ДОДАТОК А

Опис технічних характеристик та інструкція користування системою

Призначення програмного продукту: ця система створена для виявлення, стеження та визначення швидкості транспортних засобів у реальному часі, базуючись на відеопотоці. Програма забезпечує автоматичне фіксування об’єктів, швидкість яких перевищує встановлену межу (наприклад, 50 км/год), та їх візуальне виокремлення.

Технічні характеристики :

– операційна система: Windows 11;

– мова програмування: Python 3.13.1.

Обов’язкові бібліотеки:

– ultralytics;

– opencv-python;

– numpy.

Встановлення бібліотек :

pip install ultralitics opencv-python numpy

Підготовка до запуску :

– завантаження моделі YOLOv8n (при першому запуску Ultralytics зробить це автоматично);

– перевірка, що файл відео або камера доступні системі;

– відкриття файлу main.py.

Параметри налаштування (у коді) :

– SOURCE = 'video.mp4' – шлях до відео або 0 для вебкамери;

– PIXEL_TO_KMH = 0.15 – коефіцієнт перетворення піксельної швидкості в км/год;

– SPEED_LIMIT = 50 – гранична швидкість, при перевищенні якої об’єкт буде підсвічено червоним.

Запуск системи :

– відкриття терміналу у каталозі з проєктом;

– запуск програми: python main.py;

– на екрані з’явиться вікно з відео та рамками навколо транспортних засобів;

– над кожним об'єктом виводиться швидкість у форматі: «car 52.7 km/h».

Правила користування :

– для завершення роботи системи потрібно натиснути клавішу «ESC»;

– відео має бути без будь-яких ривків (плавне).

Таблиця А1 – Позначення на екрані та їх пояснення

Повідомлення на екрані	Пояснення
car 34.5 km/h	Автомобіль їде із допустимою швидкістю – підсвічено зеленим
bus 62.1 km/h	Автобус перевищив ліміт – підсвічено червоним
No frame	Відеопотік недоступний або завершився

Таблиця А2 – Типові проблеми та їх вирішення

Проблема	Рішення
Чорне вікно або нічого не відображається	Перевірити джерело відео (SOURCE) і права доступу
Повільна обробка	Зменшити роздільну здатність кадру (640×360 рекомендовано)
Всі об'єкти помічено зеленим	Можливо, значення PIXEL_TO_KMH занадто мале – змінити

Рекомендації :

– для зручного перегляду запускати на екрані, роздільна здатність якого не нижче 1280×720;

– якщо використовується веб-камера, вона повинна бути розташована на підвищенні, під кутом, щоб була видима вся смуга руху.

Можливості для подальшого розширення :

– збереження логів порушень у файл (CSV, JSON);

– розпізнавання номерних знаків (інтеграція з ANPR);

– вебінтерфейс на Flask або Streamlit;

– Telegram-бот для сповіщень про порушення.

ДОДАТОК Б

Реалізований програмний код

```
import cv2
import time
from ultralytics import YOLO
import numpy as np
# Завантаження моделі YOLOv8
model = YOLO(«yolov8n.pt»)
# Класи, які нас цікавлять
VEHICLE_CLASSES = {'car','truck'}
# Масштаб для піксель/сек → км/год (підбирається під
відео)
PIXEL_TO_KMH = 0.15
SPEED_LIMIT = 50
SOURCE = 'potok.mp4'  # або 0 для вебкамери
cap = cv2.VideoCapture(SOURCE)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
# Відстеження центрів авто для розрахунку швидкості
object_speeds = {}
# Область ROI: тільки нижня половина кадру
ROI_TOP = 100
ROI_BOTTOM = 360
ROI_LEFT = 0
ROI_RIGHT = 640
prev_time = time.time()
frame_count = 0
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame_count += 1
    if frame_count % 2 != 0:
        continue  # Пропускаємо кожен другий кадр
```

```
        frame = cv2.resize(frame, (640, 360))
        #  ПОПЕРЕДНЯ ОБРОБКА ЗОБРАЖЕННЯ
        pre_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)  #
HSV
        pre_frame = cv2.GaussianBlur(pre_frame, (5, 5), 0)  #
Розмиття
        h, s, v = cv2.split(pre_frame)
        clahe = cv2.createCLAHE(clipLimit=2.0,
tileGridSize=(8, 8))
        v = clahe.apply(v)
        pre_frame = cv2.merge((h, s, v))
        frame = cv2.cvtColor(pre_frame, cv2.COLOR_HSV2BGR)  #
Назад у BGR
        #  Детекція + трекінг
        for results in model.track(frame, persist=True,
verbose=False):         break
        current_time = time.time()
        dt = current_time - prev_time
        prev_time = current_time
        if results.boxes is not None:
            for box in results.boxes:
                cls_id = int(box.cls[0])
                cls_name = model.names[cls_id]
                if cls_name not in VEHICLE_CLASSES:
                    continue
                x1, y1, x2, y2 = map(int, box.xyxy[0])
                cx, cy = int((x1 + x2) / 2), int((y1 + y2) /
2)
                # ROI обмеження
                if not (ROI_LEFT <= cx <= ROI_RIGHT and
ROI_TOP <= cy <= ROI_BOTTOM):
                    continue
                track_id = int(box.id[0]) if box.id is not
None else None
                speed_kmh = 0
                color = (0, 255, 0)
```

```
                if track_id is not None:
                    if track_id in object_speeds:
                        px, py, pt = object_speeds[track_id]
                        dist = ((cx - px) ** 2 + (cy - py) **
2) ** 0.5
                        dt = current_time - pt
                        if dt > 0:
                            speed_kmh = dist / dt *
PIXEL_TO_KMH
                        object_speeds[track_id] = (cx, cy,
current_time)
                    else:
                        object_speeds[track_id] = (cx, cy,
current_time)
                    if speed_kmh > SPEED_LIMIT:
                        color = (0, 0, 255)
                label = f»{cls_name} {speed_kmh:.1f} km/h»
                cv2.rectangle(frame, (x1, y1), (x2, y2),
color, 2)
                cv2.putText(frame, label, (x1, y1 - 10),
                            cv2.FONT_HERSHEY_SIMPLEX, 0.6,
color, 2)
        # Відображення зони ROI
        cv2.rectangle(frame, (ROI_LEFT, ROI_TOP), (ROI_RIGHT,
ROI_BOTTOM), (255, 255, 0), 2)
        cv2.putText(frame, «ROI zone», (ROI_LEFT + 10, ROI_TOP
+ 25),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.6, (255, 255,
0), 2)
        # Показ результату
        cv2.imshow(«Vehicle Speed Detection», frame)
        # Вихід за ESC
        if cv2.waitKey(1) & 0xFF == 27:
            break
    cap.release()
    cv2.destroyAllWindows()
```

Результат наведено на рисунках Б.1–Б.5.

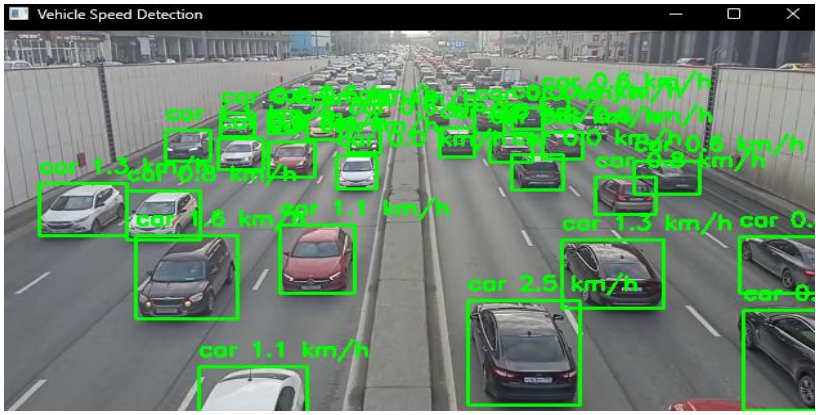

Рисунок Б.1 – Результат

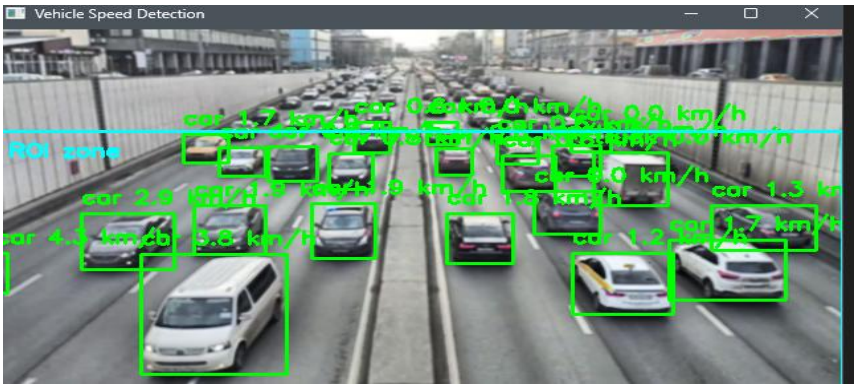

Рисунок Б.2 – Результат

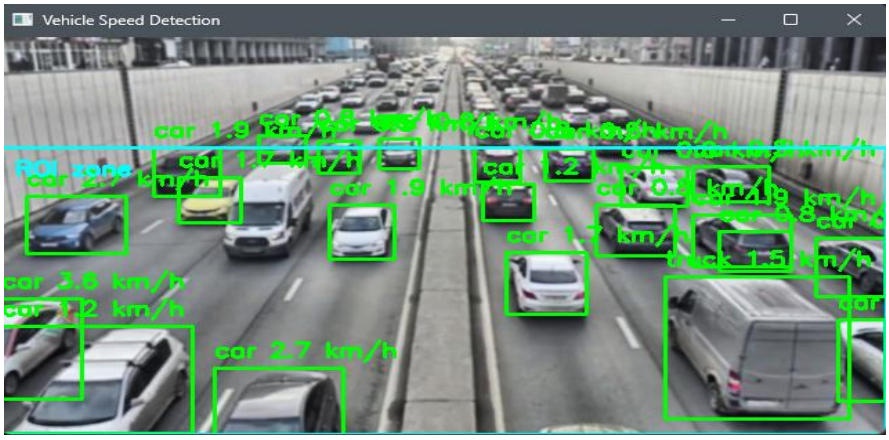

Рисунок Б.3 – Результат

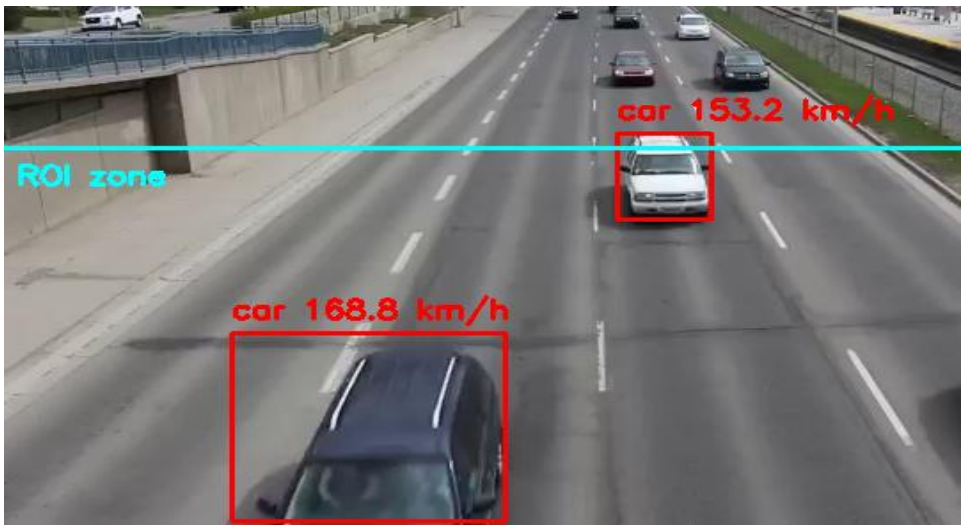

Рисунок Б.4 – Результат

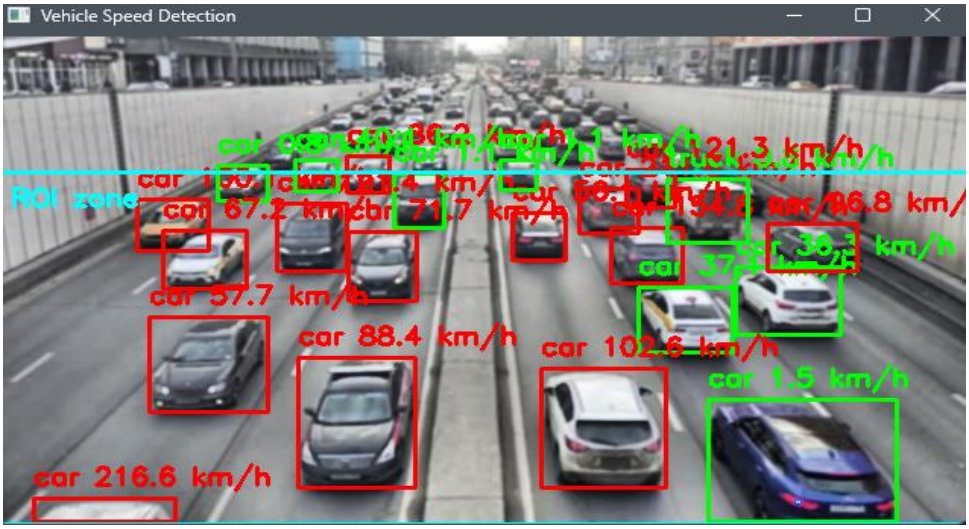

Рисунок Б.5 – Результат

ДОДАТОК В

Відомість кваліфікаційної роботи

Позначення	Найменування	Дод. відомості
	Текстові документи	
1.	Пояснювальна записка	52 с.
	Інші документи	
2.	Презентаційні матеріали	__ плакатів

.	.	Прізвище та ініціали.	Підп.	Дата			
Розробив		Погребняк Я.В.			Алгоритмічні підходи до ідентифікації транспортних засобів за допомогою комп’ютерного зору	Шифр групи	Код напр./спец.
Перевірив		Єрохін М.А.				СШІм-23-2	122
Н.контр.		Малєєва І.А.				ХНУРЕ кафедра ШІ	
Затв.		Золотухін О.В.					