**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №4**

**з навчальної дисципліни «Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПОКРАЩЕННЯ ЯКОСТІ ЦИФРОВИХ**

**ЗОБРАЖЕНЬ ДЛЯ ЗАДАЧ COMPUTER VISION**

**Виконав:**

Студент 3 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Трикош І. В.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

## **І. Мета:**

Дослідити принципи та особливості практичного застосування технологій покращення якості цифрових зображень для задач Computer Vision з використанням спеціалізованих програмних бібліотек.

## **ІІ. Завдання:**

Здійснити R&D дослідження та реалізувати програмний скрипт із конкретикою методів і технологічних етапів Computer Vision: вибір цифрового зображення та об’єкта ідентифікації; завантаження цифрового зображення; покращення якості цифрового зображення; векторизація об’єкта ідентифікації – встановлення геометричної ознаки; ідентифікація об’єкта за геометричною ознакою.

Вибір цифрового зображення та об’єкту ідентифікації встановлено варіантами таблиці додатку.

Для покращення якості цифрового зображення використовувати:

корекцію кольору;

корекцію гістограми яскравості (для всього зображення (глобальна) / для сегменту

зображення (локальна));

методи / алгоритми фільтрації зображень.

Вибір переліку методів покращення якості має бути обґрунтованим та забезпечувати побудову контору об’єкту ідетифікації.

Для векторизації зображення (визначення контуру) використовувати методи базових бібліотек python для обробки цифрових зображень;

Ідентифікацію здійснювати за технологією порівняння геометричних ознак (контуру) образу та об’єкту ідентифікації.

**Завдання І рівня**

Здійснити виконання завдання лабораторної роботи для статичного цифрового зображення за варіантами таблиці додатку. Цифрове зображення - автомагістраль, об’єкт ідентифікації - номери державної реєстрації на автомобілях.

## **ІІІ. Результати виконання лабораторної роботи.**

* 1. **Синтезована математична модель**

Відповідно до умов задачі синтезовано математичні моделі операцій над структурами вхідних графічних об’єктів.

Модель реалізує виділення контурів графічних об’єктів (номери державної реєстрації) на автомобілях. Для цього вона покращить зображення з допомогою blackhat, що дозволить виявити темні області (текст на номерному знаку) на світлому фоні (тобто сам знак). Далі відбувається Гаусівське розмиття зображення, згладжування для закриття зображення, і потім застосовується алгоритм OTSU для визначення порогу бінаризації, щоб розділити зображення на фон та об’єкти. Далі застосовується ерозія та розширення для зменшення шуму та покращення більших об’єктів. Після цього відбувається пошук контурів на покращеному зображенні, їх апроксимація та фільтрація. Якщо контур має 4 сторони, і ширина втричі більша за висоту та висота не є дуже малою, то ідентифікуємо цей контур як номер державної реєстрації та зобразимо цей контур на початковому зображенні.

Обробка зображення відбувається з допомогою засобів бібліотеки OpenCV з використанням Python.

* 1. **Результати архітектурного проектування та їх опис**

На рис. 1 зображена блок-схема до моделі до програмного скрипту:

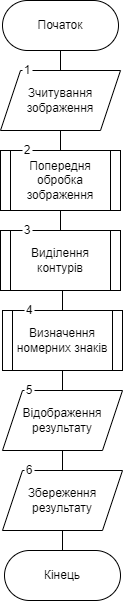


Рисунок 1 – Блок схема алгоритму

Робота алгоритму розпочинається із зчитування зображення з допомогою бібліотеки OpenCV. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.1.

У блоці 2 виконується попередня обробка зображень. Тут зображення перетворюється у відтінки сірого, застосовується операція blackhat, Гаусівське розмиття, згладжування для закриття, визначення порогу бінаризації для розділення зображення на фон та об’єкти, та застосовується ерозія та розширення для зменшення шуму та покращення більших об’єктів.

У блоці 3 виконується виділення зовнішніх контурів об’єктів.

У блоці 4 виконується визначення номерних знаків державної реєстрації на автомобілях. Це відбувається за допомогою фільтрації по 4 вершинах, ширині, яка має бути втричі більша за висоту, і висоті, яка не має бути дуже малою.

У блоці 5 відображається результат виділення контурів.

У блоці 6 виконується збереження результату. На цьому виконання алгоритму завершено.

* 1. **Опис структури проекту програми**

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

Проєкт базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

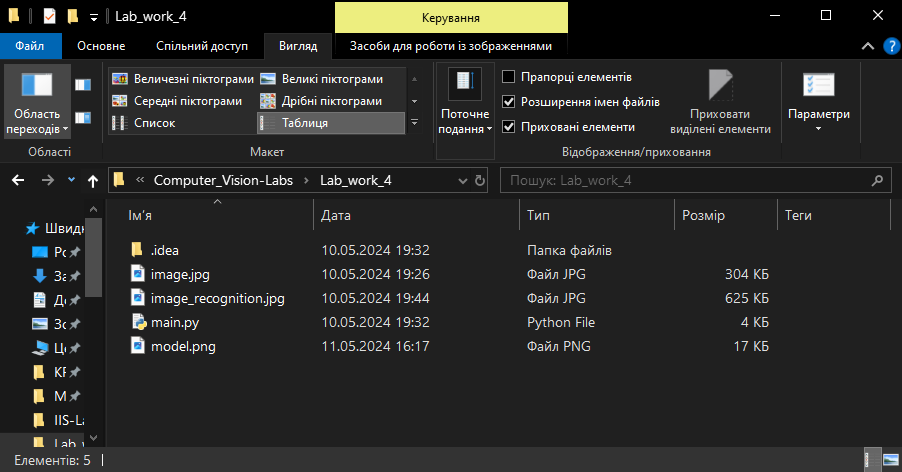


Рисунок 2 – Структура проєкту

Lab\_work\_4 – головний каталог проєкту

main.py – файл програмного коду

image.jpg – зображення для тестування програми

image\_recognition.jpg – результат виділення контурів об’єктів

* 1. **Результати роботи програми відповідно до завдання (допускається у формі скриншотів)**

Результатом роботи програми є сукупність послідовності графічних вікон, що реалізують умови завдання лабораторної роботи.

1. Початкове зображення:

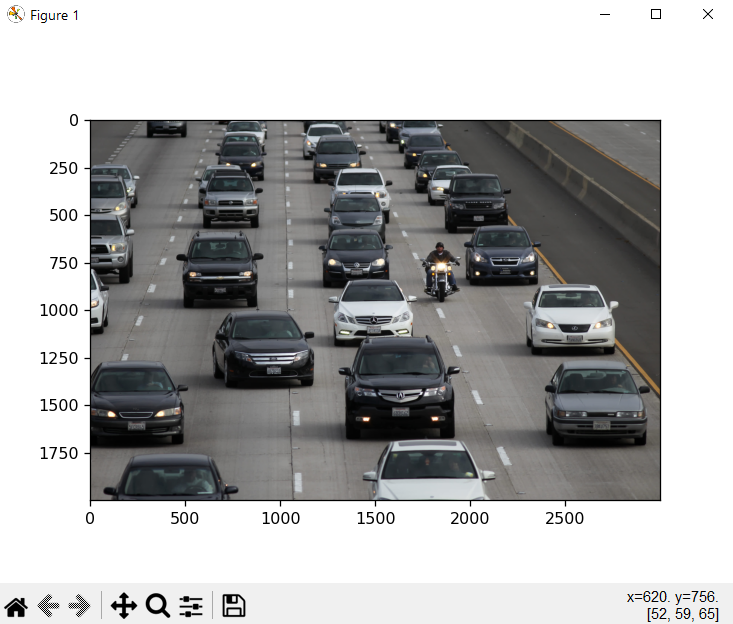


Рисунок 3 – Початкове зображення

1. Оброблене зображення:

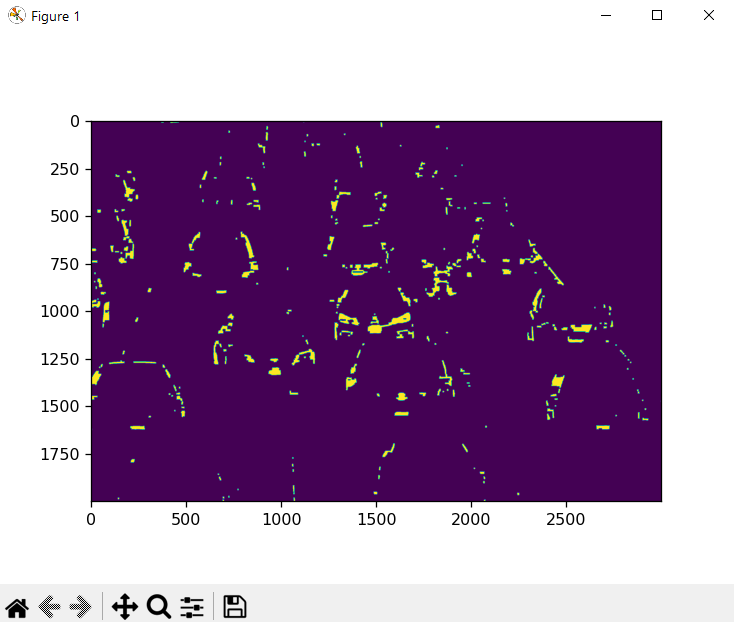


Рисунок 4 – Оброблене зображення

1. Виділені контури об’єктів:



Рисунок 5 – Виділені контури номерних знаків на автомобілях

Бачимо, що не всі номерні знаки виділено, проте задача ідентифікації таких об’єктів є складною і для цього краще застосовувати нейронні мережі. Таким чином, представлені результати відповідають завданню лабораторної роботи.

* 1. **Програмний код, що забезпечує отримання результату (допускається у формі скриншотів)**

Програмний код послідовно реалізує алгоритм на рис.1 та спрямований на отримання результатів, поданих на рис.3-5.

Для спрощення програмного коду і раціоналізації обчислень застосовано функціональні механізми створення підпрограм.

При цьому використано можливості Python бібліотек: matplotlib, OpenCV.

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

main.py

*"""  
Ідентифікація номерів державної реєстрації на автомобілях  
"""  
  
import* cv2  
*import* matplotlib.pyplot *as* plt  
  
  
*def* show\_image(image):  
 *""" Показ зображення """* plt.imshow(image)  
 plt.show()  
  
  
*def* image\_read(file\_image):  
 *""" Зчитування зображення з файлу """* image = cv2.imread(file\_image)  
 image\_in\_rgb = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)  
 *return* image\_in\_rgb  
  
  
*def* image\_processing(image):  
 *""" Покращення зображення """  
  
 # Перетворюємо зображення у відтінки сірого, виконаємо операцію blackhat, яка дозволить  
 # виявити темні області (текст на номерному знаку) на світлому фоні (тобто сам знак)* gray\_image = cv2.cvtColor(image, cv2.COLOR\_RGB2GRAY)  
 rectangular\_kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (13, 5))  
 blackhat\_transformed = cv2.morphologyEx(gray\_image, cv2.MORPH\_BLACKHAT, rectangular\_kernel)  
  
 *# Розмиваємо зображення, застосовуємо згладжування для закриття та алгоритм OTSU для  
 # визначення порогу бінаризації, щоб розділити зображення на фон та об'єкти* blurred\_image = cv2.GaussianBlur(blackhat\_transformed, (5, 5), 0)  
 blurred\_image = cv2.morphologyEx(blurred\_image, cv2.MORPH\_CLOSE, rectangular\_kernel)  
 binary\_threshold\_image = cv2.threshold(blurred\_image, 0, 255, cv2.THRESH\_BINARY | cv2.THRESH\_OTSU)[1]  
  
 *# Ерозія та розширення для зменшення шуму* processed\_image = cv2.erode(binary\_threshold\_image, *None*, iterations=3)  
 processed\_image = cv2.dilate(processed\_image, *None*, iterations=3)  
 *return* processed\_image.copy()  
  
  
*def* find\_contours(processed\_image):  
 *""" Знаходження контурів на зображення """* contours, \_ = cv2.findContours(processed\_image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)  
 *return* contours  
  
  
*def* draw\_contours(image, contours):  
 *"""  
 Малювання контурів на початковому зображенні.  
 До уваги беруть ті контури, які мають 4 сторони, де ширина втричі більша за висоту і висота не є малою  
 """  
 for* contour *in* contours:  
 peri = cv2.arcLength(contour, *True*)  
 approx = cv2.approxPolyDP(contour, 0.03 \* peri, *True*)  
 \_, \_, w, h = cv2.boundingRect(approx)  
 *if* len(approx) == 4 *and* w > h \* 3 *and* h > 10:  
 cv2.drawContours(image, [approx], -1, (0, 255, 0), 4)  
  
 *return* image  
  
  
*def* save\_result(file\_name, image\_in\_rgb):  
 *""" Збереження результату у файл """* image\_in\_bgr = cv2.cvtColor(image\_in\_rgb, cv2.COLOR\_RGB2BGR)  
 cv2.imwrite(file\_name, image\_in\_bgr)  
  
  
image\_entrance = image\_read("image.jpg")  
show\_image(image\_entrance)  
  
processed\_img = image\_processing(image\_entrance)  
show\_image(processed\_img)  
  
found\_contours = find\_contours(processed\_img)  
  
result\_image = draw\_contours(image\_entrance, found\_contours)  
save\_result("image\_recognition.jpg", result\_image)  
show\_image(result\_image)

## **IV. Висновки.**

У ході виконання лабораторної роботи було проведено дослідження принципів та особливостей практичного застосування технологій покращення якості цифрових зображень для задач Computer Vision з використанням спеціалізованих програмних бібліотек. Було реалізовано алгоритм, що покращує вхідне зображення та розпізнає номерні знаки на автомобілях на цифровому растровому зображенні з використанням засобів мови програмування Python.

Виконав: студент Трикош І. В.