format(filename, image.shape[1],
image.shape[0]))
                self.goodCount += 1
            except Exception as ex:
                print('Error writing to file', str(ex))
                exit()

```

```

elif (self.mode == '-b'):
    try:
        filename = 'bad/{}.bmp'.format(self.badCount)
        cv2.imwrite('./data/' + filename, image)
        print(image.shape)
        with open(self.badDataFile, 'a') as writer:
            writer.write('{}\n'.format(filename))
        self.badCount += 1
    except Exception as ex:
        print('Error writing to file', str(ex))
        exit()

    with open(self.configFile, 'w') as writer:
        writer.write('Good      {}\nBad      {}'.format(self.goodCount,
self.badCount))

```

```

class Selector:
    __wndName = 'select'

    def __init__(self, frame):
        self.startPoint = (0, 0)
        self.stopPoint = (0, 0)
        self.selectionMode = False
        self.frame = frame

    def __onMouse(self, event, x, y, flags, params):
        if event == cv2.EVENT_LBUTTONDOWN:
            self.selectionMode = True
            self.startPoint = (x, y)
            print('Starting in point: ', self.startPoint)

            elif event == cv2.EVENT_LBUTTONUP:
                self.selectionMode = False
                self.stopPoint = (x, y)
                print('Stop: ', self.startPoint, self.stopPoint)

            elif event == cv2.EVENT_MOUSEMOVE:
                if (self.selectionMode):
                    self.stopPoint = (x, y)

    def __show(self):
        img = self.frame.copy()
        cv2.namedWindow(self.__wndName)

```

```

cv2.setMouseCallback(self.__wndName, self.__onMouse)
cv2.rectangle(img, self.startPoint, self.stopPoint, (255, 0, 0), 2)
cv2.imshow(self.__wndName, img)

def select(self):
    while (True):
        self.__show();
        key = cv2.waitKey(33) & 0xFF
        if (key == ord('s')):
            cv2.destroyWindow(self.__wndName)
            return (self.startPoint, self.stopPoint)

        elif (key == ord('r')):
            self.startPoint = (0, 0)
            self.stopPoint = (0, 0)
            self.selectionMode = False

        elif (key == ord('q')):
            cv2.destroyWindow(self.__wndName)
            return -1

def main():
    configFilePath = './data/trainer.conf'
    badDataFilePath = './data/bad.dat'
    goodDataFilePath = './data/good.dat'

    currentMode = sys.argv[1]
    if (currentMode != '-g' and currentMode != '-b'):
        print('Invalid value for argument <mode> provided, please specify -
g(good) or -b(bad)')
        exit()

    webcam = cv2.VideoCapture(0)
    if (webcam.isOpened() == False):
        print('Error opening video file!')

    frameProvider = FrameProvider(webcam)
    dataController = DataController(configFilePath, goodDataFilePath,
badDataFilePath, currentMode)
    dataController.readConfig()

    while(1):
        frame = frameProvider.capture()
        selector = Selector(frame)
        region = selector.select()

```

```
        if (region != -1):
            print('Selected:', region)
            cropped = frame[region[0][1]:region[1][1],
region[0][0]:region[1][0]]
            dataController.appendImage(cropped)

    videoCapture.release()
    cv2.destroyAllWindows()

main()
```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька
політехніка»

Магістерська робота на тему

Дослідження та програмна реалізація методів розпізнавання об'єктів на виробництві

Виконав

ст.гр. КНТ-129м

Шемберко Ю.А.

Керівник:

к.ф.-м.н., доцент

Терещенко Е.В.

Рисунок Б.1 – Слайд №1

Мета роботи, об'єкт та предмет дослідження

Мета роботи – розробка системи машинного зору, що дозволяє виконувати розпізнавання об'єктів заданого класу: маски, каски та захисних окулярів у людині, що працює на виробництві в реальному часі у відеопотоках з високим ступенем достовірності розпізнавання.

Об'єкт дослідження – процеси розпізнавання об'єктів у відеопотоці.

Предмет дослідження – методи та системи розпізнавання об'єктів у відеопотоці.

2

Рисунок Б.2 – Слайд №2

Задачі дослідження

- дослідити методи розпізнавання об'єктів у відеопотоці на виробництві;
- виконати дослідження програмних засобів для розпізнавання об'єктів у відеопотоці;
- розробити систему розпізнавання об'єктів заданого класу у відеопотоці;
- виконати експериментальне дослідження розробленої системи.

3

Рисунок Б.3 – Слайд №3

Програмні засоби для розпізнавання об'єктів у відеопотоці

- Xeoma Mask Detector PRO;
- Aerialtronics Face Mask Detection Software;
- Hikvision Mask Detection.

4

Рисунок Б.4 – Слайд №4

Огляд бібліотек для розпізнавання об'єктів

- Google Vision Application Programming Interface
- Simple CV
- Imutils
- TensorFlow
- **Open Source Computer Vision Library**

5

Рисунок Б.5 – Слайд №5

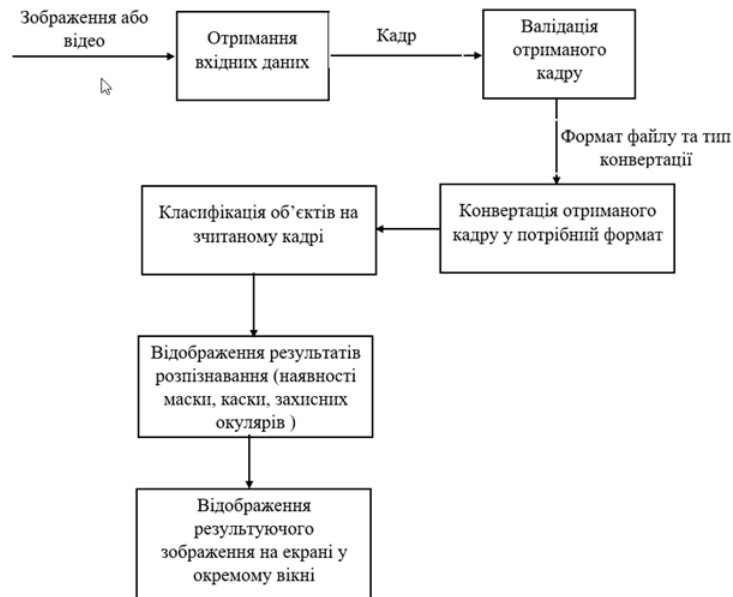
Використані засоби при розробці

- Мова програмування - Python
- Засоби обробки зображень - OpenCV: Image Processing.
- Засоби відтворення користувацького інтерфейсу - OpenCV: High-level GUI.
- Засоби розпізнавання об'єктів - OpenCV: Object Detection.

6

Рисунок Б.6 – Слайд №6

Технологія розпізнавання об'єктів на виробництві



7

Рисунок Б.7 – Слайд №7

Опис процесу розпізнавання

Етап 1. Ініціалізація:

- імпорт бібліотеки OpenCV до виконуваного скрипт-файлу;
- завантаження класифікаторів з файлу за допомогою засобів OpenCV (CascadeClassifier);
- захоплення та валідація відео-потoku. (VideoCapture, VideoCapture.isOpened()).

Етап 2. Розпізнавання:

- зчитування кадру з відеопотоку (VideoCapture.read());
- валідація отриманого кадру: перша змінна з кортежу, що повертає VideoCapture.read() – це флаг, що сигналізує про успішне зчитування кадру;
- конвертація отриманого кадру у форматі BGR (blue, green, red) у зображення у відтінках сірого (gray) за допомогою функції cvtColor(), у яку передаємо кадр та бажаний тип конвертації, у даному випадку COLOR_BGR2GRAY;
- класифікація об'єктів на зчитаному кадрі. За це відповідає метод CascadeClassifier.detectMultiScale(). Викликається для кожного класифікатора.
- відображення результатів розпізнавання. Опрацьовуємо результати роботи класифікатора: малюємо прямокутник зеленого (для правильних об'єктів) або червоного (для неправильних) на кадрі (функція rectangle()), та пишемо текст відповідний текст (функція putText());
- відображаємо результуюче зображення на екрані у окремому вікні (imshow()).

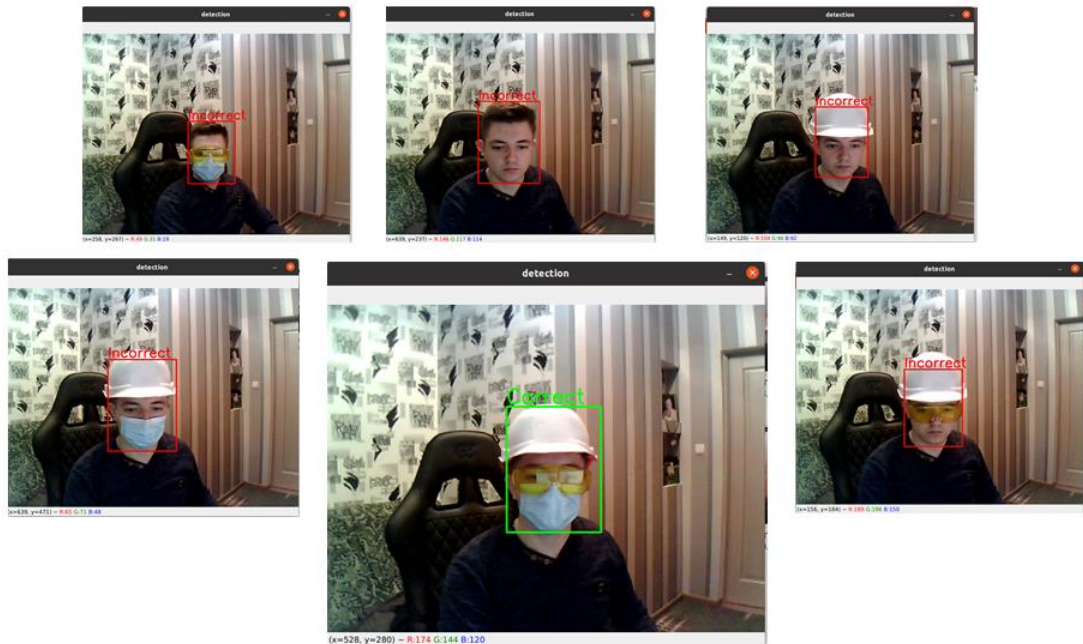
Етап 3. Звільнення ресурсів:

- videoCapture.release() – закриття потоку відео.
- destroyAllWindows() – знищення створених за допомогою OpenCV вікон.

8

Рисунок Б.8 – Слайд №8

Приклад роботи програми



9

Рисунок Б.9 – Слайд №9

Призначення розробленої системи

Розроблена система дозволяє розпізнавати об'єкти на виробництві за допомогою двох каскадних класифікаторів:

- розпізнає “коректні” об'єкти: у контексті задачі - це людина, у якої фізично наявні маска, шолом та захисні окуляри
- Розпізнає “некоректні” об'єкти: навпаки, людина, у якої відсутній будь-який з елементів.

10

Рисунок Б.10 – Слайд №10

Порівняльний аналіз функціональних можливостей системи з відомими аналогами

Системи\ Параметри	Xeoma Mask Detector PRO	Aerialtronics Face Mask Detection Software	Hikvision Mask Detection	Розроблена система
Пошук наявності медичної маски	+	+	+	+
Відстеження каска	–	–	–	+
Відстеження захисних окулярів	–	–	–	+
Робота в режимі реального часу	+	+	+	+
Портативність	–	–	–	+

11

Рисунок Б.11 – Слайд №11

Оцінка показника достовірності розпізнавання

№	Освітленість, лк (люкс)	Кут, градус	Відстань, м	Кількість правильно розпізнаних об'єктів	Показник достовірності, %
1	1600	90	5	153	15,3
2	5800	135	2	396	39,6
3	19000	180	1	768	76,8

12

Рисунок Б.12 – Слайд №12

Практична цінність

Створене програмне забезпечення, може бути використане в якості модуля для розробки автоматизованої системи контролю заходів безпеки на виробництві.

У перспективі, можна покращити показник достовірності розпізнавання, інтегрувати модуль сигналізації на основі ModBus TCP.

13

Рисунок Б.13 – Слайд №13

Висновки

Під час виконання магістерської роботи було проведено дослідження методів розпізнавання об'єктів на виробництві.

Було створено програмне забезпечення, що здатне розпізнавати об'єкти на зображенні з відносно високим показником достовірності (76,8 %).

В магістерській роботі було розглянуто сучасні технології та алгоритми розпізнавання об'єктів на зображенні.

14

Рисунок Б.14 – Слайд №14