ДОДАДОК А

Програмний код для кваліфікаційної роботи

Лістинг частини коду для збору фото в базу даних.

script.py

```
import cv2
import os
import time
import mediapipe as mp
from gtts import gTTS
import pygame

# === Налаштування ===
user_name = "Andrii"
dataset_path = r"C:\Users\noryk\PyCharmMiscProject\face_recognition_project\dataset"
user_path = os.path.join(dataset_path, user_name)
os.makedirs(user_path, exist_ok=True)

images_per_position = 51
positions = [("Подивись прямо", "center"),
             ("Поверни голову вліво", "left"),
             ("Поверни голову вправо", "right")]
image_size = (200, 200)

# === Голосова підказка ===
def speak_uk(text):
    tts = gTTS(text=text, lang='uk')
    filename = "temp_voice.mp3"
```

```
            tts.save(filename)

            pygame.mixer.init()
            pygame.mixer.music.load(filename)
            pygame.mixer.music.play()
            while pygame.mixer.music.get_busy():
                pygame.time.Clock().tick(10)

            pygame.mixer.quit()
            os.remove(filename)

        # === Камера ===
        cap = cv2.VideoCapture(0)
        if not cap.isOpened():
            print(" Камера не відкрилася.")
            exit()
        print(" Камера запущена. Натисніть 'q' для виходу.")

        # === MediaPipe ===
        mp_face_detection = mp.solutions.face_detection

        with mp_face_detection.FaceDetection(model_selection=0,
min_detection_confidence=0.7) as face_detection:
            total_images = 0

            for instruction, pos in positions:
                speak_uk(instruction)
                print(f" 🎙 {instruction}")
                count = 0
```

```
            while count < images_per_position:
                ret, frame = cap.read()
                if not ret:
                    continue

                image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                results = face_detection.process(image_rgb)

                if results.detections:
                    for detection in results.detections:
                        bbox = detection.location_data.relative_bounding_box
                        ih, iw, _ = frame.shape
                        x = int(bbox.xmin * iw)
                        y = int(bbox.ymin * ih)
                        w = int(bbox.width * iw)
                        h = int(bbox.height * ih)

                        if x < 0 or y < 0 or x + w > iw or y + h > ih:
                            continue

                        face_img = frame[y:y + h, x:x + w]
                        if face_img.size == 0:
                            continue

                        face_img_gray = cv2.cvtColor(face_img,
cv2.COLOR_BGR2GRAY)
                        face_resized = cv2.resize(face_img_gray, image_size)

                        img_path = os.path.join(user_path, f"{pos}_{count}.jpg")
                        cv2.imwrite(img_path, face_resized)
```

```
                print(f" Збережено: {img_path}")
                count += 1
                total_images += 1
                time.sleep(1)

                cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

            else:
                cv2.putText(frame, "Обличчя не знайдено", (30, 40),
                            cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

            cv2.imshow(" Збір даних (Натисніть 'q' для виходу)", frame)

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

cap.release()
cv2.destroyAllWindows()
final_message = f"Готово. Збережено {total_images} зображень."
print(f" {final_message}")
speak_uk(final_message)
import os
import cv2
import numpy as np

# === Шляхи ===
dataset_path = "dataset"
trainer_save_path = "trainer.yml"

# === Ініціалізація ===
```

```
recognizer = cv2.face.LBPHFaceRecognizer_create()
face_samples = []
ids = []
label_map = {}  # ім’я -> ID
current_id = 0

# === Обхід усіх підпапок (користувачів) ===
for user_name in os.listdir(dataset_path):
    user_folder = os.path.join(dataset_path, user_name)
    if not os.path.isdir(user_folder):
        continue

    if user_name not in label_map:
        label_map[user_name] = current_id
        current_id += 1
    user_id = label_map[user_name]

    # === Зчитування зображень ===
    for image_name in os.listdir(user_folder):
        img_path = os.path.join(user_folder, image_name)
        img = cv2.imread(img_path)
        if img is None:
            print(f" Неможливо зчитати: {img_path}")
            continue

        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        face_samples.append(gray)
        ids.append(user_id)

# === Навчання ===
```

```
    print(" Навчання моделі...")
    recognizer.train(face_samples, np.array(ids))
    recognizer.save(trainer_save_path)
    print(f" Модель збережена: {trainer_save_path}")

    # === Збереження відповідності ID -> ім’я ===
    labels_file = "labels.txt"
    with open(labels_file, "w", encoding="utf-8") as f:
        for name, idx in label_map.items():
            f.write(f"{idx}:{name}\n")

    print(f" Відповідності ID -> ім’я збережено у {labels_file}")
    import cv2
    import os
    import mediapipe as mp

    dataset_path =
r"C:\Users\noryk\PyCharmMiscProject\face_recognition_project\dataset"
    model_path = "trainer.yml"
    image_size = (200, 200)

    # === Завантаження моделі ===
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read(model_path)
    print(" Модель завантажена. Починаємо розпізнавання...")

    # === Імена користувачів ===
    label_ids = {}
    for i, name in enumerate(sorted(os.listdir(dataset_path))):
        label_ids[i] = name
```

```
        # === Ініціалізація MediaPipe ===
        mp_face_detection = mp.solutions.face_detection

        cap = cv2.VideoCapture(0)
        if not cap.isOpened():
            print(" Камера не відкрилась.")
            exit()

        with mp_face_detection.FaceDetection(model_selection=0,
min_detection_confidence=0.7) as face_detection:
            while True:
                ret, frame = cap.read()
                if not ret:
                    continue

                image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
                results = face_detection.process(image_rgb)

                if results.detections:
                    for detection in results.detections:
                        bbox = detection.location_data.relative_bounding_box
                        ih, iw, _ = frame.shape
                        x = int(bbox.xmin * iw)
                        y = int(bbox.ymin * ih)
                        w = int(bbox.width * iw)
                        h = int(bbox.height * ih)

                        if x < 0 or y < 0 or x + w > iw or y + h > ih:
                            continue
```

```
        face = frame[y:y + h, x:x + w]
        if face.size == 0:
            continue

        face_gray = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY)
        face_resized = cv2.resize(face_gray, image_size)

        id_, conf = recognizer.predict(face_resized)

        if conf < 80:
            name = label_ids.get(id_, "Невідомо")
            color = (0, 255, 0)
            label = f"{name} ({round(conf, 1)}%)"
        else:
            name = "Невідомо"
            color = (0, 0, 255)
            label = name

        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, label, (x, y - 10),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.9, color, 2)

else:
    cv2.putText(frame, "Обличчя не знайдено", (30, 40),
                cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

cv2.imshow(" Розпізнавання облич (натисніть 'q' для виходу)", frame)

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

```
            break

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

ДОДАТОК Б

Демонстраційний матеріал

Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій

Кафедра КІТАР

Спеціальність 151 Автоматизація та комп'ютерно інтегровані технології

Кваліфікаційна робота на тему:

«Розроблення автоматичної підсистеми моніторингу та сповіщення про несанкціонований доступ із використанням технології комп'ютерного зору»

Виконав:
студент групи АКТАКІТ-21-1
Норик А. О.

Керівник роботи:
ас. каф. КІТАР
Слюсар А. П.

Мета і завдання роботи

Мета проєкту – створити автоматизовану підсистему, що забезпечує якісний контроль санкціонованого доступу до приміщення з використанням технологій комп'ютерного зору.

Об'єкт розробки – процес контролю доступу до приміщень.

Предмет розробки – підсистема моніторингу та сповіщення про несанкціонований доступ на базі комп'ютерного зору.

Завдання:

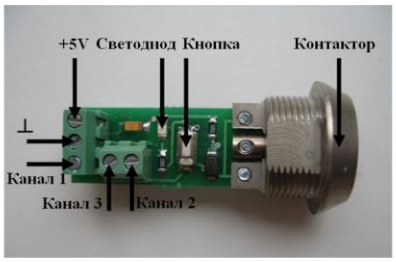
- провести аналіз існуючих методів, засобів та автоматизованих систем контролю доступу до приміщень;
- розробити структурну та алгоритмічну схему підсистеми моніторингу на основі технологій комп'ютерного зору;
- створити програмне забезпечення для виявлення, ідентифікації та сповіщення про несанкціонований доступ;
- розглянути питання охорони праці.

Актуальність роботи

- У разі спроби проникнення до приміщення сторонньої особи без відповідного дозволу, оперативне виявлення інциденту дозволяє своєчасно зреагувати та запобігти матеріальним збиткам чи витоку конфіденційної інформації. Моніторинг з використанням комп'ютерного зору дозволяє автоматично фіксувати факти доступу, ідентифікувати осіб та визначати потенційні загрози без участі людини.
- Безперервний контроль доступу є критично важливим для підприємств, установ і об'єктів із підвищеними вимогами до безпеки. Підвищення ефективності таких систем можливе за рахунок автоматизації процесів виявлення порушень та сповіщення.
- Автоматизована підсистема моніторингу та сповіщення забезпечує високу точність, знижує навантаження на персонал охорони, зменшує вплив людського фактора та підвищує загальний рівень безпеки об'єкта.

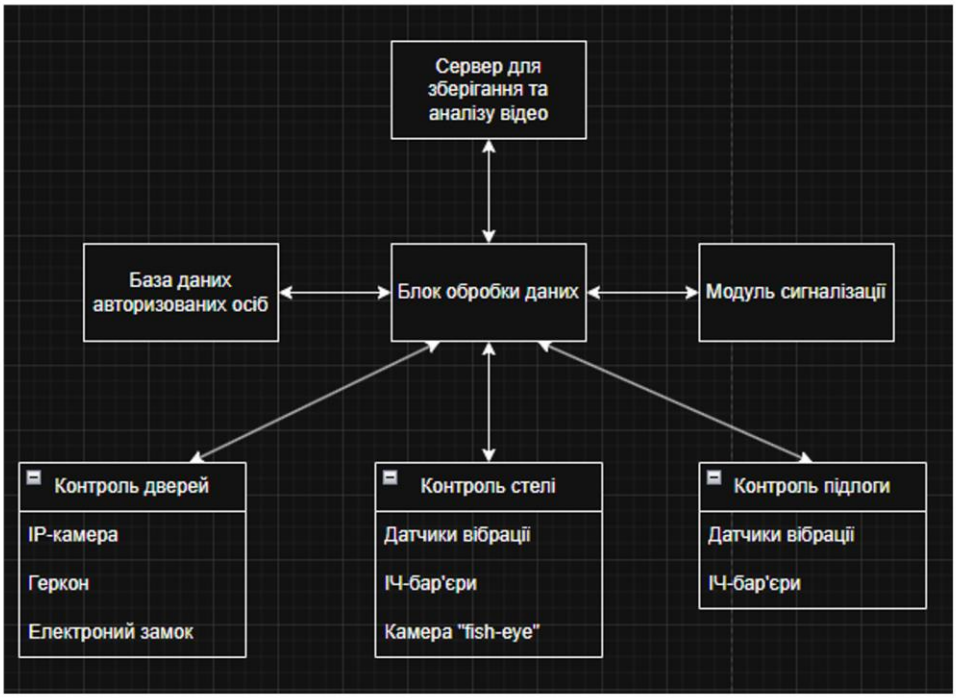
- Механічний замок

- Електроні засоби доступу



- Біометричні засоби доступу

Структурна схема



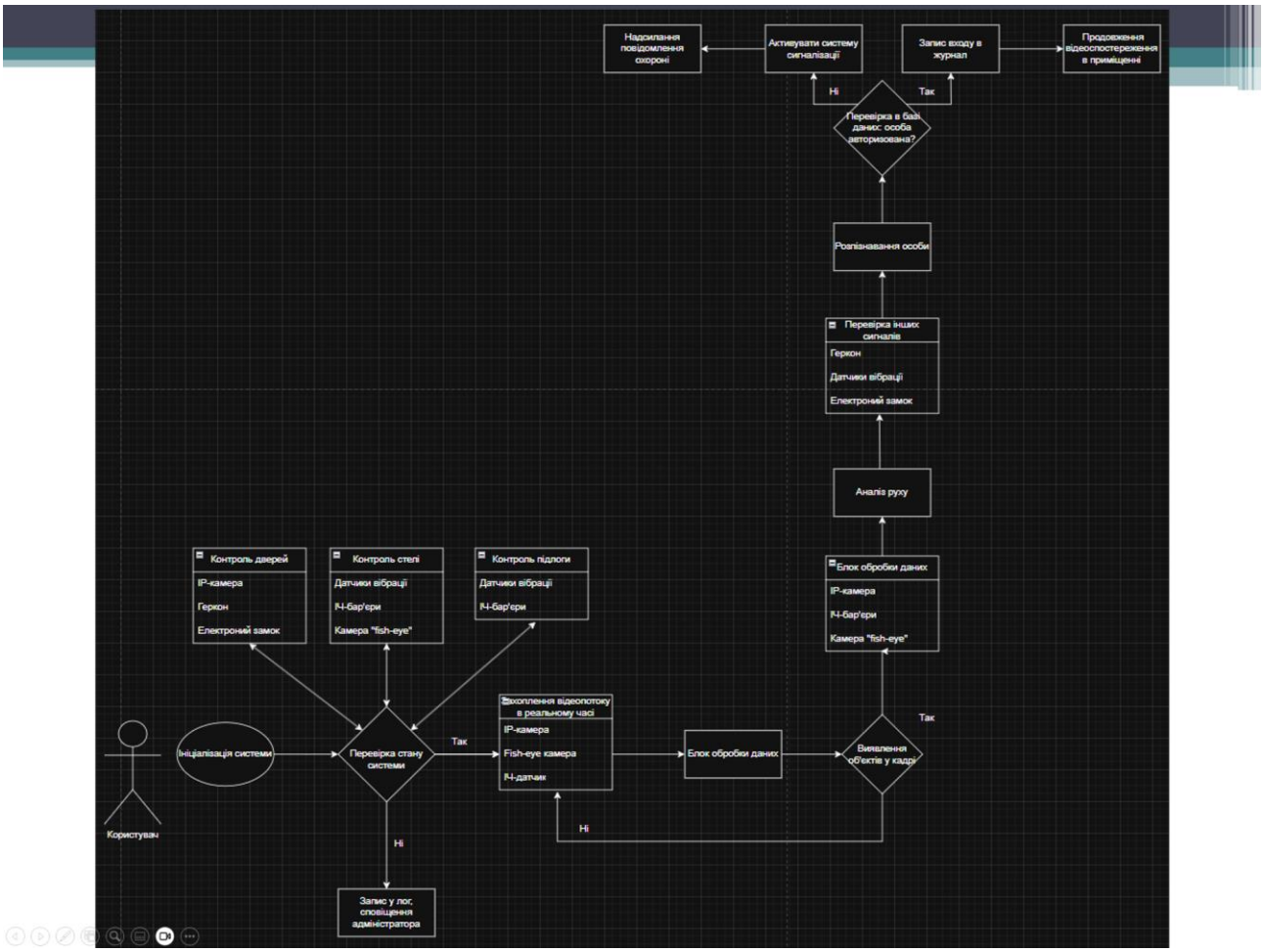
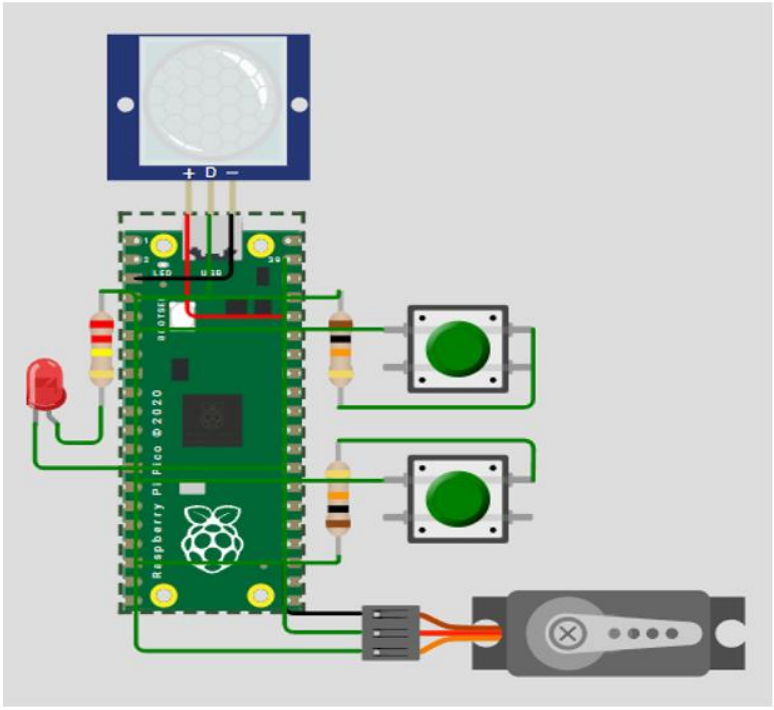

Надсилання повідомлення охороні
Активувати систему сигналізації
Запис входу в журнал
Продовження відеоспостереження в приміщенні
Ні
Так
Перевірка в базі даних: особа авторизована?
Розпізнавання особи
Перевірка інших сигналів
Геркон
Датчики вібрації
Електронний замок
Аналіз руху
Контроль дверей
IP-камера
Геркон
Електронний замок
Контроль стелі
Датчики вібрації
Н-бар'єри
Камера "fish-eye"
Контроль підлоги
Датчики вібрації
Н-бар'єри
Блок обробки даних
IP-камера
Н-бар'єри
Камера "fish-eye"
Користувач
Ініціалізація системи
Перевірка стану системи
Так
Захоплення відеопотоку в реальному часі
IP-камера
Fish-eye камера
Н-датчик
Блок обробки даних
Виявлення об'єктів у кадрі
Так
Ні
Ні
Запис у лог, сповіщення адміністратора

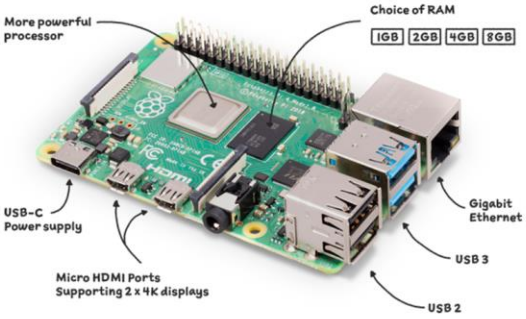
Схема підключення модулю



Вибір апаратної частини системи

Для реалізації підсистеми було обрано компоненти, які забезпечують надійність, сумісність із мікроконтролерами та ефективну роботу в режимі реального часу:

- Raspberry Pi 4 Model B – центральний модуль обробки відео та керування всіма компонентами. Має достатню продуктивність для виконання алгоритмів комп'ютерного зору.
- Камера Raspberry Pi HQ Camera – високоякісна модульна камера з широким кутом огляду. Підходить для постійного відеоспостереження та інтеграції з MediaPipe.

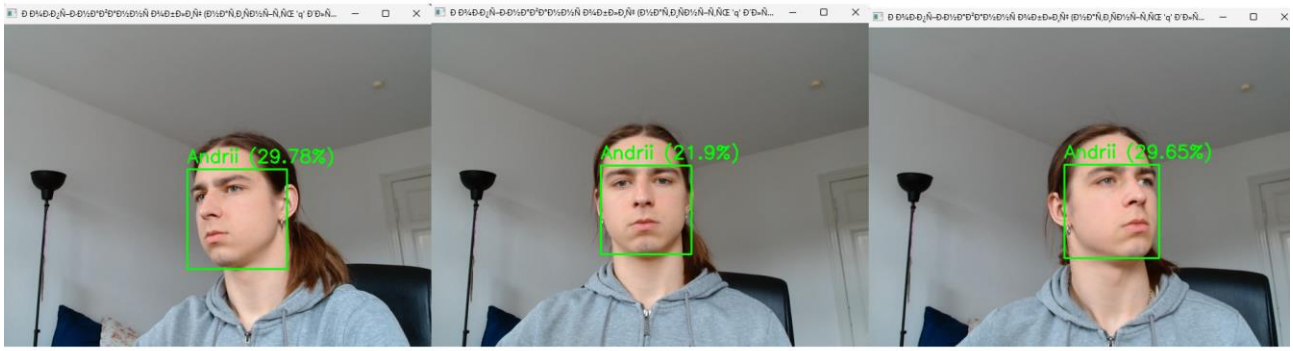


Порівняння алгоритмів розпізнавання облич

У проєкті було протестовано три алгоритми:

- MediaPipe – висока швидкодія, стабільна робота в реальному часі.
- HOG + SVM – добра точність на статичних зображеннях, повільна реакція.
- FaceNet – велика точність, але потребує значних обчислювальних ресурсів та повільно працює.

Результати роботи алгоритму MediaPipe



Перспективи розвитку системи

Система має потенціал для масштабування та вдосконалення. Надалі можливе підключення до хмарних сервісів, інтеграція з існуючими СКУД, впровадження більш точних алгоритмів розпізнавання облич і додавання функцій виявлення аномальної поведінки.

Висновки

У межах кваліфікаційної роботи було успішно реалізовано автоматичну підсистему моніторингу та сповіщення про несанкціонований доступ із використанням технології комп'ютерного зору.

Поставлену мету досягнуто: розроблено апаратно-програмне рішення, яке дає змогу автоматично виявляти та фіксувати спроби несанкціонованого проникнення, проводити ідентифікацію особи в реальному часі, а також оперативно сповіщати відповідальних осіб.

Підсистема підвищує рівень безпеки об'єкта, знижує залежність від людського фактора й забезпечує гнучкість для масштабування та інтеграції з іншими охоронними рішеннями.

Отже, розробка такої підсистеми є актуальною та перспективною в умовах зростання вимог до автоматизації безпеки.

Також, отримані результати роботи можно віднести до Цілі сталого розвитку 9 «Промисловість, інновації та інфраструктура», а саме п. 9.1 «Розвивати якісну надійну, сталу та доступну інфраструктуру, яка базується на використанні інноваційних технологій, у т.ч. екологічно чистих видів транспорту».

№ докумен-та	Позначення	Найменування	Додаткові відомості
		Текстові документи	
1	ГЮИК.503320.013 ПЗ	Пояснювальна записка	А4, 57 с.
		Додаткові матеріали	
2		Демонстраційний матеріал	А4, 12 с.
		у вигляді презентації	

Змін.	Арк	Номер докум.	Підпис	Дата	ГЮИК. 503320.013 ВД			
Розроб.		Норик А. О.			Розроблення автоматичної підсистеми моніторингу та сповіщення про несанкціонований доступ із використанням технології комп'ютерного зору	Літера	Арку	Арку
Перев.		Слюсар А. П.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.конт		Демська Н. П.						
Затв.		Невлюдов І.Ш.						