Додаток А

Апробація результатів кваліфікаційної роботи

Бас Віталій Олегович, здобувач вищої освіти факультету автоматики і комп'ютеризованих технологій
«Харківський національний університет радіоелектроніки», Україна

Наук керівник: Цимбал Олександр Михайлович, доктор технічних наук, професор кафедри КІТАР
«Харківський національний університет радіоелектроніки», Україна

РОЗПІЗНАВАННЯ ОБ'ЄКТІВ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЙ КОМП'ЮТЕРНОГО ЗОРУ

Актуальність теми пояснюється зростаючою потребою в автоматизації обробки візуальних даних, що наближається до людських можливостей. Ця технологія є ключовою для галузей, де оперативність, точність і масштабність аналізу зображень мають вирішальне значення. Зростання використання комп'ютерного зору підкреслює його перспективність.

Згідно дослідження І. М. Гангало, Д. О. Лісовий, В. В Жебка, Комп'ютерний зір, характеризується як галузь комп'ютерних наук, що зосереджена на розробці систем, здатних обробляти, аналізувати та інтерпретувати зображення чи відео подібно до людського сприйняття. Комп'ютерний зір включає класифікацію, ідентифікацію та стеження за об'єктами, використовуючи алгоритми, що розпізнають візерунки та форми на рівні пікселів [2].

Технології комп'ютерного зору застосовуються для вирішення спеціалізованих задач, таких як виявлення, ідентифікація та оцінка положення. Виявлення передбачає перевірку відеоданих на наявність певних умов, наприклад, аномалій у медичних зображеннях, що часто слугує першим етапом для подальшого детального аналізу. Ідентифікація фокусується на розпізнаванні унікальних екземплярів, таких як конкретне обличчя чи автомобіль, оцінка положення визначає орієнтацію об'єкта відносно камери, що має практичне застосування в робототехніці, наприклад, для маніпуляцій на конвеєрних лініях [4].

Розпізнавання об'єктів базується на методах контурного аналізу, який визначає форму об'єкта через обробку контурів, алгоритму Віоли-Джонса, що використовує ознаки Хаара для швидкого виявлення рис зображення, статистичних методів для аналізу даних, та нейронних мереж, які самостійно виділяють інформаційні ознаки, демонструючи найвищу перспективність завдяки адаптивності [1].

Розпізнавання об'єктів відбувається за допомогою штучного інтелекту та полягає в кодуванні візуальних даних у вектори ознак, їх порівнянні з еталонними шаблонами в пам'яті системи та класифікації об'єктів за категоріями. Алгоритми машинного та глибокого навчання обробляють великі набори зображень, що забезпечує високу точність розпізнавання навіть у складних сценаріях. Системи можуть не лише ідентифікувати об'єкти, а й передавати дані для подальшої обробки, або керувати об'єктами через зворотний зв'язок [3].

Застосування систем комп'ютерного зору охоплює широкий спектр галузей. У медицині вони забезпечують автоматизований аналіз медичних зображень для

Рисунок А.1 – Тезиси, перша сторінка

діагностики та моніторингу. У виробництві сприяють контролю якості та розвитку робототехніки. У транспорті технології розпізнавання об'єктів є основою для автономного водіння, дозволяючи виявляти дорожні знаки чи пішоходів. Безпека, роздрібна торгівля, сільське господарство, геоінформаційні системи та охорона навколишнього середовища також активно використовують ці технології для вирішення спеціалізованих завдань, таких як моніторинг урожаю чи аналіз кліматичних змін [5].

ВИСНОВОК

Розпізнавання об'єктів є одним із напрямів розвитку комп'ютерного зору, що поєднує в собі сучасні досягнення інформатики, штучного інтелекту та обробки зображень. Завдяки використанню алгоритмів машинного та глибокого навчання, ці системи здатні ефективно аналізувати, класифікувати та ідентифікувати об'єкти у складних та динамічних середовищах. Високий рівень точності, гнучкість і масштабність роблять технології комп'ютерного зору незамінними в різних сферах, від медицини та транспорту до, виробництва й екологічного моніторингу.

Список використаних джерел:

1. Волових Б. П., Перевозніков С. І., Озеранський В. С. Актуальні методи розпізнавання об'єктів у системах комп'ютерного зору. Вінницький національний технічний університет. 3с.
2. Гангало І. М., Лісовий Д. О., Жебка В. В. Розпізнавання об'єктів за допомогою технологій комп'ютерного зору. Телекомунікаційні та інформаційні технології. 2022. № 4 (77). С. 46-52.
3. Гордійченко О. В. Використання штучного інтелекту для розпізнавання образів та систем комп'ютерного зору. URL: https://conf.ztu.edu.ua/wp-content/uploads/2024/05/202.pdf (дата звернення: 07.05.2025)
4. Основи комп'ютерного зору. Лекція №8. URL: https://learn.ztu.edu.ua/pluginfile.php/294778/mod_resource/content/1/%D0%9E%D0%A8%D0%86_%D0%9B-8_%D0%9A%D0%97%D1%96%D1%80.pdf (дата звернення: 07.05.2025)
5. Придатько Д. Р. Огляд методів розпізнавання об'єктів за допомогою систем технічного зору. Автоматизація та розвиток електронних пристроїв. 2023. Ч. 2. С. 8-11.

Рисунок А.2 – Тезиси, друга сторінка

Додаток Б

Лістинг програмного коду

train2.py:

```
import os
os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"

from ultralytics import YOLO

if __name__ == '__main__':
    # Шлях до файлу data.yaml
    data_yaml_path = 'J:/yolo/dataset/data.yaml'

    model = YOLO('yolov8m.pt')

    # Параметри навчання
    epochs = 200
    imgsz = 640
    batch = 16
    workers = 2
    device = 0
    lr0 = 0.001
    augment = True      # аугментації
    hsv_h = 0.01        # налаштування аугментацій
    hsv_s = 0.7
    hsv_v = 0.4
    translate = 0.1
    scale = 0.5
    shear = 0.1
    flipud = 0.0
    fliplr = 0.5
    mosaic = 1.0
    mixup = 0.0
    copy_paste = 0.0
    patience = 15

    print(f"Починаємо навчання моделі...")
    print(f"Файл конфігурації: {data_yaml_path}")
    print(f"Кількість епох: {epochs}")
    print(f"Розмір зображення: {imgsz}")
    print(f"Розмір пакета: {batch}")
    print(f"Кількість робітників: {workers}")
    print(f"Пристрій: {device}")

    # Запуск навчання
      results = model.train(data=data_yaml_path, epochs=epochs, imgsz=imgsz, batch=batch,
workers=workers, device=device,

 lr0=lr0,
augment=augment,
hsv_h=hsv_h,
hsv_s=hsv_s,
```

```
hsv_v=hsv_v,
translate=translate,
scale=scale,
shear=shear,
flipud=flipud,
fliplr=fliplr,
mosaic=mosaic,
mixup=mixup,
copy_paste=copy_paste)

    print("\nНавчання завершено")

    # Оцінка моделі на валідаційному наборі
    print("\nОцінка моделі на валідаційному наборі...")
    metrics = model.val(data=data_yaml_path, device=device)
    print(f"Результати валідації: {metrics}")

    # Збереження навченої моделі
    trained_model_path = 'J:/yolo/runs1/train/exp/weights/best.pt'
    print(f"\nНавчена модель збережена за шляхом: {trained_model_path}")
```

podil.py:

```
import os
import shutil
import random

# Шляхи до датасету
dataset_path = r"J:\yolo4\dataset"
labels_path = os.path.join(dataset_path, "labels")  # TXT-файли з лейблами
images_path = os.path.join(dataset_path, "images")  # Зображення

# Новий шлях для поділеного датасету
output_dir = os.path.join(dataset_path, "split_dataset")
splits = ["train", "valid", "test"]
split_ratios = [0.7, 0.2, 0.1]  # 70% train, 20% valid, 10% test

try:
    # Створення необхідних папок
    os.makedirs(output_dir, exist_ok=True)
```

```
    for split in splits:
        os.makedirs(os.path.join(output_dir, split, "images"), exist_ok=True)
        os.makedirs(os.path.join(output_dir, split, "labels"), exist_ok=True)

    # Перевірка наявності .txt лейблів
    all_label_files = [f for f in os.listdir(labels_path) if f.endswith(".txt")]
    all_image_files = [f.replace(".txt", ".jpg") for f in all_label_files]

    # Перевірка чи всі зображення відповідають лейблам
    missing_images = [f for f in all_image_files if not os.path.exists(os.path.join(images_path, f))]
    if missing_images:
        print(f"[!] Пропущено зображення: {', '.join(missing_images)}")

    # Якщо зображення не знайдено, видаляємо їх з списку
        all_label_files = [f for f in all_label_files  if os.path.exists(os.path.join(images_path,
f.replace(".txt", ".jpg")))]

    # Поділ файлів на train, valid, test
    random.shuffle(all_label_files)

    # Визначаємо межі для поділу
    total_files = len(all_label_files)
    train_size = int(total_files * split_ratios[0])
    valid_size = int(total_files * split_ratios[1])

    train_files = all_label_files[:train_size]
    valid_files = all_label_files[train_size:train_size+valid_size]
    test_files = all_label_files[train_size+valid_size:]

    def move_files(file_list, split):
        for label_file in file_list:
            image_file = label_file.replace(".txt", ".jpg")
            label_src = os.path.join(labels_path, label_file)
            image_src = os.path.join(images_path, image_file)
```

```
            label_dst = os.path.join(output_dir, split, "labels", label_file)
            image_dst = os.path.join(output_dir, split, "images", image_file)

            if os.path.exists(label_src) and os.path.exists(image_src):
                shutil.copy2(label_src, label_dst)
                shutil.copy2(image_src, image_dst)
            else:
                print(f"[!] Пропущено: {label_file} або {image_file} не знайдено")

    # Переміщуємо файли в кожну папку для train, valid, test
    move_files(train_files, "train")
    move_files(valid_files, "valid")
    move_files(test_files, "test")

    print("Звичайний поділ завершено успішно")

except Exception as e:
    print(f"Виникла помилка: {e}")
```

detect_gui.py:

```
import cv2
import os
import csv
import time
from datetime import datetime
from ultralytics import YOLO
import tkinter as tk
from PIL import Image, ImageTk
# Налаштування
MODEL_PATH = 'J:/yolo4/runs/detect/train4/weights/best.pt'
CLASSES_VIOLATIONS = {'No_BreathingApparatus', 'No_Glove', 'No_Goggles',
'No_Helmet'}
SAVE_INTERVAL = 5.0  # секунд між збереженням кадрів
```

```
OUTPUT_FOLDER = 'violations_captures'
CSV_LOG = 'violations_log.csv'

# Ініціалізація
os.makedirs(OUTPUT_FOLDER, exist_ok=True)
model = YOLO(MODEL_PATH)
cap = cv2.VideoCapture(0)

# Створення CSV-файлу з заголовками
if not os.path.exists(CSV_LOG):
    with open(CSV_LOG, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(['Time', 'Violations'])

# Tkinter
root = tk.Tk()
root.title("Монітор безпеки")
video_label = tk.Label(root)
video_label.pack()

status_label = tk.Label(root, text="Стан: Перевірка...", font=("Arial", 12), fg="blue")
status_label.pack(pady=5)

# Логіка
last_save_time = 0
def process_frame():
    global last_save_time
    ret, frame = cap.read()
    if not ret:
        root.after(10, process_frame)
        return

    results = model(frame, verbose=False)[0]
    names = model.names
    boxes = results.boxes.xyxy
```

```
classes = results.boxes.cls

person_box = None
violations_detected = set()

# Пошук першої людини
for box, cls_id in zip(boxes, classes):
    label = names[int(cls_id)]
    if label == "Person":
        person_box = box
        break

# Якщо є людина то перевіряємо перетин порушень з її рамкою
if person_box is not None:
    px1, py1, px2, py2 = person_box

    for box, cls_id in zip(boxes, classes):
        label = names[int(cls_id)]
        if label in CLASSES_VIOLATIONS:
            x1, y1, x2, y2 = box
            if not (x2 < px1 or x1 > px2 or y2 < py1 or y1 > py2):
                violations_detected.add(label)

violations_detected = sorted(violations_detected)
person_present = person_box is not None

# Візуалізація результатів
annotated_frame = results.plot()

# Відображення у вікні
frame_rgb = cv2.cvtColor(annotated_frame, cv2.COLOR_BGR2RGB)
img = Image.fromarray(frame_rgb)
imgtk = ImageTk.PhotoImage(image=img)
video_label.imgtk = imgtk
video_label.configure(image=imgtk)
```

```
    # Логіка фіксації порушення
names = model.names
current_labels = [names[int(cls)] for cls in results.boxes.cls]
person_present = 'Person' in current_labels
violations_detected = sorted(set(current_labels) & CLASSES_VIOLATIONS)

if person_present and violations_detected:
    status_label.config(text=f"Порушення: {', '.join(violations_detected)}", fg="red")

    current_time = time.time()
    if current_time - last_save_time >= SAVE_INTERVAL:
        timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
        cv2.imwrite(f"{OUTPUT_FOLDER}/violation_{timestamp}.jpg",  annotated_frame)

        with open(CSV_LOG, 'a', newline='') as file:
            csv.writer(file).writerow([timestamp, ', '.join(violations_detected)])

        last_save_time = current_time
else:
    status_label.config(text="Все в нормі", fg="green")

    root.after(10, process_frame)

# Запуск
process_frame()
root.protocol("WM_DELETE_WINDOW", lambda: (cap.release(), root.destroy()))
root.mainloop()
```

ДОДАТОК В

Демонстраційний матеріал

№ докумен-та	Позначення	Найменування	Додаткові відомості
		Текстові документи	
1	ГЮИК. 651400.002	Пояснювальна записка	А4, 88 с.
		Додаткові матеріали	
2		Апробація результатів	А4, 2 с.
		кваліфікаційної роботи	
3		Лістинг програмного коду	А4, 7 с.
4		Демонстраційний матеріал	А4, 15 с.

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮИК. 651400.002			
Розроб.		Бас В. О.			Розроблення системи автоматизації розпізнавання об’єктів у виробничому приміщенні	Літера	Аркуш	Аркушів
Перевір.		Цимбал О. М.				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Стародубцев М. Г.						
Затв.		Невлюдов І.Ш.						