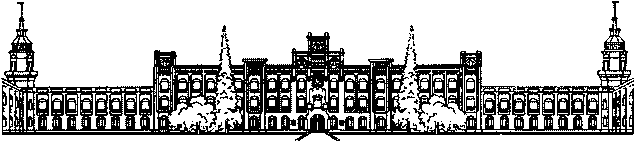
****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №4

**З дисципліни «Технології Computer Vision»**

*ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ПОКРАЩЕННЯ ЯКОСТІ ЦИФРОВИХ ЗОБРАЖЕНЬ ДЛЯ ЗАДАЧ COMPUTER VISION*

| Виконала  студент кафедри ІСТ ФІОТ, групи ІА-12: |  | Перевірив:  пос. Баран Д. Р. |
| --- | --- | --- |
| Яковенко Д. 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:**

*Лабораторія провідної ІТ-компанії реалізує масштабний проект розробки універсальної платформи з цифрової обробки зображень для задач Computer Vision. Платформа передбачає розташування back-end компоненти на власному хмарному сервері з наданням повноважень користувачам заздалегідь адаптованого front-end функціоналу універсальної платформи. Цим формується унікальна для потреб замовника ERP система з технологіями Computer Vision.*

*Замовниками ресурсів платформи є: державні та комерційні компанії, що розробляють медичне обладнання з діагностування захворювань за візуальною інформацією; автоматизації аграрного бізнесу в аспекті обліку посівних територій за даними з БПЛА; візуального контролю безпекових заходів на об’єктах критичної інфраструктури: аеропорти, торгівельно-розважальні центри, житлові комплекси тощо.*

*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 ІІ рівня – максимально 9 балів.**

**Здійснити виконання завдання лабораторної роботи для відеопотоку за варіантами**

**таблиці додатку.**

**ІІІ. Результати виконання лабораторної роботи.**

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

Відповідно до умов задачі синтезовано математичну модель операцій з відеопотоком із автомобілями, що рухаються по автомагістралі. При цьому над нею маємо міст, де рух транспорту здійснюється перпендикулярно руху автомобілів на автомагістралі.

Модель реалізує операцію ідентифікації автомобілів на відео, при цьому ідентифікуючи їх тип (легковий чи вантажний автомобіль).

### Фонове віднімання

fgbg = cv2.createBackgroundSubtractorMOG2(history=500,

varThreshold=50, detectShadows=True)

Використовується об'єкт BackgroundSubtractorMOG2 для створення фонової моделі. Це дозволяє виділити рухомі об'єкти, ігноруючи статичний фон.

### Застосування фонової відмінності

fgmask = fgbg.apply(frame)

На кожен кадр застосовується фонове віднімання, щоб отримати маску рухомих об'єктів.

### Розмиття для зменшення шуму

blurred = cv2.GaussianBlur(fgmask, (5, 5), 0)

Маска розмивається за допомогою гаусівського розмиття, щоб зменшити дрібні шуми і зробити контури об'єктів більш чіткими.

### Порогова обробка

\_, thresh = cv2.threshold(blurred, 200, 255, cv2.THRESH\_BINARY)

Застосовується порогова обробка для перетворення розмитої маски в бінарне зображення, де рухомі об'єкти стають білими, а фон - чорним.

### Морфологічні операції

kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (15, 15))

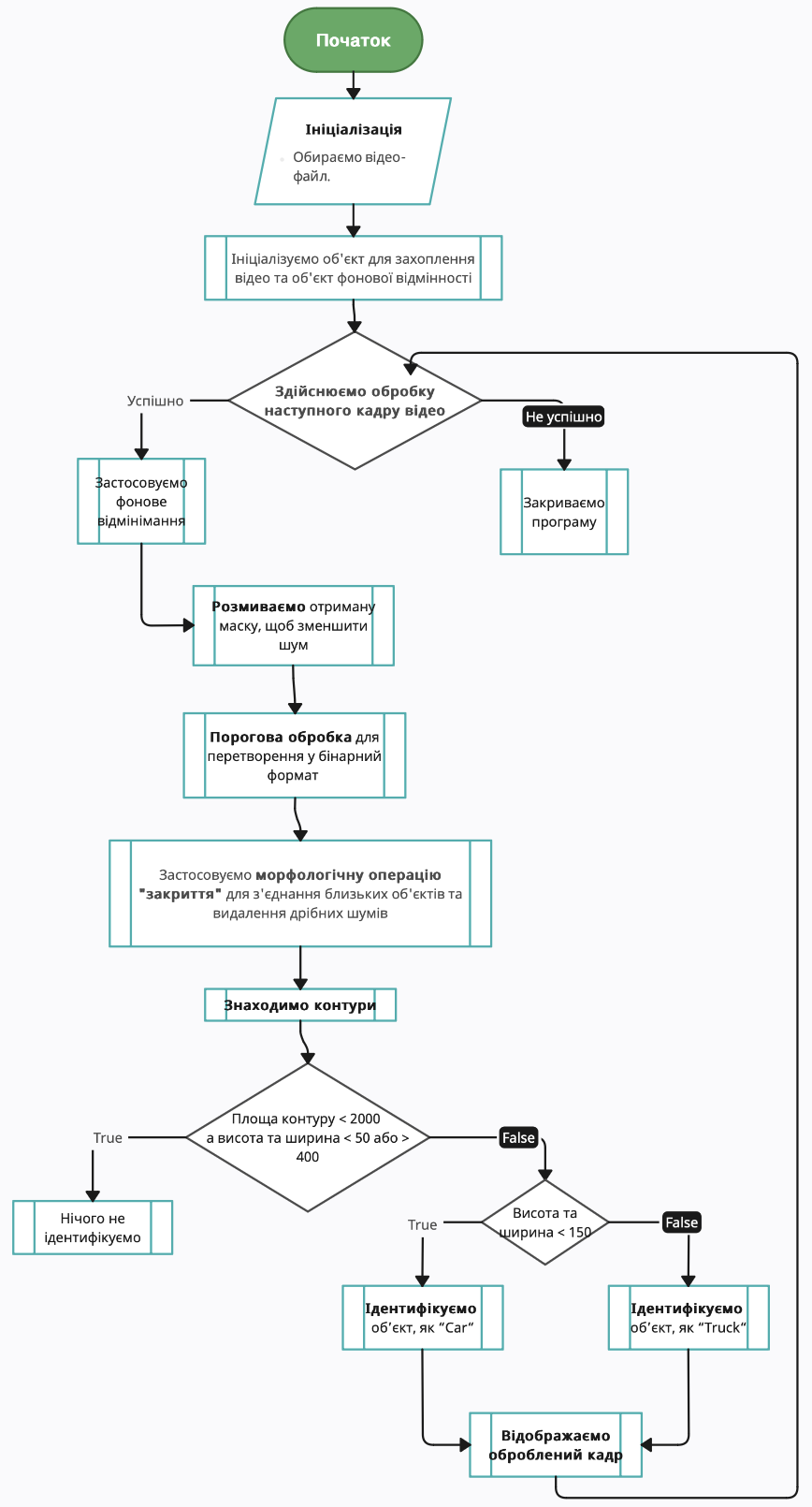
morph = cv2.morphologyEx(thresh, cv2.MORPH\_CLOSE, kernel)

Застосовується морфологічна операція "закриття" для видалення дрібних шумів і з'єднання близьких об'єктів. Використовується структурний елемент розміром 15x15.

Додатково визначаємо обмежувальний прямокутник, що обчислюється для кожного контуру і відповідно обводить автомобіль, далі фільтруємо їх за розміром, аби не виділяти надто малі та великі обʼєкти. Оскільки у нас здійснюється класифікація транспорту на дорозі – задаємо, що розмір легкового авто не перевищує 150x150. І, нарешті, додаємо під квадратом підпис. Завдяки клавіші **q** ми можемо вийти з програми.

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

Застосування синтезованих моделей здійснюється у порядку, що відображає суть алгоритму реалізації завдань лабораторної роботи.



| Рис.1. Блок-схема алгоритму програми для завдання другої складності. |
| --- |
|  |

Розглянемо даний алгоритм. Алгоритм починається з завантаження відеофайлу та ініціалізації об'єкта для захоплення відео. Далі створюється об'єкт фонової відмінності BackgroundSubtractorMOG2 з певними параметрами для виявлення рухомих об'єктів. Після цього починається обробка кадрів відео в циклі, який триває доти, доки не закінчиться відео або не буде виявлено помилку при завантаженні чергового кадру.

На кожному кадрі спочатку застосовується фонове віднімання для виділення рухомих об'єктів, після чого отримана маска розмивається для зменшення шуму. Потім проводиться порогова обробка, яка перетворює розмите зображення у бінарний формат. Застосовуються морфологічні операції, зокрема операція "закриття", для з'єднання близьких об'єктів та видалення дрібних шумів.

На бінарному зображенні знаходяться контури. Для кожного з цих контурів, якщо його площа менша за певний поріг, він ігнорується. Далі для кожного значущого контуру обчислюється обмежувальний прямокутник. Якщо розміри цього прямокутника не відповідають типічним розмірам транспортних засобів (занадто малі або занадто великі), він також ігнорується.

На основі розмірів обмежувального прямокутника проводиться класифікація: якщо ширина і висота менша за певний поріг, об'єкт класифікується як "Car", інакше як "Truck". Для кожного визначеного транспортного засобу малюється обмежувальний прямокутник на кадрі, і над ним додається підпис із типом транспортного засобу.

Оброблений кадр відображається у вікні, і цикл обробки продовжується. Варто зазначити, що робота виконується до тих пір, доки користувач не натисне клавішу 'q', що завершує виконання програми. Наприкінці звільняються всі ресурси, пов'язані із захопленням відео, та закриваються всі відкриті вікна.

На цьому робота алгоритму завершується.

**3.3. Опис структури проекту програми в середовищі PyCharm.**

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

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

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

CVLab4 – головний каталог проекту

lab4.py – файл програмного коду лабораторної роботи;

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

Результатом роботи програми є програма, в якій при її запуску отримуємо такий результат:

|  |
| --- |
| Рис.3. Результат обробки зображення.    Рис.4. Результат обробки зображення (продовження). |

Як бачимо з наданих gif – майже кожен транспорт розпізнається і йому призначається відповідний тип, що залежить від його розміру.

Таким чином, представлені результати у повному обсязі відповідають завданню лабораторної роботи.

**3.5. Програмний код.**

Програмний код послідовно реалізує алгоритми для виконання завдання другого рівня складності.

При цьому використано можливості Python бібліотеки openCV.

import cv2

# Відео файл

video\_path = 'video.mp4'

cap = cv2.VideoCapture(video\_path)

# Створюємо фоновий субтракційний об'єкт

fgbg = cv2.createBackgroundSubtractorMOG2(history=500, varThreshold=50, detectShadows=True)

while cap.isOpened():

ret, frame = cap.read()

if not ret:

break

# Застосовуємо фонове віднімання

fgmask = fgbg.apply(frame)

# Розмиття для зменшення шуму

blurred = cv2.GaussianBlur(fgmask, (5, 5), 0)

# Виконуємо порогову обробку для виділення основних об'єктів

\_, thresh = cv2.threshold(blurred, 200, 255, cv2.THRESH\_BINARY)

# Використовуємо морфологічні операції для видалення дрібних шумів і з'єднання близьких об'єктів

kernel = cv2.getStructuringElement(cv2.MORPH\_RECT, (15, 15))

morph = cv2.morphologyEx(thresh, cv2.MORPH\_CLOSE, kernel)

# Знаходимо контури на зображенні

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

for contour in contours:

# Ігноруємо маленькі контури

if cv2.contourArea(contour) < 2000:

continue

# Обчислюємо обмежувальний прямокутник для кожного контуру

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

# Фільтруємо контури, які не відповідають розмірам транспортних засобів

if w < 50 or h < 50 or h > 400 or w > 400:

continue

# Класифікація на основі розмірів

if w < 150 and h < 150:

vehicle\_type = "Car"

else:

vehicle\_type = "Truck"

# Малюємо прямокутник та підписуємо тип авто

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

cv2.putText(frame, vehicle\_type, (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (0, 255, 0), 2)

# Показуємо кадр

cv2.imshow('Vehicle Detection', frame)

# Вихід при натисканні клавіші 'q'

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

break

cap.release()

cv2.destroyAllWindows()

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

Результати відлагодження та тестування довели працездатність розробленого коду.

Верифікація функціоналу програмного коду, порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу доводять, що завдання виконано у повному обсязі.

**IV. Висновки.**

У ході виконання лабораторної роботи проведено дослідження принципів та особливостей технологій для покращення та обробки зображення (кадру відео). Здійснено обробку кадру таким чином, аби мати змогу розпізнавати транспорт у потоці та визначати тип транспорту. Такі програми можна покращувати, наприклад додати методи, що будуть використовуватися для визначення швидкості автомобіля, тим самим регулюючи швидкісний режим; також для пошуку даних про водія транспортного засобу по номерному знаку.

Виконала: студент Яковенко Д.О.