**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

Головня О.Р.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

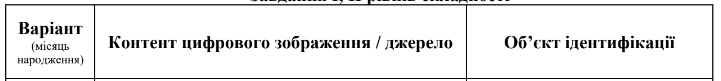
**І. Мета роботи:**

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

**ІІ. Завдання**

Завдання І рівня – максимально 8 балів.

Здійснити виконання завдання лабораторної роботи для статичного цифрового зображення за варіантами таблиці додатку.





**ІІІ. Результати виконання лабораторної роботи.**

**3.1. Синтезована математична модель перетворень графічних об’єктів відповідно до індивідуального завдання.**

Відповідно до завдання, синтезовано математичну модель операцій над зображенням нафтесховища, отриманого за допомогою ДЗЗ.

Конвертація кольорів: Спочатку, зображення конвертується з формату BGR в HSV. Це дозволяє зручно працювати з кольорами, оскільки HSV (відтінок, насиченість, значення) представляє кольори у формі, яка більш придатна для фільтрації.

Фільтрація кольорів: Два фільтри застосовуються для виділення об'єктів на зображенні. Для цього використовуються діапазони кольорів у просторі HSV.

Фільтрація кольорів(Приклад коричневого фільтру)

*lower\_brown=np.array([10,50,50]) upper\_brown=np.array([30,255,255])upper\_brown=np.array([30,255,255]) brown\_mask=cv2.inRange(hsv,lower\_brown,upper\_brown)*

Покращення якості зображення: Використовується покращення якості зображення, включаючи перетворення в градації сірого, вирівнювання гістограми та розмиття.

Конвертація в градації сірого: *gray=cv2.cvtColor(result,cv2.COLOR\_BGR2GRAY)*

Вирівнювання гістограми:

*equalized=cv2.equalizeHist(gray)*

Розмиття:

*blurred=cv2.medianBlur(equalized,5)*

Бінаризація і виявлення контурів: Зображення бінаризується, роблячи всі пікселі або чорними, або білими, в залежності від їх яскравості. Потім за допомогою методу знайдення контурів знаходяться зовнішні контури об'єктів на зображенні.

Бінаризація: *\_,thresh=cv2.threshold(blurred,127,255,cv2.THRESH\_BINARY)*

Ерозія та дилатація для покращення контурів: *thresh=cv2.erode(thresh,kernel,iterations=1) thresh=cv2.dilate(thresh,kernel,iterations=1)*

Ідентифікація бочок палива за геометричною ознакою: Для кожного знайденого контуру обчислюються параметри, такі як координати верхнього лівого кута та ширина та висота обмежуючого прямокутника. Ці параметри можна використовувати для визначення форми об'єкта.

*x, y, w, h = cv2.boundingRect(contour), де:*

*x = координата X верхнього лівого кута контуру*

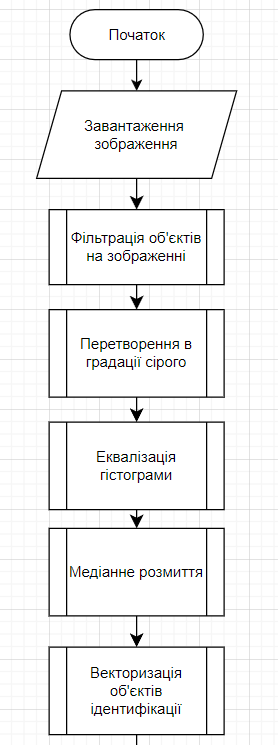
*y = координата Y верхнього лівого кута контуру*

*w = ширина обмежуючого прямокутника контуру*

*h = висота обмежуючого прямокутника контуру*

**3.2. Блок схема алгоритму та її опис.**

Блок-схема розв’язку матиме вигляд на рис. 1.



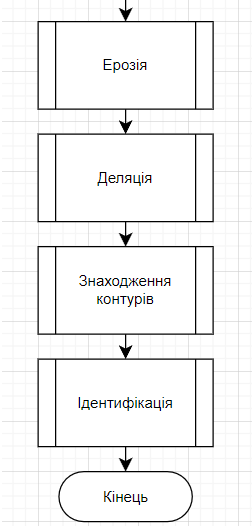


Рис. 1. Блок-схема логіки програми first.py

**3.3. Опис структури проекту програми**

|  |
| --- |
|  |
| Рис.2. Структура проекту. |

google\_map\_image.png – зображення для зміни

Holovnia\_lr\_4.docx – звіт

first.py - файл програми

**3.4. Результати роботи програми відповідно до завдання.**

Для виконання роботи було обрано тестове зображення нафтесховища, ідентифікуватись будуть об’єкти збереження нафти – бочки з паливом



Рис. 3. Зображення нафтосховища GoogleMap

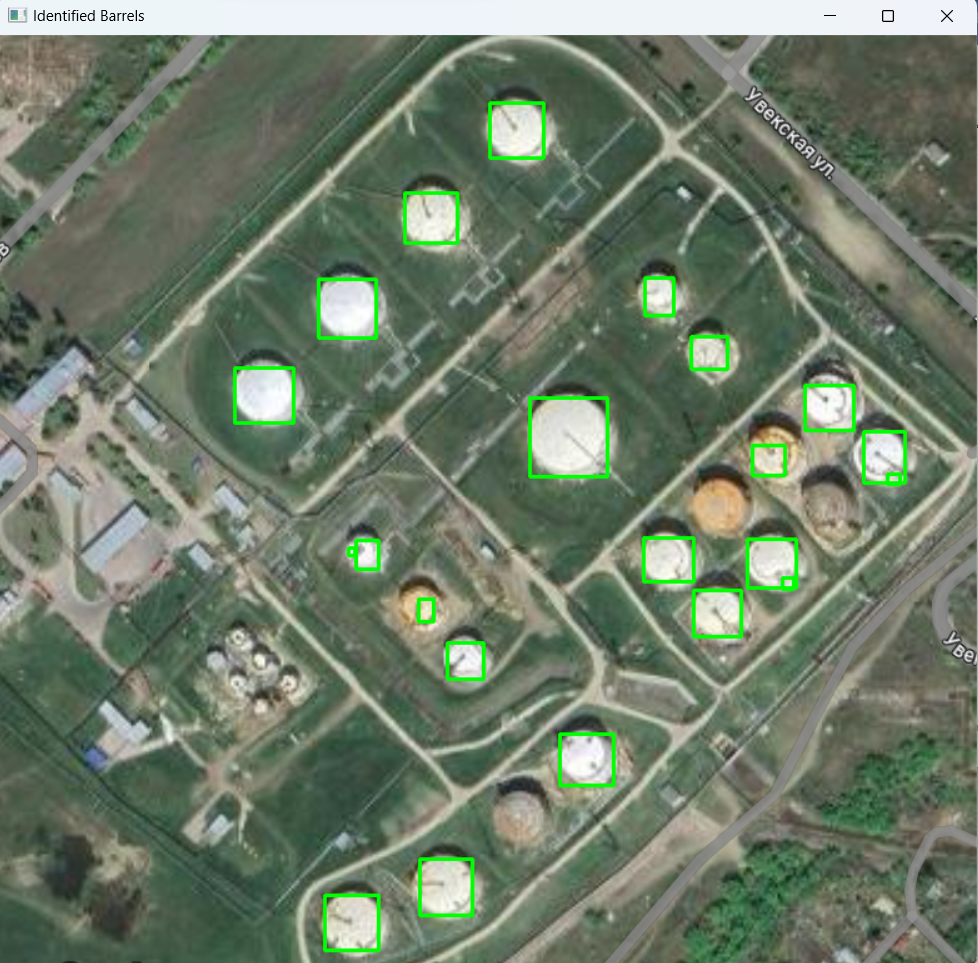


Рис. 4. Ідентифікація бочок з паливом на зображенні (Зображення після змін)

**3.5. Програмний код.**

**First.py**

import cv2

import numpy as np

# Завантаження зображення Google Карт

image = cv2.imread('google\_map\_image.png')

# Фільтрування об'єктів на зображенні

hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

lower\_brown = np.array([10, 50, 50])

upper\_brown = np.array([30, 255, 255])

brown\_mask = cv2.inRange(hsv, lower\_brown, upper\_brown)

lower\_white = np.array([0, 0, 200])

upper\_white = np.array([255, 50, 255])

white\_mask = cv2.inRange(hsv, lower\_white, upper\_white)

mask = cv2.bitwise\_or(brown\_mask, white\_mask)

result = cv2.bitwise\_and(image, image, mask=mask)

# Покращення якості зображення

gray = cv2.cvtColor(result, cv2.COLOR\_BGR2GRAY)

equalized = cv2.equalizeHist(gray)

blurred = cv2.medianBlur(equalized, 5)

# Векторизація об'єктів ідентифікації

\_, thresh = cv2.threshold(blurred, 127, 255, cv2.THRESH\_BINARY)

kernel = np.ones((5, 5), np.uint8)

thresh = cv2.erode(thresh, kernel, iterations=1)

thresh = cv2.dilate(thresh, kernel, iterations=1)

contours, \_ = cv2.findContours(thresh, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for contour in contours:

# Визначення координат і розмірів квадрата

x, y, w, h = cv2.boundingRect(contour)

# Відображення квадрата навколо ідентифікованого об’єкта

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

# Відображення результатів

cv2.imshow('Identified Barrels', image)

cv2.waitKey(0)

cv2.destroyAllWindows()

**3.6. Аналіз результатів відлагодження та верифікації результатів роботи програми.**

Результати відладки та тестування підтвердили працездатність розробленого коду. Порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу та верифікація функціоналу програмного коду показали, що всі вимоги були виконані у повному обсязі.

**IV. Висновки.**

Під час виконання лабораторної роботи було розроблено програмний скрипт обробки зображень та ідентифікації об’єктів. Досліджено принципи та особливості практичного застосування технологій покращення якості цифрових зображень для задач Computer Vision з використанням спеціалізованих програмних бібліотек.