

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
ЗАСОБІВ СЕГМЕНТАЦІЇ ЗОБРАЖЕНЬ
RESEARCH AND SOFTWARE IMPLEMENTATION OF
IMAGE SEGMENTATION TOOLS

Виконав: студент 2 курсу, групи КНТ-112м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ПЕРЕГУДА В.Б.

(ПРІЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПЕРЕГУДИ Владислава Борисовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів сегментації зображень. Research and Software Implementation of Image Segmentation Tools

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “14” листопада 2023 року № 437

2. Строк подання студентом проєкту (роботи) 01 грудня 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання “16” жовтня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій	2 тиждень	Розділ 2
	розробки		
4	Розробка структури програми.	3 тиждень	Розділ 3
5	Розробка програми.	4-7 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження	8 тиждень	Розділ 4
	програмного забезпечення.		
7	Оформлення пояснювальної записки та документів	9-10 тижні	Додатки
	до неї.		
8	Нормоконтроль та рецензування.	11 тиждень	
9	Захист роботи.	12 тиждень	

Студент(ка)

(підпис) Владислав ПЕРЕГУДА
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Андрій ОЛІЙНИК
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
83 с., 35 рис., 3 табл., 3 дод., 16 джерел.

PYTHON, ЗОБРАЖЕННЯ, МОВА ПРОГРАМУВАННЯ, ОБРОБКА
ЗОБРАЖЕНЬ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, СЕГМЕНТАЦІЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для
обробки зображень.

Предмет дослідження – програмні засоби для сегментації зображень.

Метою роботи є дослідження та розробка програмного забезпечення
для сегментації зображень.

Матеріали, методи та технічні засоби: мова програмування Python,
середовище розробки PyCharm, бібліотеки NumPy, OpenCV, Tkinter.

Результати. Розроблено програмне забезпечення для обробки
зображень шляхом їх сегментації за допомогою мови програмування Python та
середовища розробки PyCharm.

Практична цінність роботи полягає у розробці програмного
забезпечення, що реалізує методи обробки зображень шляхом їх сегментації та
дозволяє виявляти окремі об'єкти на зображеннях.

Висновки. Виконано проєктування програмного забезпечення для
сегментації зображень. Розроблено програмне забезпечення для сегментації
зображень. Здійснено тестування розробленого програмного забезпечення для
сегментації зображень.

Галузь використання – програмні засоби обробки зображень.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 83 pages, 35 figures, 3 tables, 3 appendixes, 16 sources.

PYTHON, IMAGE, PROGRAMMING LANGUAGE, IMAGE PROCESSING, SOFTWARE, SEGMENTATION.

The object of research is the process of developing software for image processing.

The subject of the research is software for image segmentation.

The purpose of this work is to develop the software for image segmentation.

Materials, methods and technical tools: Python programming language, PyCharm development environment, NumPy, OpenCV, Tkinter libraries.

Results. The software tools for image processing by segmenting them using Python programming language and PyCharm development environment have been developed.

The practical value of the work lies in the development of software that implements methods of image processing through their segmentation and allows you to detect of individual objects in images.

Conclusions. The software for image segmentation has been designed. The software for image segmentation has been developed. The developed software for image segmentation has been tested.

Scope of use – software tools for image processing.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Поняття про обробку та сегментацію зображень	11
1.2 Аналіз програмного забезпечення для сегментації зображень	24
1.3 Висновки за розділом 1	32
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для сегментації зображень	36
2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для сегментації зображень	39
2.4 Висновки за розділом 2	40
3 Опис програми	42
3.1 Структура програмного забезпечення для сегментації зображень	42
3.2 Функціонування програмного забезпечення для сегментації зображень	45
3.3 Проектування програмного забезпечення для сегментації зображень ...	47
3.4 Проектування інтерфейсу програмного забезпечення для сегментації зображень	49
3.5 Висновки за розділом 3	51
4 Експлуатація, тестування та експериментальне дослідження програми	52
4.1 Призначення й умови застосування програми	52
4.2 Характеристики програми для сегментації зображень	53
4.3 Інструкція по експлуатації програми	53
4.3.1 Звернення до програми	53
4.3.2 Вхідні й вихідні дані	53
4.3.3 Повідомлення	54
4.4 Виконання програмного забезпечення для сегментації зображень	54

4.5 Тестування програмного забезпечення для сегментації зображень.....	57
4.6 Висновки за розділом 4	58
Висновки	59
Перелік джерел посилання	61
Додаток А Технічне завдання	63
Додаток Б Фрагмент тексту програми	67
Додаток В Слайди презентації	76

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

IDE – Integrated Development Environment, інтегроване середовище для розробки;

PyCharm – інтегроване середовище розробки для мови програмування Python, надає засоби для розробки та аналізу коду;

OpenCV – бібліотека функцій та алгоритмів комп'ютерного зору, обробки зображень і чисельних алгоритмів загального призначення;

UML – Unified Modeling Language, уніфікована мова моделювання

ПЗ – програмне забезпечення.

ВСТУП

У наш час широко використовуються методи та програмні засоби для сегментації зображень, що допомагають ідентифікувати конкретні області на зображеннях різного характеру [1]-[6].

Програмні засоби сегментація зображень допомагають ідентифікувати об'єкти при медичному діагностуванні, допомагають лікарям виявляти патології і приймати діагностичні рішення. У сфері віртуальної та розширеної реальності сегментація допомагає інтегрувати віртуальні об'єкти у реальний світ, роблячи взаємодію більш натуральною. Сегментація зображень на автомобільних дорогах та їх оточенні допомагає автономним автомобілям розрізняти різні об'єкти на дорозі, включаючи інші транспортні засоби, пішоходів, дорожні знаки та світлофори. У метеорології, сегментація зображень допомагає визначати хмарність, опади та інші погодні умови на основі зображень, отриманих зі спостережень [1]-[6]. Для оптимізації сільськогосподарського виробництва використовується сегментація зображень для аналізу ґрунту, визначення стану рослин та виявлення хвороб [1]-[6].

Отже, у наш час програмне забезпечення для сегментації зображень широко використовується у медицині, науці, технологічних рішеннях та інших сферах, де точний аналіз та розпізнавання об'єктів на зображеннях є важливим завданням [1]-[6]. Проте деякі програми для сегментації зображень можуть бути важкими для налаштування, особливо для користувачів без попереднього досвіду в галузі комп'ютерного бачення. Деяке програмне забезпечення для точної сегментації зображень може вимагати потужних обчислювальних ресурсів, що може ускладнити їх використання на менш потужних пристроях. Оскільки різні об'єкти мають різні форми та контексти, не існує універсального підходу для сегментації, і відповідне програмне забезпечення може виявлятися менш ефективними для певних типів об'єктів [1]-[6]. Тому актуальною є розробка програмного забезпечення для сегментації зображень, що використовує різні методи сегментації.

У дипломній кваліфікаційній роботі магістра розв'язується актуальне завдання розробки програмного забезпечення для сегментації зображень.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для обробки зображень шляхом їх сегментації;
- розробити модель взаємодії користувача з програмним забезпеченням для сегментації зображень;
- здійснити проєктування програмного забезпечення для сегментації зображень;
- створити програмне забезпечення для сегментації зображень;
- виконати тестування розробленого програмного забезпечення для сегментації зображень.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття про обробку та сегментацію зображень

Сегментація зображення – це процес розділення цифрового зображення на групи, відомі як сегменти зображення, з метою спрощення зображення та можливості подальшої обробки чи аналізу кожного сегмента окремо. Технічно, сегментація включає присвоєння пікселям міток, які можуть служити для ідентифікації об'єктів, людей або інших значущих аспектів зображення [1]-[6].

Однією з загальних використовуваних сфер сегментації зображень є виявлення об'єктів. Замість обробки усього зображення, часто спочатку застосовується алгоритм сегментації для виявлення всіх цікавих об'єктів на зображенні. Ці об'єкти потім можуть бути оброблені детектором об'єктів в межах області, визначеної алгоритмом сегментації. Це підвищує точність та скорочує час обробки, оскільки детектору потрібно працювати лише з визначеною зоною зображення [1]-[6].

Сегментація зображення виступає як суттєва частина комп'ютерного зору, використовувана у різних практичних контекстах. Вона застосовується в широкому спектрі областей, таких як аналіз медичних зображень, відомості для безпілотних транспортних засобів, розпізнавання обличчя, відеоспостереження, та обробка супутникових знімків [1]-[6].

Суть сегментації зображення полягає в перетворенні вхідного зображення у результат, який представляє собою маску або матрицю, де кожен елемент відображає клас або приналежність до певного об'єкта чи елемента для кожного пікселя [1]-[6].

Для досягнення сегментації використовуються різні високо рівневі методи або евристики зображення, що відіграють ключову роль у загальних алгоритмах сегментації. Ці алгоритми використовують методи кластеризації,

такі як перетворення та аналіз гістограм, для створення моделей сегментації [1]-[6].

Один з прикладів відомої евристики – це використання кольору. Графічні дизайнери можуть застосовувати зелений фон, щоб переконатися, що фонове зображення має однорідний колір, що полегшує роботу програм для визначення фону та проведення змін під час подальшої обробки [1]-[6].

Інший приклад корисної евристики – це контраст. Програма сегментації зображення може легко відрізнити темне зображення від яскравого фону, наприклад, неба. Вона виявляє межі пікселів з високими рівнями контрастності [1]-[6].

Звичайні методи сегментації зображень, що базуються на таких евристичках, можуть бути швидкими та простими, але часто вимагають значних корекцій для адаптації до конкретних сценаріїв використання, розроблених вручну. Вони не завжди забезпечують достатню точність для складних зображень. Нові методи сегментації використовують машинне та глибоке навчання для підвищення точності та гнучкості в цій сфері [1]-[7].

Алгоритми сегментації зображень, що базуються на методах машинного навчання, використовують процес навчання моделей для допомоги програмі визначити відповідні регіони на зображенні. Особливо корисними для завдань сегментації зображень є глибокі нейронні мережі, які дозволяють досягти високої точності в цьому процесі [1]-[7].

Існують різні моделі та підходи до використання нейронних мереж для сегментації зображень, але вони спільно мають такі три основні компоненти [1]-[7]:

- енкодуювальник: цей блок включає послідовність шарів, які поглиблено аналізують зображення за допомогою глибоких і вузьких фільтрів. Цей етап витягає ключові особливості зображення, які пізніше використовуються для сегментації [1]-[7];

- декодер: ця частина перетворює вихід енкодуювальника на маску сегментації, де кожному пікселю призначається певний клас або значення. Це

допомагає визначити, до якого класу належить кожен піксель на зображенні [1]-[7];

– пропуск з'єднань: багато архітектур нейронних мереж мають довгострокові з'єднання, які дозволяють моделі впізнавати об'єкти різних масштабів на зображенні. Це збільшує точність сегментації, оскільки модель може впізнавати деталі на різних рівнях деталізації [1]-[7].

Загалом, використання глибоких нейронних мереж для сегментації зображень дозволяє досягти високої точності і визначити об'єкти на зображеннях з великою деталізацією [1]-[7].

Проаналізуємо різноманітні методи, які застосовуються для сегментації зображень [1]-[7].

Семантична фрагментація організовує пікселі зображення у семантичні категорії. У цьому підході кожен піксель призначається певному класу, і ця модель сегментації не залежить від контексту або іншої інформації [1]-[7]. Для прикладу, розглянемо зображення з деревами та транспортними засобами. Застосування семантичної сегментації вимагатиме групування кожного типу дерева в одну категорію (наприклад, "дерево"), а також групування кожного типу транспортного засобу, такого як автобус, автомобіль або велосипед, в іншу категорію (наприклад, "транспортний засіб"). Це дозволяє створити маску, яка визначає, які області зображення відносяться до окремих категорій. Звісно, такий підхід може стикатися з труднощами, особливо коли одна категорія включає в себе кілька подібних об'єктів. Наприклад, семантична сегментація багатолюдної вулиці може об'єднати всіх людей у клас "люди", не враховуючи індивідуальних рис обличчя чи дій. Такий підхід може бути обмежений в забезпеченні деталізованого розподілу складних зображень [1]-[7].

Сегментація екземплярів – це процес визначення пікселів на зображенні з урахуванням конкретного екземпляра об'єкта, а не лише його класу. В інших словах, алгоритми сегментації не визначають до якої категорії відноситься певна область. Замість цього вони виділяють схожі або частково

накладені області на зображенні, враховуючи межі між окремими об'єктами [1]-[7]. Припустимо, що ми використовуємо модель сегментації для аналізу зображення перевантаженої дороги. Ідеальною ситуацією було б визначити окремі об'єкти в групі, щоб підрахувати кількість повторень на зображенні. Проте модель не може передбачити конкретний регіон або ціль (наприклад, "особа") для кожного з екземплярів [1]-[7].

Сегментація паноптичного типу: паноптична сегментація є новим підходом до виділення об'єктів на зображеннях, який включає в себе елементи як семантичної, так і епістемічної сегментації. Цей метод визначає та класифікує всі об'єкти на зображенні, надаючи їм ідентичність [1]-[7]. Паноптична сегментація є корисною для різних застосунків, які потребують великих обсягів даних для своєї роботи. Наприклад, для безпілотних автомобілів важливо швидко та точно розрізняти об'єкти навколишнього середовища. Це досягається завдяки використанню алгоритмів паноптичної сегментації, які обробляють живий потік зображень [1]-[7].

Розглянемо техніки сегментації зображень. Сегментація на основі країв є популярним методом обробки зображень, який визначає краї різних об'єктів на зображенні. Він допомагає виділити важливі аспекти зображення, використовуючи інформацію про його контури. Виявлення контурів може призвести до видалення надлишкової інформації зі зображення, зменшити його розмір та спростити подальший аналіз [1]-[7].

Алгоритми сегментації на основі контурів визначають ці контури, враховуючи відмінності в контрасті, текстурі, відтінку та насиченості. Їх можна зобразити у вигляді ланцюжків контурів, які формуються навколо меж об'єктів на зображенні [1]-[7].

Сегментація на основі порогового значення є найпростішим методом сегментації зображення, коли точки зображення сегментуються відповідно до їхньої інтенсивності щодо певного значення або порогу. Ідеально підходить для сегментації об'єктів із більшою інтенсивністю, ніж інші об'єкти чи фони. Порогове значення може діяти як константа в зображеннях з низьким рівнем

шуму. У деяких випадках можна використовувати динамічні пороги. Поріг ділить зображення у градаціях сірого на дві частини на основі їх співвідношення з порогом, утворюючи бінарне зображення [1]-[7].

Сегментація на основі регіонів включає розбиття зображення на окремі регіони з подібними властивостями. Кожен регіон представляє собою групу пікселів, яку алгоритм визначає від певної початкової точки. Після визначення початкової точки алгоритм може розширювати або зменшувати розмір регіону, додаючи або з'єднуючи пікселі з іншими точками [1]-[7].

Кластеризаційна сегментація. Алгоритми кластеризації є контрольованими методами класифікації, які сприяють виявленню прихованих ділянок на зображенні. Вони підсилюють здатність людського зору, виділяючи кластери, тіні та текстурні деталі. Алгоритм розбиває зображення на групи пікселів із схожими характеристиками, відокремлюючи елементи даних та групує подібні елементи [1]-[7].

Алгоритми сегментації водних ресурсів використовують відмінності у відтінках сірого на зображеннях. Вони аналогічні алгоритмам визначення топографічних рельєфів, де яскравість пікселів може слугувати ознакою їх висоти. Ці методи визначають контури та лінії, що формують басейни, і встановлюють межі між ними. Зображення розбивається на різні області на основі яскравості пікселів, а подібні пікселі групуються [1]-[7]. Алгоритми сегментації по аналогії з водними ресурсами має важливе практичне застосування, зокрема при обробці медичних зображень. Наприклад, це може допомогти виділити та аналізувати різницю між ділянками найсвітліших та найтемніших областей на зображеннях, отриманих з магнітно-резонансної томографії, що сприяє більш точній діагностиці та аналізу [1]-[7].

Процес сегментації полягає у розділенні або виділенні окремих частин зображення, які називаються сегментами. Рекомендується не обробляти усе зображення в одному блоку, оскільки деякі області на зображенні можуть бути дещо неінформативними. Розділяючи зображення на окремі сегменти, можна фокусуватися на обробці конкретних частин [1]-[7].

Зображення може розглядатися як сукупність окремих пікселів. За допомогою методів та засобів сегментації пікселі зі схожими характеристиками групуються [1]-[7].

Процес виявлення об'єктів створює межі, які відповідають кожному класу об'єкта на зображенні. Однак це не завжди дає повну інформацію. Наприклад, можна отримати лише координати меж об'єктів, але цього, як правило, недостатньо, оскільки часто потрібні додаткові деталі, які виходять за рамки цього опису [1]-[7].

Сегментація зображення включає створення докладної маски пікселів для кожного окремого об'єкта на зображенні. Це інноваційний підхід, який дозволяє більш детально ідентифікувати та виділяти об'єкти на фотографії або іншому зображенні [1]-[7].

Важливе значення в цьому процесі мають методи сегментації зображення. Вони допомагають підходити до аналізу з більш глибокого погляду, отримуючи більш важливі результати [1]-[7].

Технологія сегментації зображення вносить зміни в різні галузі, такі як:

- регулювання дорожнього руху;
- розвиток автономних автомобілів;
- визначення положення об'єктів на супутникових фотографіях [1]-[7].

Також існують інші сфери, де використання сегментації зображення може бути корисним [1]-[7].

Види компонентів зображення можна взагалі класифікувати на дві категорії [1]-[7]:

- смисловий розподіл: визначення зон або областей зображення за їх змістом та значущістю;
- індивідуальне виділення: виділення окремих об'єктів чи екземплярів на зображенні для подальшого аналізу [1]-[7].

Групування за регіонами: зручним підходом до розділення об'єктів може бути використання їхніх значень пікселів. Важливо відзначити, що значення пікселів фону об'єктів та зображень можуть змінюватися, особливо

якщо між ними існує значна контрастність [1]-[7]. У такому випадку можна встановити порогове значення. Піксельні значення, які нижче або вище цього порогу, можна класифікувати відповідно (як об'єкти або фон). Цей метод відомий як порогова сегментація [1]-[7]. Проте цей метод має свої обмеження. Якщо відтінки сірого майже однакові або значення пікселів сірого неоднозначні, буде важко досягти точної сегментації [1]-[7]. Між двома сусідніми областями завжди є межі з різними значеннями відтінків сірого. Краї можна розглядати як нестійкі локальні особливості зображення. Можна використовувати ці обмеження для виявлення країв і таким чином визначити межі для об'єкта, що допомагає визначити кількість кількох об'єктів на зображенні [1]-[7]. Для цього, як правило, використовуються фільтри та згортки [1]-[7].

Значення вагової матриці описують результати згортки. Вибір певних значень для цих вагових матриць дозволяє мати горизонтальні або вертикальні грані (або комбінацію горизонтальних і вертикальних граней) [1]-[7].

Таким чином, сегментація зображення є процесом розділення зображення на декілька значущих та однорідних частин або об'єктів на основі їх внутрішніх характеристик, таких як колір, текстура, форма чи яскравість. Метою сегментації є спрощення або зміна способу подання зображення, щоб зробити його більш зрозумілим та піддатливим для аналізу. Кожен піксель на зображенні визначається. Пікселям, які належать до одного класу, надається спільний ідентифікатор [1]-[7]. Процес сегментації можна здійснити двома способами [1]-[7]:

- заснований на подібності: цей підхід полягає у створенні сегментів шляхом визначення подібності між пікселями зображення. Часто це виконується за допомогою порогових значень (детальніше про це нижче). Алгоритми машинного навчання, такі як кластерний аналіз, базуються на цьому підході до сегментації зображень [1]-[7];

- заснований на розривності: у цьому випадку сегменти зображення формуються відповідно до змін в інтенсивності пікселів на зображенні. Цей

метод використовується для визначення ліній, точок та контурів, які можуть бути оброблені для отримання остаточного результату сегментації зображення [1]-[7].

Система сегментації зображень поділяється на три основні категорії в залежності від кількості та характеру отриманої інформації [1]-[7]. Об'єкти – це окремі елементи на зображенні, які можуть бути виділені за особливим унікальним ідентифікатором, у той час як фон представляє набір об'єктів на зображенні, такі як небо, водні поверхні і так далі. Ясне розмежування між об'єктами та фоном сприяє легшому сприйняттю різних застосувань сегментації зображень та їх використанню [1]-[7].

Сегментація вибірки – це метод сегментації зображень, який зосереджений на визначенні та виділенні всіх об'єктів на зображенні. Це подібно до процесу виявлення об'єктів, проте основна мета полягає в розмежуванні меж об'єктів. Цей підхід не передбачає класифікацію площин, але дозволяє відрізняти об'єкти, що перетинаються. Використання сегментації вибірки особливо корисне в програмах, де необхідно ідентифікувати та відстежувати окремі об'єкти [1]-[7].

Семантична сегментація – це форма сегментації зображення, в якій кожен піксель на зображенні маркується міткою класу без залежності від контексту. Мета полягає в тому, щоб призначити ідентифікатор кожному пікселю на зображенні, що надає зображенню щільну мітку. Алгоритми семантичної сегментації обробляють вхідне зображення, створюючи карту сегментації. Цей підхід особливо корисний там, де важливо розрізняти різні функції на зображенні [1]-[7].

Паноптична сегментація – це комбінація семантичної та епістемічної сегментації. Це означає, що кожен піксель маркується міткою класу, а також ідентифікується кожний екземпляр об'єкта на зображенні. Цей підхід до сегментації зображень забезпечує найвищу якість даних для алгоритмів машинного навчання. Використання паноптичної сегментації особливо корисне в додатках, де моделі комп'ютерного зору повинні розпізнавати різні

об'єкти в навколишньому середовищі та взаємодіяти з ними, наприклад, в автономних роботах [1]-[7].

Тривалий час у галузі комп'ютерного зору застосовувалися традиційні методи сегментації зображень для видобутку значущої інформації з візуальних даних. Ці методи ґрунтуються на математичних моделях та алгоритмах, що визначають області на зображенні з подібними характеристиками, такими як кольори, текстури чи яскравість. Такі традиційні методи сегментації зображень, як правило, є ефективними в обчисленнях та прості у виконанні. Їх часто використовують у програмах, які вимагають швидкої та точної сегментації зображень, наприклад, для виявлення об'єктів, відстеження або розпізнавання [1]-[7].

Метод порогового значення є однією з базових технік сегментації зображень. Під час використання цього методу, пікселі на зображенні розділяються на різні класи в залежності від їхніх інтенсивностей відносно певних заданих порогових значень. Ця техніка відповідає за поділ об'єктів на дві категорії. Метод особливо ефективний для ситуацій, де велика різниця між інтенсивностями пікселів двох різних об'єктів [1]-[7].

У випадках зображень з мінімальним рівнем шуму, порог може залишатися постійним, але в зображеннях з високим рівнем шуму використання динамічного порогу може бути доречним. Під час сегментації за методом порогового значення, градації сірого зображення поділяються на дві частини відносно порогового значення. Це призводить до створення бінарного (двійкового) зображення. На цих двійкових зображеннях можуть застосовуватися методи виявлення та розпізнавання країв об'єктів [1]-[7].

Двома найпоширенішими методами порогового значення є такі. Глобальне порогове значення: цей метод використовується для розділення зображення на передній та фоновий регіони на основі значень інтенсивності пікселів. Визначене порогове значення розділяє дві області, призначаючи пікселі з вищою інтенсивністю області переднього плану, а інші пікселі – області фону. Цей метод простий та ефективний, але може показати недоліки

на зображеннях із змінним освітленням чи контрастом. В таких випадках більш прийнятними можуть бути адаптивні методи порогового значення [1]-[7]. Загалом, метод порогового значення є одним з основних підходів до сегментації зображень, що заснований на використанні порогів для розділення об'єктів на зображенні [1]-[7].

Метод адаптивного порогового значення є підходом, що використовується в процесі сегментації зображень для розділення зображення на області переднього та фонового плану шляхом динамічної настройки порогових значень на основі характеристик самого зображення. Цей метод включає в себе визначення порогу для кожної конкретної області або невеликого блоку, враховуючи статистику значень пікселів у цьому блоку. Адаптивний підхід до порогової обробки особливо ефективний для зображень зі змінним освітленням або контрастом. Зазвичай цей метод застосовується при скануванні документів, процесі бінаризації або сегментації зображень. Вибір конкретного підходу до порогової обробки залежить від вимог конкретної програми та особливостей самого зображення [1]-[7].

Сегментація заснована на регіонах є методом обробки зображень, який застосовується для розділення зображення на окремі регіони відповідно до спільних характеристик, таких як колір, текстура або яскравість. В цьому процесі пікселі групуються в регіони або кластери відповідно до їх подібності, після чого регіони можуть об'єднуватися або розділятися для досягнення бажаного рівня деталізації сегментації. Існують два основних методи сегментації на основі регіонів [1]-[7].

Метод розділення та об'єднання є технікою, яка включає поділ зображення на менші області досягнення критерію зупинки, а потім об'єднання схожих областей для створення більших областей. Цей метод включає розбиття зображення на малий блоки або регіони, а потім об'єднання суміжних областей, що відповідають певним характеристикам, таким як колір чи текстура. "Розділити та об'єднати" є простим і потужним інструментом для

сегментації зображень, але він може бути менш ефективним для складних зображень з перекриттям або нерівномірними областями [1]-[7].

Сегментація на основі розділення зображення на частини використовує інформацію про краї або межі об'єктів на зображенні. В цьому методі зображення може бути представлено як граф, де окремі пікселі представлені як вузли, а різниці між пікселями, як ребра. Після цього граф розділяється на області, шляхом зменшення витрат, створення нормального розрізу або побудови мінімального дерева [1]-[7].

Сегментація на основі периферії використовується в обробці зображень для виділення та відокремлення країв об'єктів на зображенні від фону. Цей метод ґрунтується на виявленні різкої зміни інтенсивності або кольору пікселів на зображенні і використанні цього для визначення межі об'єкта на зображенні [1]-[7].

Також є метод виявлення країв, який використовує багатоетапний підхід для визначення країв на зображенні [1]-[7]. Цей метод включає згладжування зображення з використанням фільтра Гауса, обчислення градієнту та його напрямку, підсилення контурів за допомогою не-максимального пригнічення та відкидання слабких країв з використанням порогового значення гістерезису [1]-[7].

Ці методи сегментації на основі регіонів дозволяють розділяти зображення на частини відповідно до спільних ознак, що сприяє аналізу та обробці зображень у більш зрозумілому контексті [1]-[7].

Підходи з використанням глибокого навчання та нейронних мереж також пропонують варіанти для сегментації зображень, де нейронні мережі вчаться розпізнавати ключові особливості зображень без залежності від передбачених функцій, як це роблять традиційні алгоритми. Зазвичай, нейронні мережі для завдань сегментації використовують структуру "кодувальник-декодер". Кодувальник використовує малим та глибокими фільтрами для виділення особливостей з зображення. Якщо кодувальник раніше був навчений виконувати певне завдання, наприклад, розпізнавання

зображень або облич, то ці знання використовуються для вилучення функцій, які потрібні для сегментації (це називається навчанням з передачею знань). Далі декодер перетворює вихід від кодувальника на маску сегментації на рівнях, які відповідають деталізації пікселів вхідного зображення [1]-[7].

Для сегментації зображень використовуються різні моделі на основі нейронних мереж. Модель U-Net є модифікованою глибокою нейромережею, часто рекомендованою для медичних застосувань. Структура такої моделі містить як кодувальник, так і декодер. Кодувальник використовується для виділення особливостей шляхом швидкого з'єднання, навпаки від звичайних згорткових мереж, які використовують передискретизацію для виділення особливостей. Архітектура U-Net спроектована так, щоб кодувальники та декодери могли ефективно обмінюватися інформацією, забезпечуючи мережі доступ до більш докладних даних шляхом комбінування високорівневих та низькорівневих функцій. Це підвищує точність роботи мережі [1]-[7].

Модель SegNet є глибокою згортковою мережею, спеціально створеною для піксельної семантичної сегментації. Подібно до U-Net, архітектура SegNet включає в себе блоки кодувальника та декодера. Однак особливістю SegNet є використання пробних функцій через свій декодер. Мережа декодера використовує індекси максимального об'єднання, розраховані на рівні згортки, що вимагає кодувальника здійснювати нелінійну підвищену дискретизацію. Це усуває необхідність вивчення передискретизації. Основна мета SegNet полягає в розпізнаванні сцен, що робить його важливим інструментом для різних застосувань [1]-[7].

Для сегментації зображень використовується модель DeepLab, яка є архітектурою нейронної мережі згорткового типу, що використовує особливості кожного згорткового блоку та об'єднує їх у власний блок деконволюції. Вона відрізняється від інших мереж тим, що використовує особливості останнього згорткового блоку та має надмірну дискретизацію, подібну до повністю згорткової мережі. Основною перевагою є зменшення

обчислювальних витрат при одночасному збереженні більшої кількості інформації [1]-[7].

Сегментація зображень має різноманітні практичні застосування, такі як медичне діагностування, робототехніка, управління автономними транспортними засобами та системи спостереження [1]-[7]:

- медичне діагностування: у медичній візуалізації сегментація зображень використовується для виявлення пухлин, виділення органів та діагностики хвороб. Це допомагає у плануванні лікування та моніторингу ходу захворювання [1]-[7];

- робототехніка: сегментація зображень є ключовою для розпізнавання та маніпулювання об'єктами в робототехніці. Роботи можуть використовувати сегментацію для розпізнавання зокрема інструментів або деталей в індустріальному середовищі [1]-[7];

- автономні транспортні засоби: у розробці автономних транспортних засобів сегментація зображень є важливою для виявлення та класифікації об'єктів у навколишньому середовищі, таких як інші транспортні засоби, пішоходи та перешкоди. Точна сегментація є важливою для безпечної та ефективної автономної навігації [1]-[7];

- системи спостереження: у відео спостереженні сегментація зображень використовується для ідентифікації та відстеження об'єктів та людей у реальному часі. Сегментація допомагає ідентифікувати об'єкти інтересу, які можуть бути підозрілими або небезпечними [1]-[7];

- сільське господарство: у сільському господарстві сегментація зображень використовується для моніторингу врожаю, виявлення хвороб та прогнозування врожайності. Це допомагає фермерам приймати обґрунтовані рішення щодо вирощування культур та підвищує врожайність [1]-[7];

- мистецтво та дизайн: сегментація зображень застосовується в мистецтві та дизайні для виконання різноманітних завдань, таких як маніпулювання зображеннями, коригування кольорів та передача стилю. Це

дозволяє виділити об'єкти або області зображення для застосування різних ефектів та трансформацій [1]-[7].

Визначено, що сегментація зображень представляє собою потужний інструмент, який дозволяє виділяти та ізолювати об'єкти або області на зображенні, та має широке застосування в медичній візуалізації, робототехніці та області комп'ютерного зору. Досліджено різноманітні підходи до сегментації зображень, включаючи традиційні методи, такі як порогове значення, регіональна сегментація, сегментація на основі контурів та кластеризація, а також методи машинного навчання, глибокі нейронні мережі. Досліджено методи та моделі використання нейронних мереж для сегментації зображень. Розглянуто метрики оцінки та набори даних, які використовуються для оцінювання ефективності алгоритмів сегментації.

1.2 Аналіз програмного забезпечення для сегментації зображень

Проаналізуємо програмне забезпечення підтримки для сегментації зображень [8]-[11].

Програмне забезпечення Sefexa Image Segmentation Tool призначено для виконання сегментації зображень [9]. Програма Sefexa Image Segmentation Tool дозволяє виділяти та ізолювати окремі об'єкти, області або структури на зображенні. Sefexa Image Segmentation Tool є важливим інструментом для роботи зі зображеннями, який знаходить своє застосування в різних галузях, таких як медична діагностика, комп'ютерний зір, робототехніка та багато інших. Програма Sefexa Image Segmentation Tool дозволяє вчені та фахівцям аналізувати та обробляти зображення, робити їх більш інформативними та аналізувати різні аспекти об'єктів на зображенні. Програмне забезпечення Sefexa Image Segmentation Tool (рис. 1.1) забезпечує можливість ретельної роботи з даними і допомагає досягти більш точних та ефективних результатів в області обробки зображень [9].

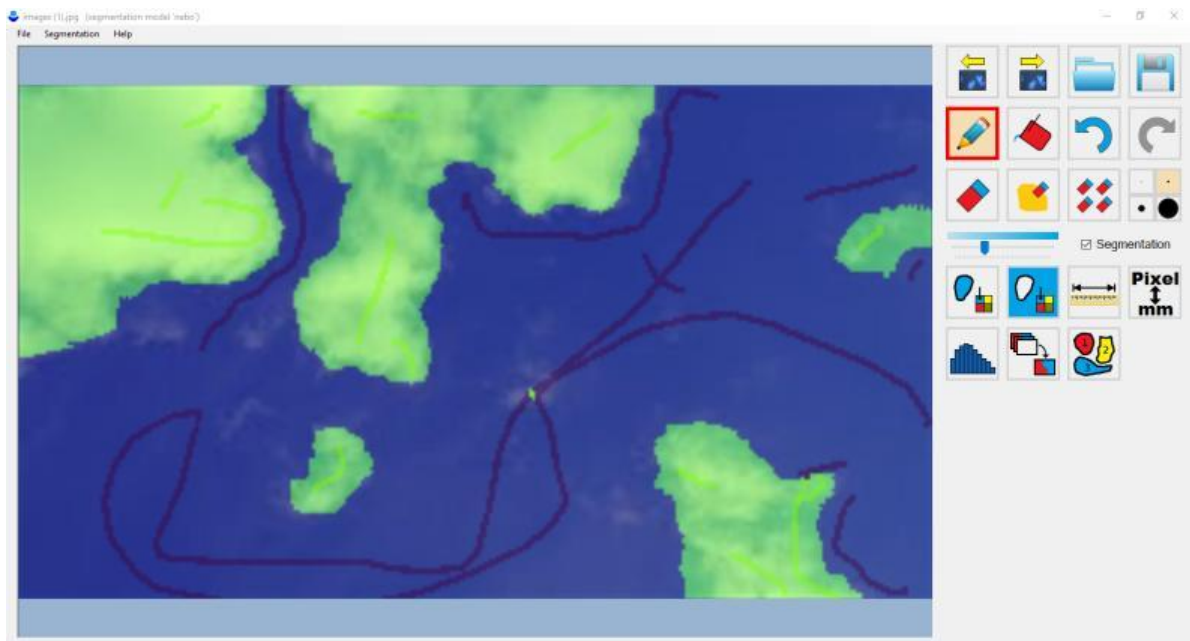


Рисунок 1.1 – Програмне забезпечення Sefexa Image Segmentation Tool [9]

Серед особливостей програмного забезпечення Sefexa Image Segmentation Tool для сегментації зображень можна виділити такі [9]:

- підтримка різноманітних методів сегментації: програма Sefexa Image Segmentation Tool пропонує різні методи сегментації зображень, включаючи традиційні підходи, такі як порогова сегментація, регіональна сегментація, сегментація на основі країв і кластеризація. Крім того, вона також підтримує сучасні методи машинного навчання та глибокого навчання для більш точних результатів [9];

- підтримка різних практичних галузей: програма Sefexa Image Segmentation Tool призначена для використання в різних галузях, включаючи медицину, комп'ютерний зір, робототехніку, наукові дослідження та багато інших. Вона пристосована до використання в різних сценаріях та допомагає зробити аналіз зображень більш ефективним та точним [9];

- засоби візуалізації результатів: програма Sefexa Image Segmentation Tool надає можливість візуалізувати результати сегментації, дозволяючи користувачам побачити, як об'єкти чи області були виділені та ізольовані на зображенні. Це полегшує аналіз та оцінку точності сегментації [9];

– автоматизований аналіз: програмне забезпечення Sefexa Image Segmentation Tool може автоматично аналізувати великі обсяги зображень, допомагаючи зберегти час та зусилля. Воно може бути корисним для обробки великої кількості зображень у наукових дослідженнях чи інших проектах [9];

– підтримка різних наборів даних: програмне забезпечення Sefexa Image Segmentation Tool може працювати з різними наборами даних для оцінки та порівняння алгоритмів сегментації. Це дозволяє вибирати оптимальний підхід для конкретного завдання [9].

Загалом, програмне забезпечення Sefexa Image Segmentation Tool є потужним інструментом для сегментації зображень, який допомагає досягти точних та якісних результатів у великому спектрі застосувань [9].

Серед недоліків програмного забезпечення Sefexa Image Segmentation Tool можна відзначити те, що це програмне забезпечення є досить складним для використання, особливо для користувачів без попереднього досвіду у сегментації зображень або програмуванні. Крім того, ця програма є досить вибагливою до обладнання, оскільки для її ефективної роботи необхідно мати досить високопродуктивне апаратне забезпечення, особливо якщо для сегментації використовуються складні методи глибокого навчання. Це може створити обмеження для користувачів з менш потужними комп'ютерами або вимагати досить багато часу на виконання сегментації зображень [9].

Програмне забезпечення Image Segmenter є програмним модулем у середовищі Matlab, що призначений для сегментації зображень. Програмне забезпечення Image Segmenter дозволяє підтримувати процес виділення об'єктів чи областей на зображенні, що має важливі застосування у багатьох галузях, включаючи медичну візуалізацію, комп'ютерний зір, аналіз зображень та інші [10]. Використання інструмента для сегментації зображень Image Segmenter може вимагати послідовного підходу, де можна використати декілька варіантів сегментації [10]. Важливо враховувати, що різні методи сегментації можуть бути ефективнішими для різних типів зображень. Після проведення сегментації зображення за допомогою програмного забезпечення

Image Segmenter (рис. 1.2), результат можна зберегти у вигляді бінарної маски. Крім того, можливо отримати вихідний код для використання інструмента сегментації зображень, який дозволяє створювати такі маски [10].

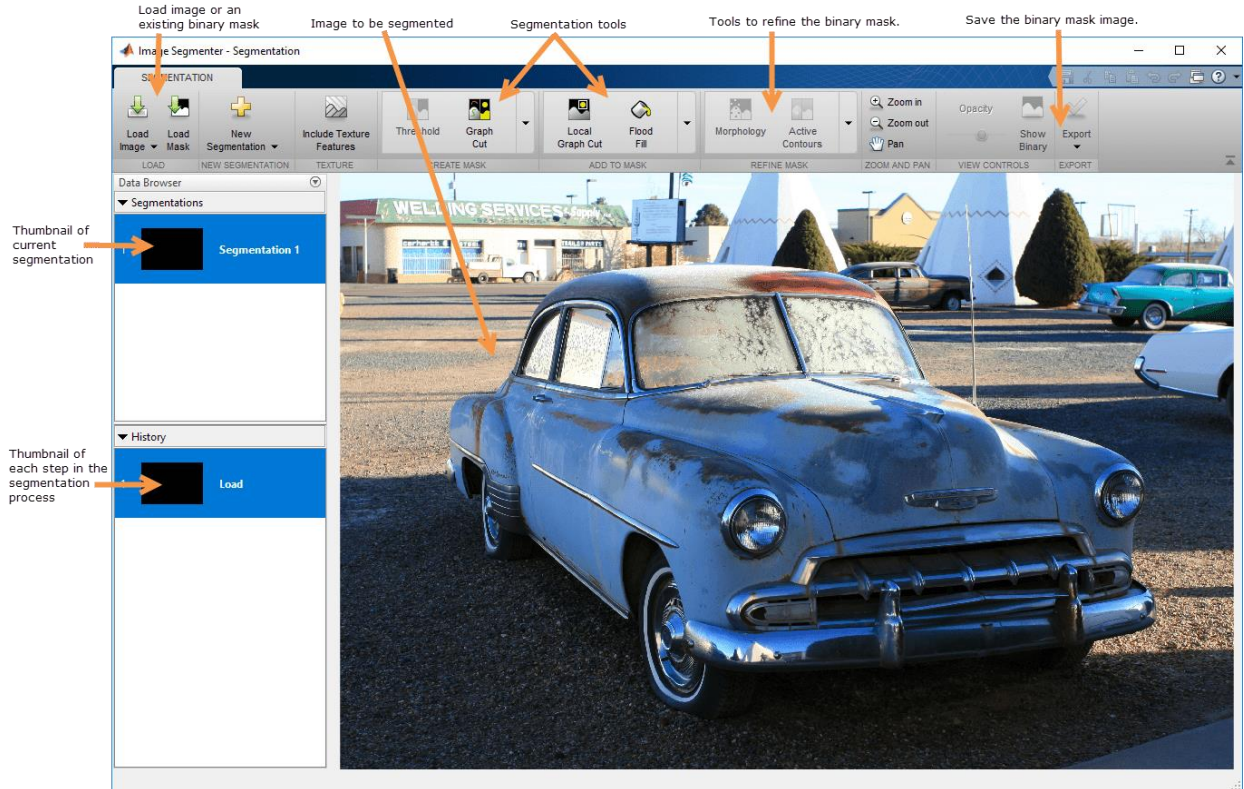


Рисунок 1.2 – Програмне забезпечення Image Segmenter [10]

Програмне забезпечення Image Segmenter дозволяє користувачам використовувати різні методи та алгоритми для проведення сегментації зображень. Основні завдання, які можна вирішити за допомогою цього застосунку, включають [10]:

- семантична сегментація: визначення кожного пікселя на зображенні за його приналежність до конкретного об'єкта чи класу. Це важливо для розпізнавання та аналізу об'єктів на зображенні [10];

- сегментація на основі порогового значення: виділення областей на зображенні, де пікселі мають певне значення яскравості чи кольору, що допомагає виділити об'єкти [10];

- сегментація на основі регіонів: групування пікселів зі схожими характеристиками (якість, кольори тощо) у регіони, що представляють окремі об'єкти чи частини зображення [10];

- сегментація на основі країв: виділення контурів об'єктів на зображенні, що дозволяє виділити їх форму та розмір [10];

- глибинна сегментація: використання глибокого навчання та нейронних мереж для вирішення завдань сегментації [10].

Програмний застосунок Image Segmenter дозволяє користувачам вибирати методи сегментації, налаштовувати параметри та бачити результати в режимі реального часу. Він також надає інструменти для візуалізації результатів та подальшої обробки сегментованих зображень [10].

Оскільки Matlab є популярним інструментом для обробки даних та аналізу зображень, Image Segmenter надає користувачам зручний і потужний інструмент для проведення сегментації зображень у різних галузях досліджень та застосувань [10].

Серед недоліків програмного забезпечення Image Segmenter є необхідність встановлення середовища Matlab для роботи з застосунком, що є досить складним та вимагає додаткових витрат коштів на придбання пакету Matlab. Крім того, програмне забезпечення Image Segmenter є досить складним для використання, особливо для користувачів без попереднього досвіду у сегментації зображень або програмуванні. Також програма Image Segmenter є досить вибагливою до обладнання, оскільки для її ефективної роботи необхідно мати досить високопродуктивне апаратне забезпечення, особливо якщо для сегментації використовуються складні методи глибокого навчання. Це може створити обмеження для користувачів з менш потужними комп'ютерами або вимагати досить багато часу на виконання сегментації зображень [10].

Програмне забезпечення CVIPtools дозволяє виконувати опрацювання та аналіз цифрових зображень у сфері комп'ютерного зору та обробки зображень [11]. Воно надає інструменти для виконання різноманітних

операцій над зображеннями, таких як застосування фільтрів, виділення об'єктів на зображенні, сегментація та інші [11]. CVIPtools дозволяє налаштовувати та вдосконалювати алгоритми обробки зображень, а також досліджувати візуальні дані для отримання важливої інформації. Програмне забезпечення CVIPtools (рис. 1.3) знаходить застосування в різних галузях, таких як медична діагностика, відеоспостереження, розпізнавання образів та графічний дизайн [11].

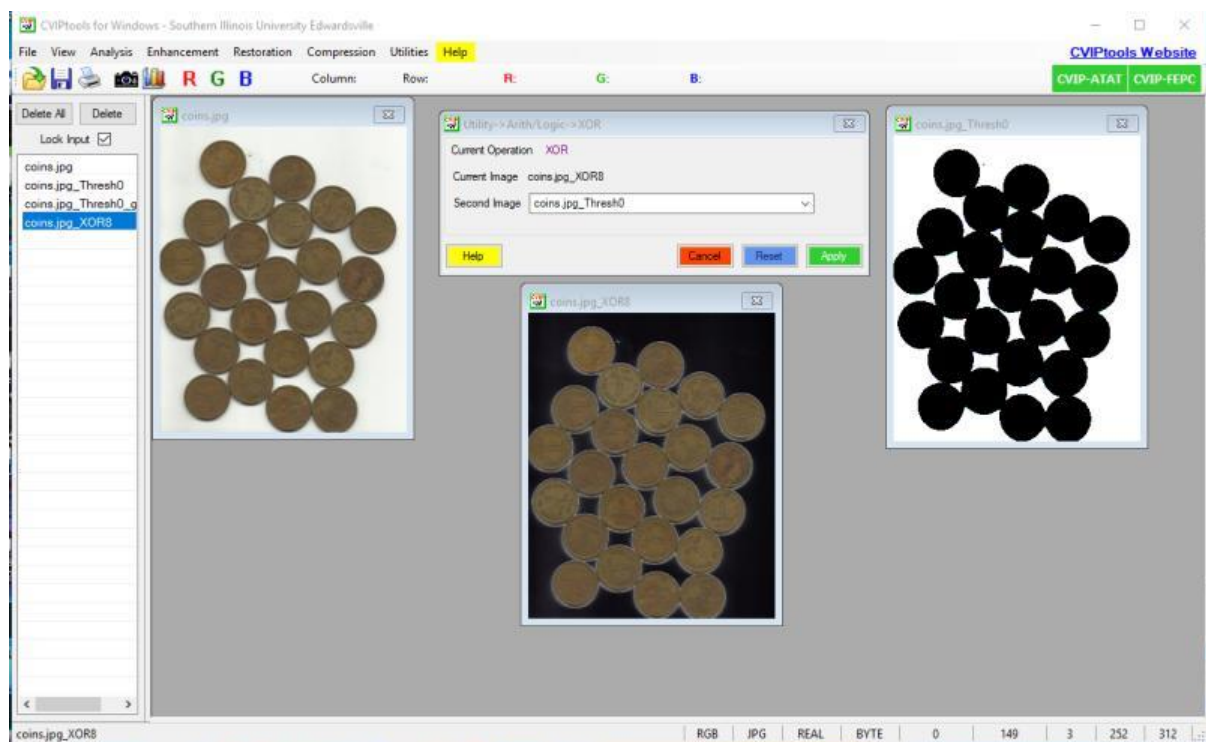


Рисунок 1.3 – Програмне забезпечення CVIPtools [11]

Програмне забезпечення CVIPtools призначено для обробки та аналізу зображень у галузі комп'ютерного зору та обробки зображень. Вона надає засоби для виконання різних операцій над зображеннями, таких як фільтрація, сегментація, виокремлення об'єктів, а також робота з цифровими зображеннями в загальному. CVIPtools дозволяє налагоджувати та вдосконалювати алгоритми обробки зображень, а також досліджувати різноманітні візуальні дані для отримання важливої інформації [11].

Програмне забезпечення CVIPtools має кілька особливостей, які роблять його важливим інструментом у галузі комп'ютерного зору та обробки зображень [11]:

- різноманітні операції обробки зображень: програмне забезпечення CVIPtools надає широкий спектр операцій для редагування та покращення зображень, що включає в себе фільтрацію, підсилення контрастності, зменшення шуму, корекцію кольору та інші [11];

- сегментація та виділення об'єктів: програма CVIPtools дозволяє виконувати сегментацію зображень, що означає виділення окремих об'єктів чи областей на зображенні. Це особливо корисно у медичній діагностиці та інших доменах [11];

- алгоритми обробки: CVIPtools надає можливість працювати з різними алгоритмами обробки зображень, включаючи вдосконалення наявних алгоритмів та розробку нових. Це дозволяє налаштовувати обробку під конкретні завдання [11];

- аналіз зображень: програма дозволяє аналізувати зображення для отримання корисної інформації, можна визначати параметри об'єктів на зображенні, виконувати вимірювання та отримувати статистичні дані [11];

- множина застосувань: CVIPtools знаходить застосування у багатьох галузях, включаючи медичну діагностику, відеоспостереження, розпізнавання образів, графічний дизайн та інші. Ця універсальність робить його корисним інструментом для різних завдань [11];

- гнучкість і налаштування: програмне забезпечення CVIPtools дозволяє користувачам налаштовувати операції обробки зображень під свої потреби. Існує можливість змінювати параметри алгоритмів, що допомагає досягти бажаних результатів [11];

- візуалізація результатів: CVIPtools надає можливість візуалізувати оброблені зображення та результати аналізу за допомогою графіків, діаграм та інших візуальних засобів [11].

Серед недоліків програмного забезпечення CVIPtools можна відзначити те, що це програмне забезпечення є досить складним для використання, особливо для користувачів без попереднього досвіду у сегментації зображень або програмуванні. Крім того, ця програма є досить вибагливою до обладнання, оскільки для її ефективної роботи необхідно мати досить високопродуктивне апаратне забезпечення, особливо якщо для сегментації використовуються складні методи глибокого навчання. Це може створити обмеження для користувачів з менш потужними комп'ютерами або вимагати досить багато часу на виконання сегментації зображень [11].

Порівняльну характеристику програмного забезпечення для сегментації зображень наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для сегментації зображень

Критерій порівняння	Sefexa Image Segmentation Tool [9]	Image Segmenter [10]	CVIPtools [11]
Підтримка широкого набору методів сегментації	+	+	—
Необхідність додаткового забезпечення для функціонування програми	—	+	—
Зручність засобів візуалізації результатів	+—	+	—
Можливість роботи з різними наборами даних	+	+	+—
Автоматизований аналіз великих наборів даних	+	+—	—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для сегментації зображень. Проте деякі програми для сегментації зображень можуть бути важкими для налаштування, особливо для користувачів без попереднього досвіду в галузі комп'ютерного бачення. Деяке програмне забезпечення для точної сегментації зображень може вимагати потужних обчислювальних ресурсів, що може ускладнити їх використання на менш потужних пристроях. Оскільки різні об'єкти мають різні форми та контексти, не існує універсального підходу для сегментації, і відповідне програмне забезпечення може виявлятися менш ефективними для певних типів об'єктів. Тому актуальною є розробка програмного забезпечення для сегментації зображень.

При розробці програмного забезпечення оптимізації для сегментації зображень необхідно забезпечити такі функціональні вимоги:

- можливість завантаження цифрового зображення;
- можливість роботи з різними методами сегментації зображень;
- можливість зміни параметрів методу сегментації зображень при необхідності;
- можливість збереження зображення після його сегментації.
- можливість порівняння результатів сегментації за різними критеріями.

1.3 Висновки за розділом 1

Визначено, що сегментація зображень представляє собою потужний інструмент, який дозволяє виділяти та ізолювати об'єкти або області на зображенні, та має широке застосування в медичній візуалізації, робототехніці та області комп'ютерного зору. Досліджено різноманітні підходи до сегментації зображень, включаючи традиційні методи, такі як порогове значення, регіональна сегментація, сегментація на основі контурів та кластеризація, а також методи машинного навчання, глибокі нейронні мережі.

Досліджено методи та моделі використання нейронних мереж для сегментації зображень. Розглянуто метрики оцінки та набори даних, які використовуються для оцінювання ефективності алгоритмів сегментації.

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для сегментації зображень. Проте деякі програми для сегментації зображень можуть бути важкими для налаштування, особливо для користувачів без попереднього досвіду в галузі комп'ютерного бачення. Деяке програмне забезпечення для точної сегментації зображень може вимагати потужних обчислювальних ресурсів, що може ускладнити їх використання на менш потужних пристроях. Оскільки різні об'єкти мають різні форми та контексти, не існує універсального підходу для сегментації, і відповідне програмне забезпечення може виявлятися менш ефективними для певних типів об'єктів. Тому актуальною є розробка програмного забезпечення для сегментації зображень.

Сформульовано функціональні вимоги до програмного забезпечення для сегментації зображень.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення оптимізації для сегментації зображень було використано мову програмування Python [12]-[14].

Мова програмування Python представляє собою високорівневу інтерпретовану мову, що характеризується зрозумілим і читабельним синтаксисом. Ця мова програмування здобула популярність завдяки своїй легкості використання, можливостям розширення та універсальності [12]-[14].

Особливості мови програмування Python такі [12]-[14]:

- універсальність: мова програмування Python володіє широкою стандартною бібліотекою, яка надає різноманітні інструменти для завдань, включаючи операції з файлами, мережами, базами даних тощо [12]-[14];

- спільнота розробників на мові Python є досить активною. Спільнота розробників сприяє швидкому вирішенню проблем і постійному удосконаленню мови [12]-[14];

- простий синтаксис: синтаксис мови програмування Python схожий на англійську мову, це сприяє читабельності і легкості розуміння коду [12]-[14];

- зручна типізація: у мови програмування Python змінні не вимагають попередньої типізації, що полегшує процес розробки [12]-[14];

- кросплатформенність: мова програмування Python може працювати на різних операційних системах без значних змін у коді [12]-[14];

- можливість розширення: мова програмування Python дозволяє поєднувати код із інших мов програмування, таких як C/C++, для поліпшення продуктивності або використання системних ресурсів [12]-[14];

- різноманітність застосувань: мова програмування Python використовується в веброботці, наукових обчисленнях, аналізі даних, штучному інтелекті, розробці ігор, автоматизації завдань і багатьох інших областях [12]-[14];

– підтримка об'єктно-орієнтованої парадигми: мова програмування Python підтримує парадигму об'єктно-орієнтованого програмування [12]-[14].

Python є популярною мовою програмування завдяки своїм перевагам, активній спільноті розробників і широкому спектру застосувань у сучасному програмуванні [12]-[14].

Отже, мова програмування Python є інтерпретованою мовою програмування, яка базується на об'єктно-орієнтованій парадигмі та відрізняється динамічною типізацією. Високорівневі структури даних разом з гнучкою семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм та для об'єднання існуючих компонентів [12]-[14].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для сегментації зображень наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для сегментації зображень

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C++
Кросплатформенність	+	—	+
Зручна типізація	+	+—	—
Простий синтаксис	+	+—	+
Активна спільнота розробників	+	—	+—
Велика кількість бібліотек для обробки зображень	+	+—	+—
Можливість розширення	+	+—	+

Мова програмування Python, хоча і є універсальною мовою програмування, вона набула широкого використання в обробці та аналізі даних, зокрема в обробці зображень, оскільки існує велика кількість якісних

бібліотек та платформ для інтелектуального аналізу даних, машинного навчання та обробки зображень [12]-[14].

Отже, для реалізації програмного забезпечення для сегментації зображень обрано мову програмування Python, яка є високорівневою інтерпретованою мовою, що характеризується зрозумілим і читабельним синтаксисом, є легкою у використанні, має можливості розширення та універсальності, дозволяє розробляти різні види програмних застосунків.

2.2 Вибір середовища та інших засобів розробки для створення програмного забезпечення для сегментації зображень

В якості середовища розробки для створення програмного забезпечення для сегментації зображень було обрано PyCharm [15], [16].

PyCharm є інтегрованим середовищем розробки (Integrated Development Environment, IDE) призначене для роботи з мовою програмування Python. Розроблене компанією JetBrains, PyCharm відзначається як одне з якісних практичних рішень для розробки різних типів програмного забезпечення на Python [15], [16].

Це IDE володіє усіма ключовими перевагами, характерними для сучасних розробницьких середовищ. Воно включає в себе засоби аналізу програмного коду, виділення синтаксису, зручну навігацію по проекту, функції рефакторінгу, інтеграцію з системами керування версіями, підтримку популярних фреймворків, а також вбудований засіб для налагодження та тестування коду [15], [16].

Інтегроване середовище розробки PyCharm представляє собою інструмент для розробки на мові програмування Python. Цей інструмент покликаний спростити процес розробки програмного забезпечення на Python, надаючи розробникам корисні функції та засоби [15], [16].

Метою PyCharm є надання розробникам практичних та продуктивних інструментів для розробки програм на Python [15], [16]. Основні характеристики цього рішення такі [15], [16]:

- редактор коду: інтегроване середовище розробки PyCharm забезпечує зручне середовище для написання, редагування та форматування коду мовою Python [15], [16];

- засоби автозаповнення та інтелектуальний аналіз: інтегроване середовище розробки PyCharm має функцію автоматичного завершення коду, виявлення можливих помилок та надання варіантів їх виправлення та оптимізації [15], [16];

- засоби відлагодження коду: інтегроване середовище розробки PyCharm включає в себе інструмент відлагодження, що допомагає виявляти та усувати помилки у програмному коді [15], [16];

- підтримка фреймворків: інтегроване середовище розробки PyCharm підтримує такі популярні фреймворки, як Django, Flask та інші, спрощуючи розробку вебзастосунків та інших типів програм [15], [16];

- засоби управління проєктами: інтегроване середовище розробки PyCharm допомагає створювати та керувати проєктами, включаючи відстеження змін у коді [15], [16];

- засоби рефакторингу: інтегроване середовище розробки PyCharm надає інструменти для покращення структури та читабельності коду шляхом змін змінних, методів, класів тощо [15], [16];

- засоби тестування: PyCharm підтримує розробку та виконання тестів для перевірки коректності роботи програм [15], [16];

- інтеграція з системами керування версіями: інтегроване середовище розробки PyCharm взаємодіє з різними системами керування версіями, такими як Git, SVN [15], [16].

Загалом, PyCharm створений з метою спрощення та оптимізації процесу розробки програм на мові Python, надаючи розробникам широкий спектр засобів для ефективної та продуктивної роботи [15], [16].

Суттєвою перевагою інтегрованого середовища розробки PyCharm є його потужність та спрямованість на підтримку розробки програмного забезпечення на мові програмування Python. PyCharm надає широкий спектр інструментів для зручного редагування коду, відлагодження, тестування та інших задач, пов'язаних з розробкою програм. Також слід зазначити, що активна спільнота розробників постійно розвиває та підтримує це середовище, що робить його найпопулярнішим інструментом для роботи з мовою Python [15], [16].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для сегментації зображень наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для сегментації зображень

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Microsoft Visual Studio	Sublime Text
Мови програмування	Python	Багато мов, включаючи C++, C#, Python, Java, та інші	Багато мов, включаючи Python та інші
Зручність розробки на мові Python	+	+—	+—
Безкоштовність	+	+—	+—
Наявність розширень для обробки зображень	+	+	—
Візуальний редактор	+	+	—

Таким чином, для створення програмного забезпечення для сегментації зображень обрано середовище розробки PyCharm, що підтримує розробку на

мові Python, є безкоштовним та дозволяє розробляти різні типи програмного забезпечення, зокрема, програми для обробки зображень.

2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для сегментації зображень

З метою можливості розробки програмного забезпечення для сегментації зображень створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для сегментації зображень враховуємо те, що користувачі програми мають можливості завантаження цифрового зображення, можливість роботи з різними методами сегментації зображень, можливість зміни параметрів методу сегментації зображень при необхідності, можливість збереження зображення після його сегментації, можливість порівняння результатів сегментації за різними критеріями та ін.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для сегментації зображень можна описати таким чином:

- користувач програмного забезпечення для сегментації зображень запускає програму;
- після запуску програми сегментації зображень користувачу відображається головна форма;
- користувач програми сегментації зображень бачить головну форму та завантажує зображення для сегментації;
- користувач обирає алгоритм сегментації зображення та за необхідності змінює його налаштування;
- користувач здійснює сегментацію зображення шляхом натискання відповідної кнопки у програмі;
- користувач завершує роботу з програмним забезпеченням для сегментації зображень.

Враховуючи сценарії роботи користувачів при роботі з програмним забезпеченням для сегментації зображень, модель взаємодії користувача з програмним забезпеченням для сегментації зображень подамо у вигляді діаграми, зображеної на рис. 2.1.

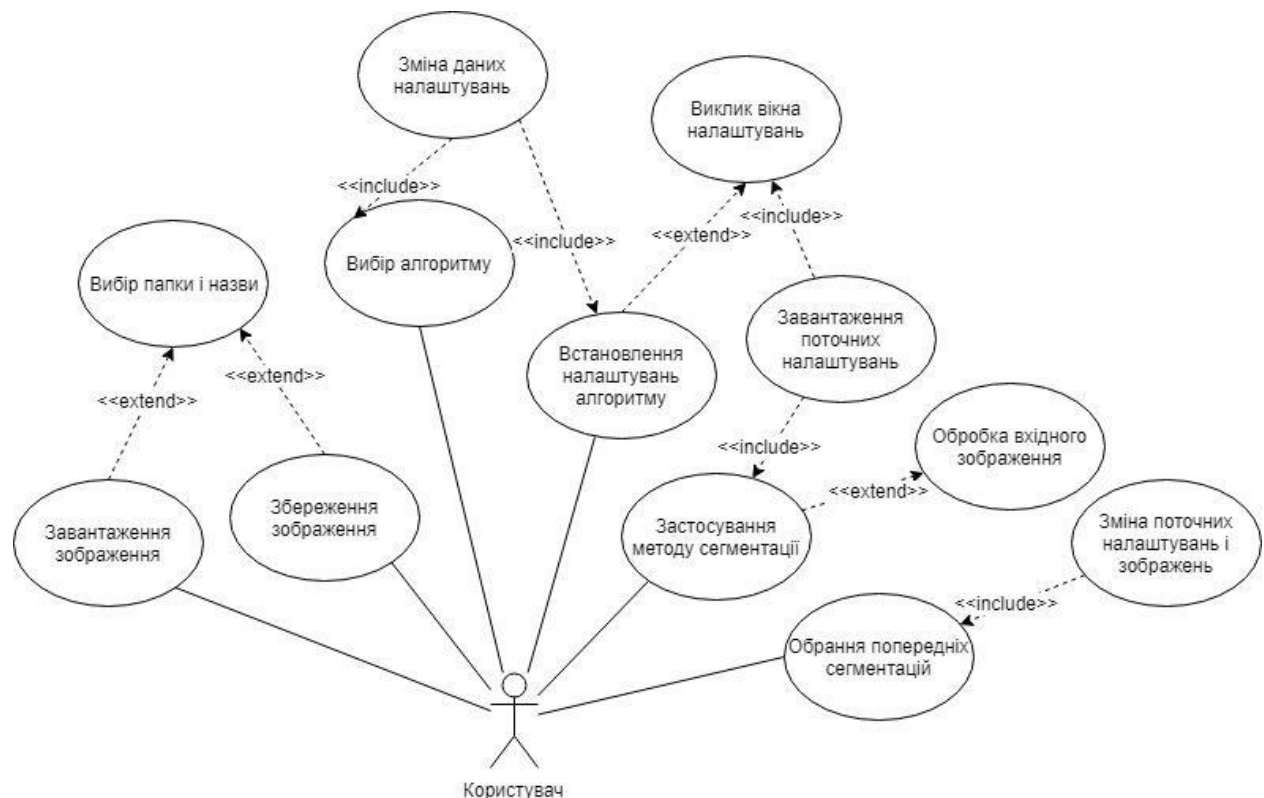


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для сегментації зображень

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для сегментації зображень, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для сегментації зображень.

2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для сегментації зображень обрано мову програмування Python, яка є універсальною інтерпретованою

мовою програмування, що характеризується зрозумілим і читабельним синтаксисом, є легкою у використанні, має можливості розширення, дозволяє розробляти різні види програмних застосунків, набула широкого використання в обробці та аналізі даних, зокрема в обробці зображень [12]-[14].

Для створення програмного забезпечення для сегментації зображень обрано середовище розробки PyCharm, що підтримує розробку на мові Python, є безкоштовним та дозволяє розробляти різні типи програмного забезпечення, зокрема, програми для обробки зображень.

Створено модель взаємодії користувача з програмним забезпеченням для сегментації зображень, яка дозволяє розроблювати програмне забезпечення для сегментації зображень.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для сегментації зображень

Структуру програмного забезпечення для сегментації зображень наведено на рис. 3.1.



Рисунок 3.1 – Структура програмного забезпечення для сегментації зображень

Як видно з рисунку, програмне забезпечення для сегментації зображень складається з таких компонентів:

- графічний інтерфейс застосунку для сегментації зображень;
- основний модуль підсистеми сегментації зображень;
- підсистема роботи з даними.

Проаналізуємо призначення основних компонентів програмного забезпечення для сегментації зображень. Модуль графічного інтерфейсу програмного застосунку включає класи та функції, які забезпечують відображення графіки та взаємодію з користувачем через графічний інтерфейс. До цієї групи входять елементи, такі як вікна, кнопки, мітки і інші стандартні компоненти графічного інтерфейсу.

Підсистема сегментації зображень програмного забезпечення для сегментації зображень включає функції, які при їх виконанні виділяють сегментоване зображення. Ця підсистема реалізовує у програмному забезпеченні для сегментації зображень операції, пов'язані з інтелектуальним аналізом даних.

Підсистема роботи з даними у програмному забезпеченні для сегментації зображень включає класи, змінні і функції, що призначені для опрацювання даних. Ця підсистема відповідає за роботу з даними під час роботи програми для сегментації зображень та при взаємодії з зовнішніми файлами, такими як завантаження і збереження зображень.

Розроблюване програмне забезпечення для сегментації зображень складається з наступних модулів: `main`, `MainWindow`, `SettingsWindow`, `showimage`, `algorithm.kmeans`, `algorithm.thresholding`, `algorithm.watershed`.

Модуль `main` являється вхідною точкою для роботи ПЗ. Тут відбувається ініціалізація і створення головного вікна програми.

Модуль `MainWindow` містить клас `MainWindow`, що являє собою головне вікно програми. Також в класі зберігаються дані, що необхідні під час функціонування програми. Більшість методів класу реалізує взаємодію з графічним інтерфейсом.

У класі `MainWindow` реалізовано наступні методи:

- `_init_* ()` – група методів для ініціалізації графічного інтерфейсу;
- `change_settings ()` – функція, що викликається під час натиснення клавіші налаштувань;
- `_on_select_algorithm ()` – функція, що викликається під час зміни обраного алгоритму в графічному інтерфейсі;
- `_on_select_listmenu ()` – функція, що викликається під час зміни обраного елементу в списку попередніх сегментацій;
- `_on_delete_listmenu ()` – функція, що викликається під час видалення обраного елементу в списку попередніх сегментацій;

- `_set_image ()` – функція, що встановлює отримані зображення в елементи графічного інтерфейсу;
- `_save_image ()` – функція, що викликається при натисненні клавіші збереження зображення;
- `_load_image ()` – функція, що викликається при натисненні клавіші збереження зображення;
- `_enter ()` – функція, що викликається при натисненні клавіші запуску обраного алгоритму;
- `_show_matplotlib ()` – функція, що викликається при натисненні клавіші відображення за допомогою бібліотеки `matplotlib`;
- `_show_open_cv()` – функція, що викликається при натисненні клавіші відображення за допомогою бібліотеки `OpenCV`.

Модуль `SettingsWindow` містить клас `SettingsWindow`, що відповідає за загальний вид та структуру вікна налаштувань. Також клас містить функції для кожного з наявних методів сегментації зображень. Клас реалізує наступні функції:

- `on_clickOK ()` – функція, що викликається під час натиснення клавіші `OK`. Призводить до завершення роботи вікна програми. Встановлює поле з бінарним значенням, що відповідає за успіх в 1;
- `on_clickCancel ()` – функція, що викликається під час натиснення клавіші `Cancel`. Призводить до завершення роботи вікна програми.

Функції `kmean_window`, `thresholding_window`, `adaptive_window` та `watershed_window` використовуються при створення вікна налаштувань для відповідних методів. Вони використовують об'єкт класу `SettingsWindow` в якості основи для вікна і додають графічний інтерфейс для налаштування відповідних методів.

Модуль `showimage` складається з двох функцій, що відповідають за виведення зображень в окремих вікнах за допомогою сторонніх бібліотек:

- `show_images_mpl ()` – функція групує два зображення(вхідне і вихідне) в одному вікні і виводить їх за допомогою бібліотеки `matplotlib` в графічному інтерфейсі, що дозволяє більш детально переглядати зображення.
- `show_cv ()` – функція виводить два зображення у повний розмір в окремих вікнах за допомогою бібліотеки `OpenCV`.

Модуль `algorithm.kmeans` має у своєму складі одну функцію, що відповідає за обробку зображення за допомогою методу `k-means`. У якості вхідних даних функція отримує вхідне зображення і параметри налаштування.

Модуль `algorithm.thresholding` має у своєму складі дві функції, що відповідають за обробку зображення за допомогою методу порогового значення. У якості вхідних даних функції отримують вхідне зображення і параметри налаштування. Серед них наступні функції:

- `thresholding ()` – функція призначена для сегментування зображення на основі глобального порогового значення;
- `thresholding_adaptive ()` – функція призначена для сегментування зображення на основі методу порогового значення, але значення порогу обчислюється для невеликих областей зображення окремо.

Модуль `algorithm.watershed` має у своєму складі одну функцію, що відповідає за обробку зображення за допомогою методу `watershed`. У якості вхідних даних функція отримує вхідне зображення і параметри налаштування.

3.2 Функціонування програмного забезпечення для сегментації зображень

Схема функціонування програмного забезпечення для сегментації зображень наведена на рис. 3.2. Алгоритм функціонування програмного забезпечення для сегментації зображень подано у вигляді графічного подання логічної структури алгоритму, де кожен етап функціонування програмного забезпечення для сегментації зображень зображається у вигляді геометричних блоків, які мають чітко визначене функціональне призначення.

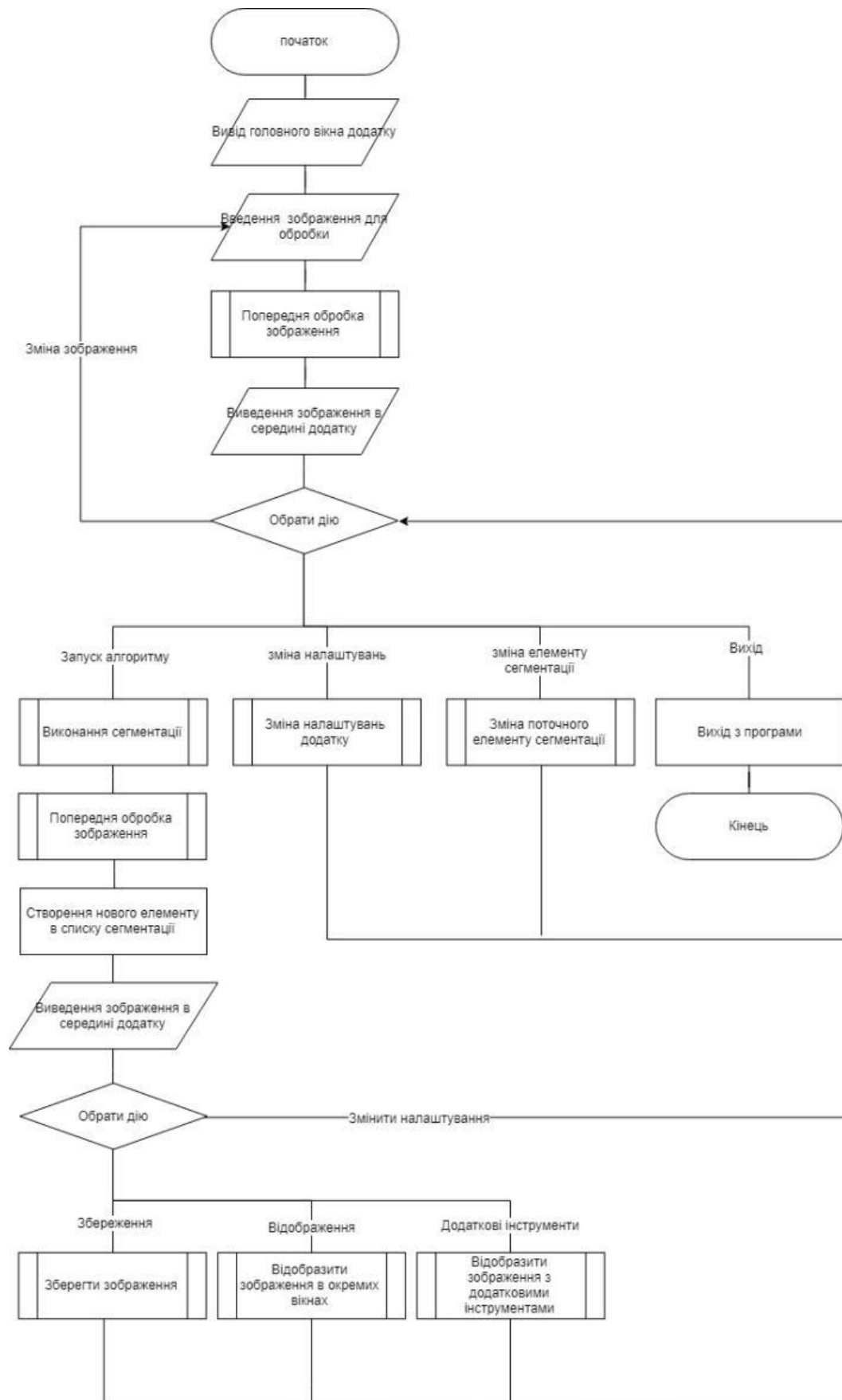


Рисунок 3.2 – Схема функціонування програмного забезпечення для сегментації зображень

Як видно з рисунку 3.2, після запуску програми користувач вводить зображення для обробки (сегментації), потім виконується попередня обробка зображення. Після цього користувач обирає дію (виконання сегментації, зміна налаштувань, зміна поточного зображення або вихід з програми). Далі відбувається пошук об'єктів на зображеннях (сегментація) та їх виділення і відображення. У користувача є можливість зберегти сегментоване зображення, вивести його в окремих вікнах та ін.

3.3 Проєктування програмного забезпечення для сегментації зображень

Загальна діаграма класів програмного забезпечення для сегментації зображень наведена на рис. 3.3.

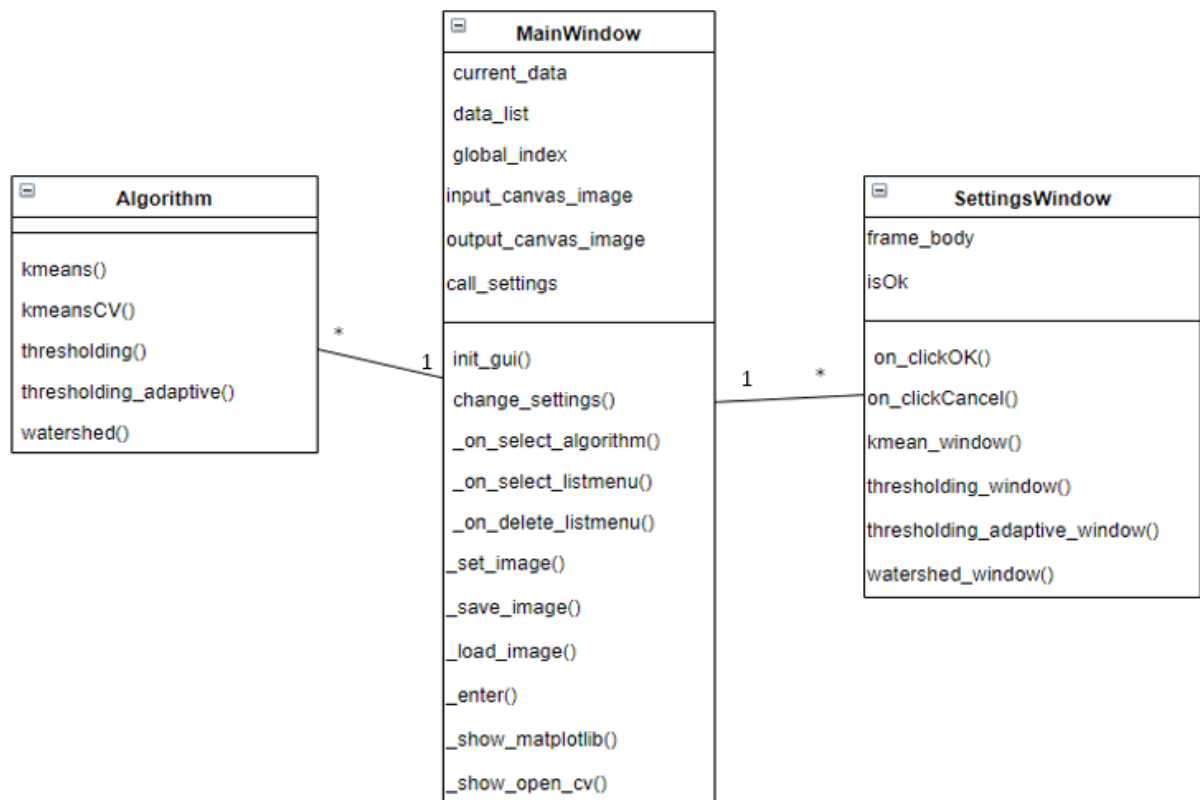


Рисунок 3.3 – Діаграма класів програмного забезпечення для сегментації зображень

В даній діаграмі на рис. 3.3 класи MainWindow і SettingsWindow представляють класи графічного інтерфейсу, а клас Algorithm функціональні можливості для сегментації зображень даного програмного забезпечення.

Діаграма компонентів програмного забезпечення для сегментації зображень наведена на рисунку 3.4. Дана діаграма описує особливості фізичної уяви системи.

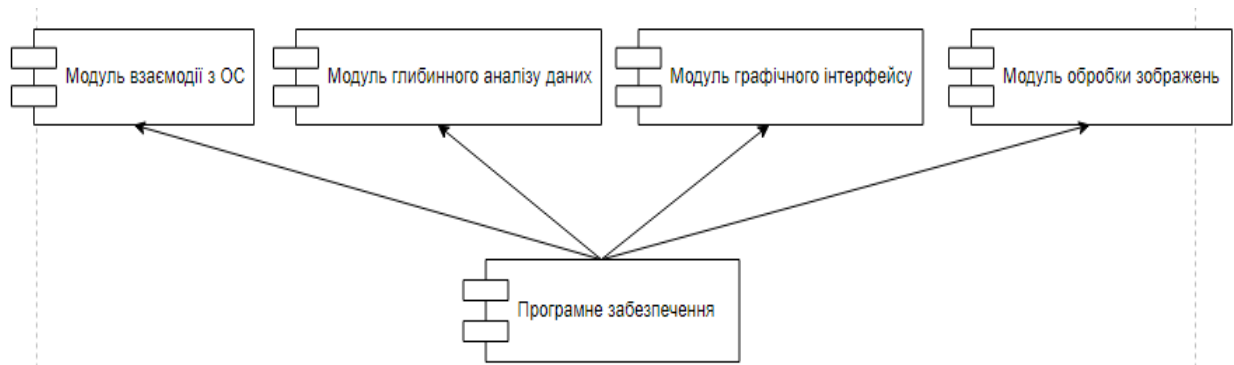


Рисунок 3.4 – Діаграма компонентів програмного забезпечення для сегментації зображень

Модуль взаємодії програмного забезпечення для сегментації зображень з ОС включає в себе функції і бібліотеки для завантаження і збереження зображень на комп'ютері.

В модулі глибинного аналізу даних програмного забезпечення для сегментації зображень скомпоновані функції, що були розроблені для сегментації зображень, а також бібліотеки якими при цьому були використані.

Модуль графічного інтерфейсу програмного забезпечення для сегментації зображень включає в себе бібліотеку tkinter для розробки графічного інтерфейсу, а також створені на її основі класи, що представляють елементи графічного інтерфейсу .

Модуль обробки зображень програмного забезпечення для сегментації зображень включає в себе бібліотеки та функції, використані при попередній обробці зображень та виконанні найпростіших перетворень.

3.4 Проектування інтерфейсу програмного забезпечення для сегментації зображень

Проектування інтерфейсу взаємодії користувача з програмним забезпеченням для сегментації зображень виконано з урахуванням вимог технічного завдання.

Приклади розроблених інтерфейсних форм наведено на рис. 3.5-3.8.

Форму головного вікна програмного забезпечення для сегментації зображень наведено на рис. 3.5.

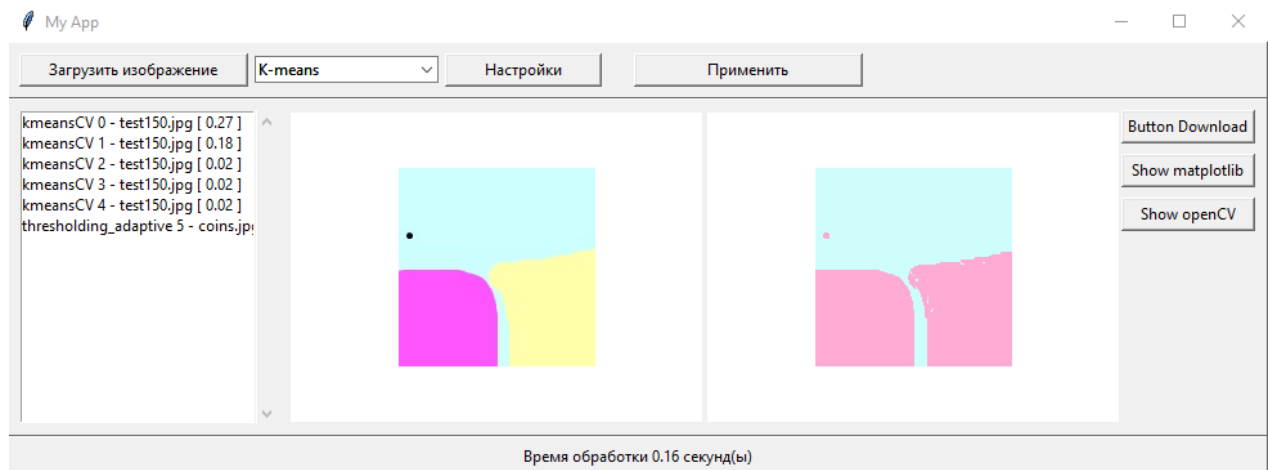


Рисунок 3.5 – Форма головної сторінки програмного забезпечення для сегментації зображень

Форму діалогового вікна вибору зображень відображено на рис. 3.6.

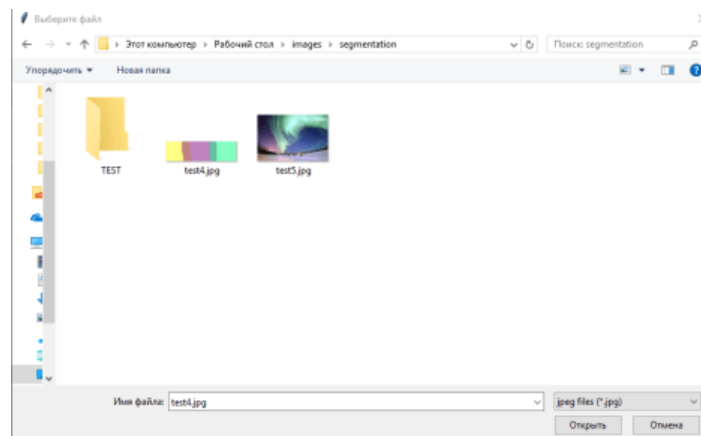


Рисунок 3.6 – Форма діалогового вікна вибору зображень

Форму вікна з вибором параметрів методу сегментації зображень наведено на рис. 3.7.

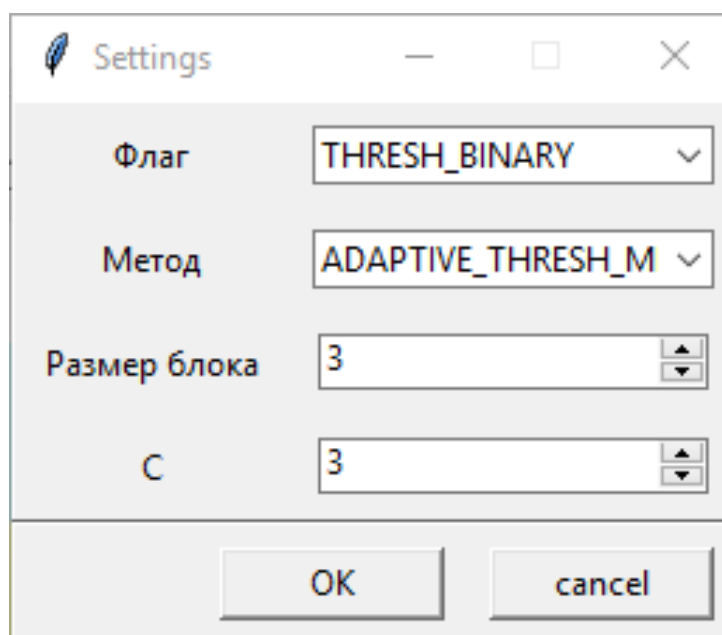


Рисунок 3.7 – Форма вікна з вибором параметрів методу сегментації зображень

Інтерфейс форми перегляду базового та сегментованого зображень відображено на рис. 3.8.

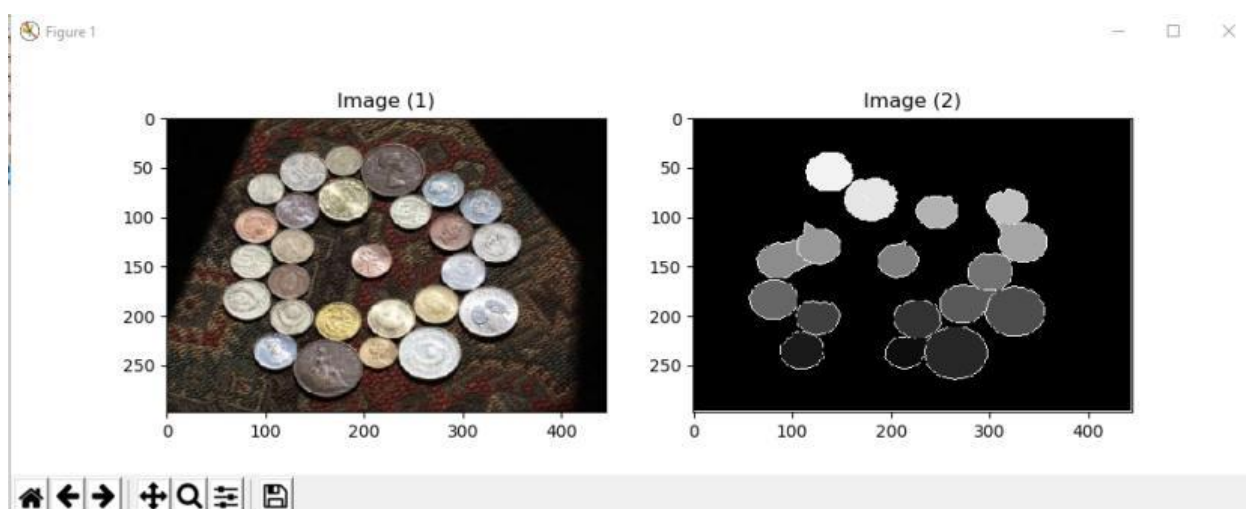


Рисунок 3.8 – Інтерфейс форми перегляду базового та сегментованого зображень

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для сегментації зображень. Запропоновано структуру та описано основні модулі програмного забезпечення для сегментації зображень.

Описано функціонування програмного забезпечення для сегментації зображень. Описано особливості реалізації програмного забезпечення для сегментації зображень. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для сегментації зображень.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для сегментації зображень. Програма написана на мові Python, яка є універсальною інтерпретованою мовою програмування, що характеризується зрозумілим і читабельним синтаксисом, є легкою у використанні, має можливості розширення, дозволяє розробляти різні види програмних застосунків, набула широкого використання в обробці та аналізі даних, зокрема в обробці зображень. В якості середовища розробки для створення програми обрано PyCharm, що підтримує розробку на мові Python, є безкоштовним та дозволяє розробляти різні типи програмного забезпечення, зокрема, програми для обробки зображень.

Програмне забезпечення для сегментації зображень забезпечує виконання таких функцій:

- можливість завантаження цифрового зображення;
- можливість роботи з різними методами сегментації зображень;
- можливість зміни параметрів методу сегментації зображень при необхідності;
- можливість збереження зображення після його сегментації.
- можливість порівняння результатів сегментації за різними критеріями.

Для експлуатації програмного забезпечення для сегментації зображень необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 1,5 GHz і більше;
- вільне місце на диску 100 Мб або більше;
- оперативна пам'ять об'ємом 2Гб і більше;
- встановлена операційна система;
- інтерпретатор мови програмування Python версії, не нижче за 3.

Функціональні характеристики розробленого програмного забезпечення для сегментації зображень наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для сегментації зображень наведено у додатку Б.

4.2 Характеристики програми для сегментації зображень

Програмне забезпечення для сегментації зображень може працювати безперервно, в залежності від потреб користувача.

Програмне забезпечення для сегментації зображень може використовуватись в будь-якій операційній системі, де встановлено інтерпретатор Python версії не нижче третьої.

Для коректного функціонування програми для сегментації зображень вміст директорії з програмним забезпеченням не можна змінювати.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для сегментації зображень необхідно виконати python скрипт `mainsegm.pyw`, що знаходиться в кореневій директорії програми.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для сегментації зображень є зображення, що обираються користувачем для сегментації, а також метод сегментації та його налаштування, що вказуються до початку процесу сегментації зображення.

Вихідними даними програмного забезпечення для сегментації зображень є результат сегментації зображення, а також додаткова інформація представлена в інтерфейсі програми.

4.3.3 Повідомлення

Користувач програмного забезпечення для сегментації зображень може отримати такі повідомлення: повідомлення про некоректно введений файл для сегментації зображення (файл не є зображенням).

4.4 Виконання програмного забезпечення для сегментації зображень

Після запуску програмного забезпечення для сегментації зображень користувачу виводиться головна форма програми (рис. 4.1).

Форму головного вікна програмного забезпечення для сегментації зображень наведено на рис. 4.1. Як видно, користувач має можливість завантажити зображення, обрати метод сегментації зображення та налаштувати його параметри, а також застосувати відповідний метод до обраного зображення.

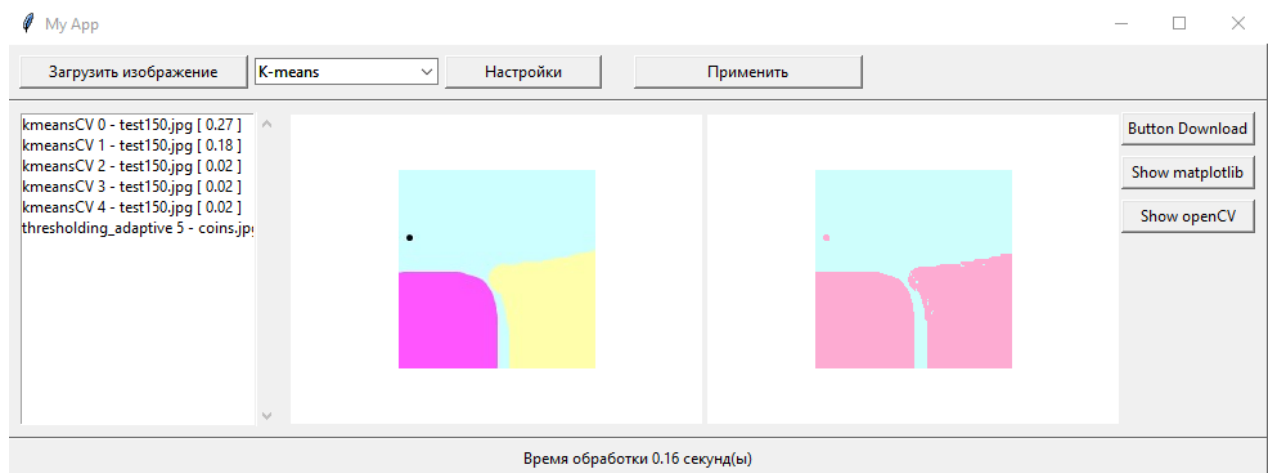


Рисунок 4.1 – Форма головного вікна програмного забезпечення для сегментації зображень

Нижня панель головного вікна носить інформативний характер і використовується для відображення інформації про поточний статус сегментації, та відображення додаткової інформації після її завершення.

При натисканні на відповідну кнопку головної форми програмного забезпечення для сегментації зображень з'являється форма діалогового вікна вибору зображень (рис. 4.2.). Тут користувач може обрати файл із зображенням, сегментацію якого необхідно виконати.

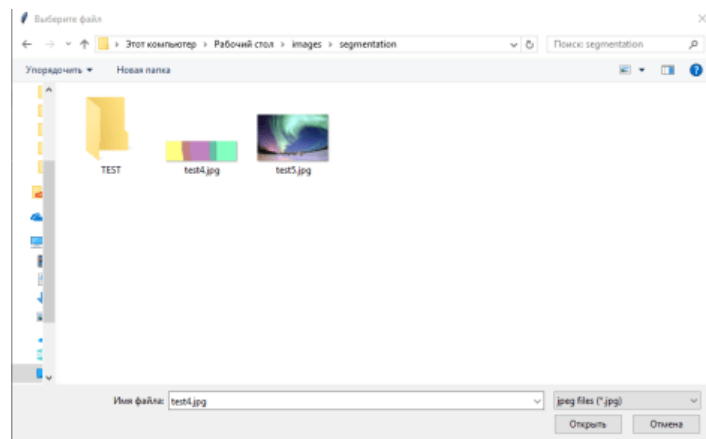


Рисунок 4.2 – Форма діалогового вікна вибору зображень

При натисканні на головній формі на кнопку налаштувань з'являється форма вікна з вибором параметрів методу сегментації зображень (рис. 4.3).

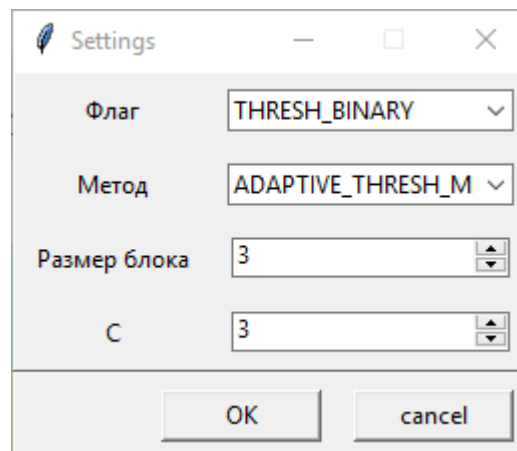


Рисунок 4.3 – Форма вікна з вибором параметрів методу сегментації зображень

Після виконання сегментації зображення користувачу виводиться форма перегляду базового та сегментованого зображень (рис. 4.4). Тут за допомогою відповідних засобів користувач може зсунути зображення, наблизити його або зберегти в окремому файлі.

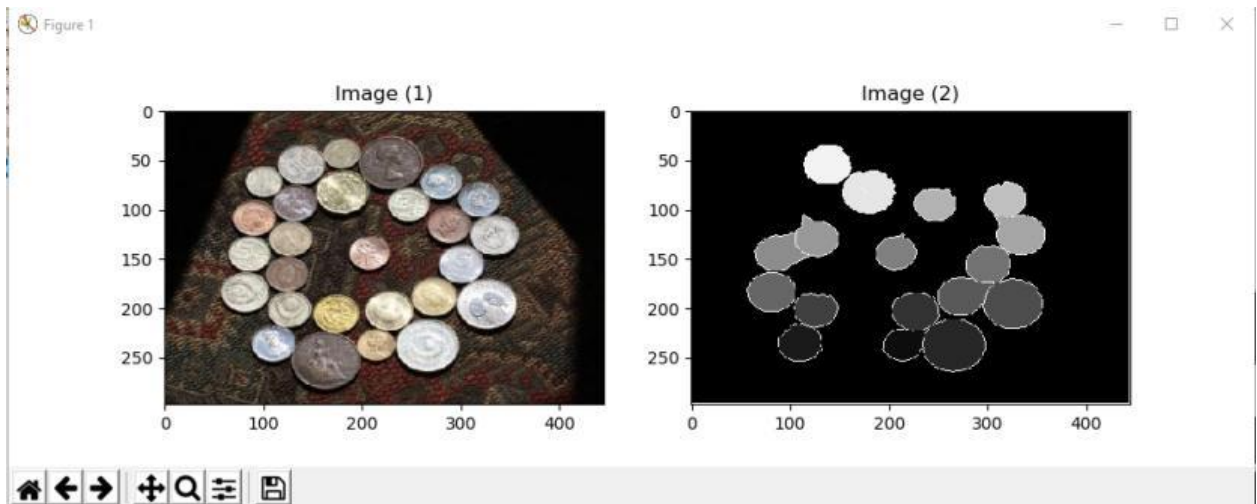


Рисунок 4.4 – Інтерфейс форми перегляду базового та сегментованого зображень

Вікно для збереження файлу (рис. 4.5) надається графічною бібліотекою tkinter і характеризується аналогічними перевагами вікна завантаження.

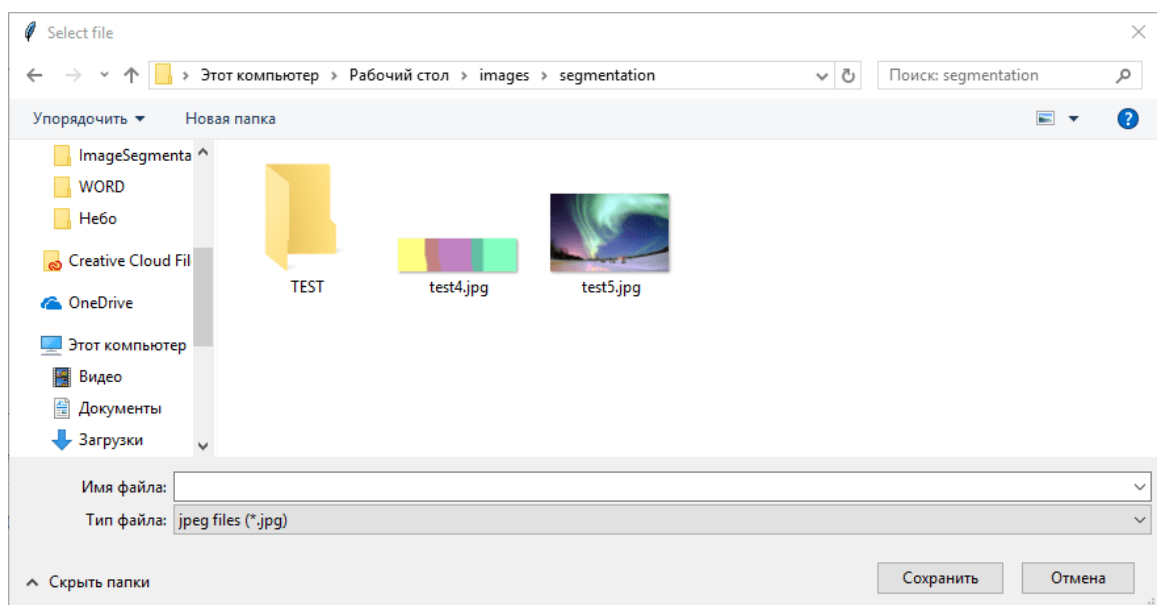


Рисунок 4.5 – Форма діалогового вікна збереження зображень

4.5 Тестування програмного забезпечення для сегментації зображень

Виконано тестування програмного забезпечення для сегментації зображень.

Виявлено, що програма для сегментації зображень функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу – сегментацію зображень за допомогою різних методів сегментації.

Приклад успішної сегментації зображень за допомогою даного програмного забезпечення наведено на рисунках 4.6 – 4.9.

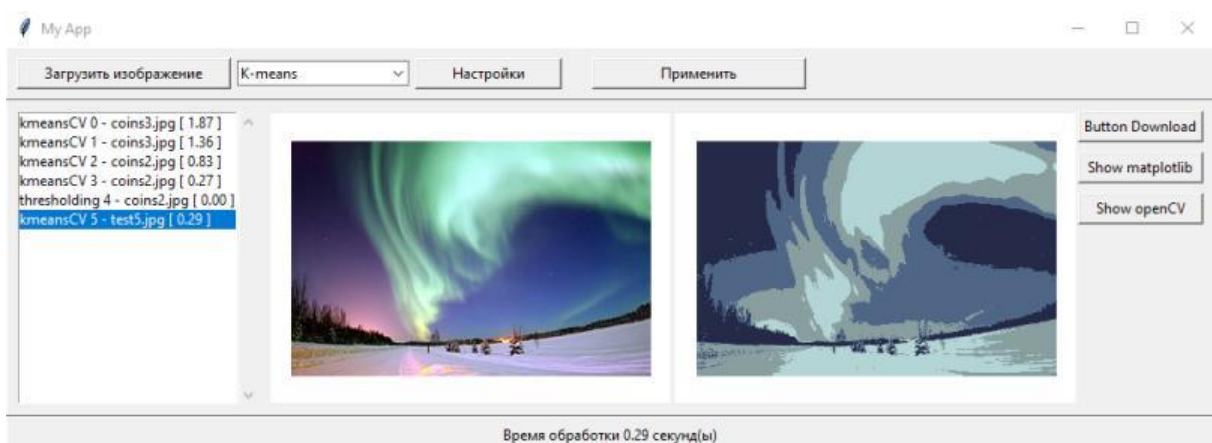


Рисунок 4.6 – Приклад сегментації за допомогою метода k-means

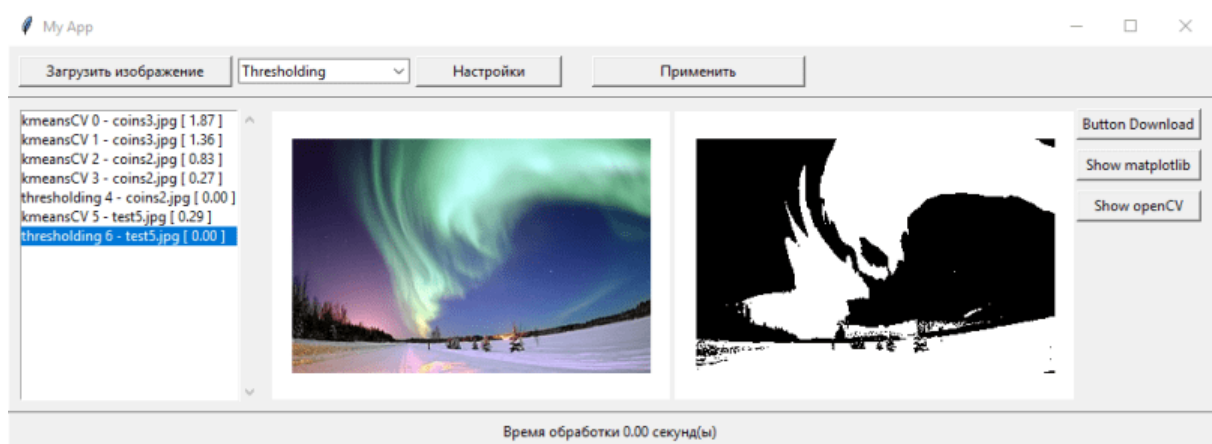


Рисунок 4.7 – Приклад сегментації за допомогою метода бінаризації

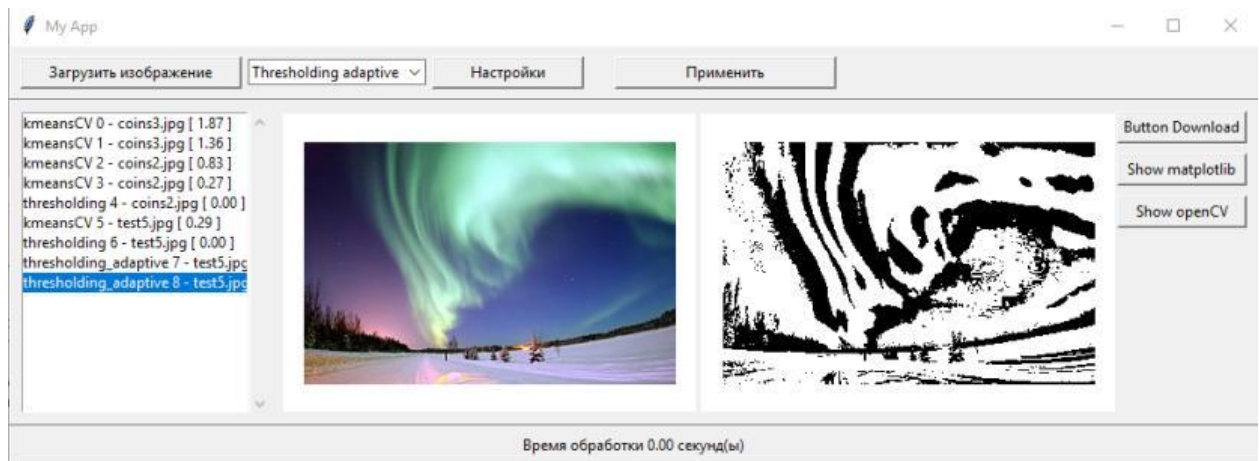


Рисунок 4.8 – Приклад сегментації за допомогою метода адаптивної бінаризації

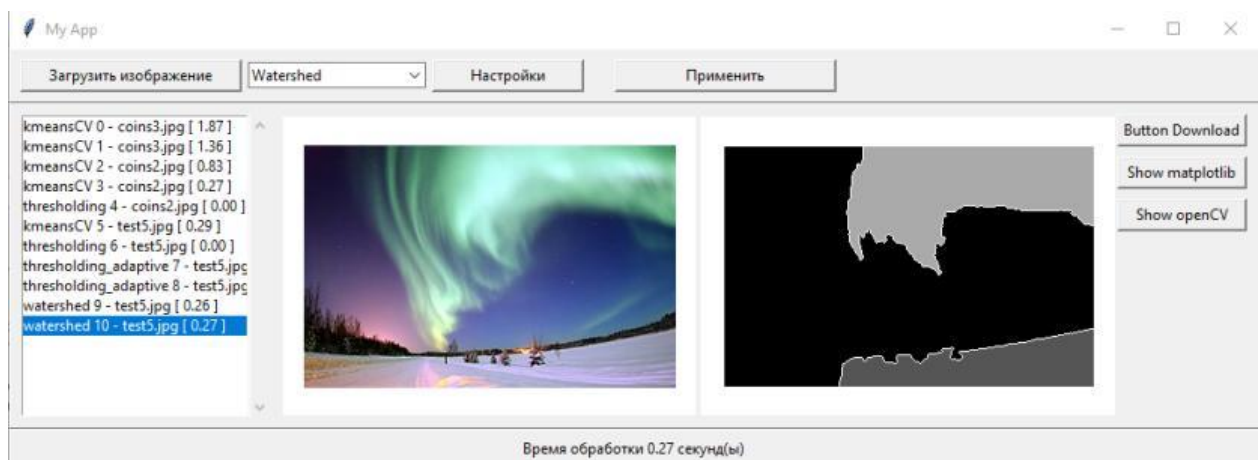


Рисунок 4.9 – Приклад сегментації за допомогою метода watershed

Розроблене програмне забезпечення дозволяє розв’язувати завдання сегментації зображень за допомогою різних методів обробки зображень.

4.6 Висновки за розділом 4

Описано програмне забезпечення для сегментації зображень. Виконано тестування розробленого програмного забезпечення для сегментації зображень. Результати тестування програмного забезпечення для сегментації зображень показали, що розроблена програма дозволяє розв’язувати завдання виділення об’єктів на зображеннях за допомогою різних методів сегментації.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для обробки зображень.

Визначено, що сегментація зображень представляє собою потужний інструмент, який дозволяє виділяти та ізолювати об'єкти або області на зображенні, та має широке застосування в медичній візуалізації, робототехніці та області комп'ютерного зору. Досліджено різноманітні підходи до сегментації зображень, включаючи традиційні методи, такі як порогове значення, регіональна сегментація, сегментація на основі контурів та кластеризація, а також методи машинного навчання, глибокі нейронні мережі. Досліджено методи та моделі використання нейронних мереж для сегментації зображень. Розглянуто метрики оцінки та набори даних, які використовуються для оцінювання ефективності алгоритмів сегментації.

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для сегментації зображень. Проте деякі програми для сегментації зображень можуть бути важкими для налаштування, особливо для користувачів без попереднього досвіду в галузі комп'ютерного бачення. Деяке програмне забезпечення для точної сегментації зображень може вимагати потужних обчислювальних ресурсів, що може ускладнити їх використання на менш потужних пристроях. Оскільки різні об'єкти мають різні форми та контексти, не існує універсального підходу для сегментації, і відповідне програмне забезпечення може виявлятися менш ефективними для певних типів об'єктів. Тому актуальною є розробка програмного забезпечення для сегментації зображень.

Сформульовано функціональні вимоги до програмного забезпечення для сегментації зображень.

Для реалізації програмного забезпечення для сегментації зображень обрано мову програмування Python, яка є універсальною інтерпретованою мовою програмування, що характеризується зрозумілим і читабельним синтаксисом, є легкою у використанні, має можливості розширення, дозволяє розробляти різні види програмних застосунків, набула широкого використання в обробці та аналізі даних, зокрема в обробці зображень [12]-[14].

Для створення програмного забезпечення для сегментації зображень обрано середовище розробки PyCharm, що підтримує розробку на мові Python, є безкоштовним та дозволяє розробляти різні типи програмного забезпечення, зокрема, програми для обробки зображень.

Новизна роботи полягає в тому, що запропоновано модель взаємодії користувача з програмним забезпеченням для сегментації зображень, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для сегментації зображень.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для сегментації зображень.

Розроблено програмне забезпечення для сегментації зображень. Запропоновано структуру та описано основні модулі програмного забезпечення для сегментації зображень.

Описано функціонування програмного забезпечення для сегментації зображень. Описано особливості реалізації програмного забезпечення для сегментації зображень. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для сегментації зображень.

Виконано тестування розробленого програмного забезпечення для сегментації зображень. Результати тестування програмного забезпечення для сегментації зображень показали, що розроблена програма дозволяє розв'язувати завдання виділення об'єктів на зображеннях за допомогою різних методів сегментації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Image Segmentation: The Basics and 5 Key Techniques [Electronic resource]. – Access mode : <https://datagen.tech/guides/image-annotation/image-segmentation/>.
2. Computer Vision Tutorial: A Step-by-Step Introduction to Image Segmentation Techniques [Electronic resource]. – Access mode : <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/>.
3. Image Segmentation [Electronic resource]. – Access mode : <https://huggingface.co/tasks/image-segmentation>.
4. Image Segmentation [Electronic resource]. – Access mode : <https://www.sciencedirect.com/topics/computer-science/image-segmentation>.
5. Image segmentation detailed overview [Electronic resource]. – Access mode : <https://www.superannotate.com/blog/image-segmentation-for-machine-learning>.
6. Image Segmentation Explained [Electronic resource]. – Access mode : <https://builtin.com/machine-learning/image-segmentation>.
7. Guide to Image Segmentation in Computer Vision: Best Practices [Electronic resource]. – Access mode : <https://encord.com/blog/image-segmentation-for-computer-vision-best-practice-guide/>.
8. Image Segmentation Software [Electronic resource]. – Access mode : [https://www.capterra.com/search/product/?search=image segmentation](https://www.capterra.com/search/product/?search=image%20segmentation).
9. Sefexa Image Segmentation Tool [Electronic resource]. – Access mode : <http://www.fexovi.com/sefexa.html>.
10. Getting Started with Image Segmenter [Electronic resource]. – Access mode : <https://www.mathworks.com/help/images/image-segmentation-using-the-image-segmenter-app.html>.
11. CVIPtools [Electronic resource]. – Access mode : <https://cviptools.ece.siue.edu/>.

12. Python. Overview [Electronic resource]. – Access mode :
https://www.tutorialspoint.com/python/python_overview.htm.

13. The Python Tutorial [Electronic resource]. – Access mode :
<https://docs.python.org/3/tutorial/index.html>.

14. Introduction To Python [Electronic resource]. – Access mode :
<https://www.geeksforgeeks.org/introduction-to-python/>.

15. PyCharm: The Python IDE for Professional Developers [Electronic resource]. – Access mode : <https://www.jetbrains.com/pycharm/>.

16. Pycharm. Introduction [Electronic resource]. – Access mode :
https://www.tutorialspoint.com/pycharm/pycharm_introduction.htm.

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для сегментації зображень.

А.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів сегментації зображень», затверджене наказом Національного університету «Запорізька політехніка» № 437 від 14 листопада 2023 р.

А.2 Призначення розробки

Програмний продукт призначений для забезпечення та підтримки процесу обробки зображень шляхом їх сегментації.

А.3 Основні вимоги до програми, що розробляється

А.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для сегментації зображень забезпечує виконання таких функцій:

- можливість завантаження цифрового зображення;
- можливість роботи з різними методами сегментації зображень;
- можливість зміни параметрів методу сегментації зображень при необхідності;
- можливість збереження зображення після його сегментації.
- можливість порівняння результатів сегментації за різними критеріями.

A.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для сегментації зображень повинен бути зручним для користувачів.

A.3.3 Вимоги до надійності

Програмне забезпечення для сегментації зображень повинно забезпечити надійне функціонування.

A.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для сегментації зображень необхідна наявність персонального комп'ютера у користувача.

A.3.5 Вимоги до складу та параметрів технічний засобів

Для експлуатації програмного забезпечення для сегментації зображень необхідно таке апаратне та програмне забезпечення:

- процесор із тактовою частотою 1,5 GHz і більше;
- вільне місце на диску 100 Мб або більше;
- оперативна пам'ять об'ємом 2Гб і більше;
- встановлена операційна система;
- інтерпретатор мови програмування Python версії, не нижче за 3.

A.3.6 Вимоги до маркування і пакування

Програма для сегментації зображень може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для сегментації зображень».

A.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для сегментації зображень є зображення, що обираються користувачем для сегментації, а також метод сегментації та його налаштування, що вказуються до початку процесу сегментації зображення.

Вихідними даними програмного забезпечення для сегментації зображень є результат сегментації зображення, а також додаткова інформація представлена в інтерфейсі програми.

ДОДАТОК Б
Фрагмент тексту програми

```

class MainWindow(Frm):
    options = {
        "K-seredns": [0, kserednsCV, kseredn_cv_window],
        "Thrshldng": [1, Thrshldng, Thrshldng_window],
        "Thrshldng modif": [2, Thrshldng_modif, Thrshldng_modif_window],
        "Vodned": [3, vodned, vodned_window]
    }

    # [vhdzobr, methd, prms, vyhzobr]
    crntzbr = {"vhdzobr":None, "nmmzobr":None, "methdorithm":None,
    "prms":{ }, 'time':0, "vyhzobr":None}
    # group crntzbr elements
    spyskdht = []
    #
    glbidx = -1
    inpcnvzobr, vyhcnvzobr = None, None
    clprms = None
    def __init__(self, haz):
        Frm.__init__(self, haz)
        self.haz = haz
        self.haz.title("Segmen")
        self.haz.updatidletasks()
        self.haz.aftridle(lambda:
self.haz.minsize(self.haz.winfwidthth(), self.haz.winfheight()))
    def
    _init_header(self):
        self.frmHeader = Frm(self, bordrwidthth=1, relief=RAISED, padx=8,
pady=8)
        self.frmHeader.pack(fill=X)
        self.btnOpenZobr =
Btn(self.frmHeader, text="Завантаж зображення", widthth=24,
cmdnd=self._load_zobr)
        self.btnChangeMethdorithm =
Combobox(self.frmHeader, stt="readonly", values=[i for i in
options.keys()])
        self.btnChangeMethdorithm.current(0)
        self.btnChangeMethdorithm.bind(self._on_select_methdorithm)
        self.btnChangeMethdorithm.event_generate()

```

```

        self.btnPrmsMethd = Btn(self.frmHeader, text="Налаштування",
widthth=16, cmnd=self.change_prms)
        self.btnEnter = Btn(self.frmHeader, text="Застосувати",
widthth=24, cmnd=self._enter)
        self.btnOpenZobr.pack(side=RIGHT)
        self.btnChangeMethdorithm.pack(side=RIGHT)
        self.btnPrmsMethd.pack(side=RIGHT)
        self.btnEnter.pack(side=RIGHT)
    def _init_body(self):
        self.frmBody = Frm(self, bordrwidthth=1, relief=RAISED)
        self.frmBody.pack(side=TOP, expand=True)
        self.frmLeftMenu = Frm(self.frmBody, height=10)
        self.frmLeftMenu.pack(side=RIGHT, fill=Y, padx=[0, 8])
    self.cnvInputZobr = Cnv(self.frmBody, widthth=640, height=480)
        self.cnvInputZobr.bind("<2>", funcdcnv)
        self.cnvInputZobr.pack(side=RIGHT, expand=True)
    self.cnvOutputZobr = Cnv(self.frmBody, widthth=640, height=480,
background='#FFF')
        self.cnvOutputZobr.bind("<1>", funcdcnv)
        self.cnvOutputZobr.pack(side=RIGHT, fill=BOTH)
        self.frmRightMenu = Frm(self.frmBody, relief=RAISED)
        self.frmRightMenu.pack(side=RIGHT, fill=Y)
        self.btnDwnld = Btn(self.frmBody, cmnd=self._save_zobr)
        self.btnDwnld .pack(pady=[0, 8])
        self.btnShwMPL = Btn(self.frmBody, cmnd=self._show_matplotlib)
        self.btnShwMPL.pack(fill=X, pady=[0, 8])
        self.btnShowCV = Btn(self.frmBody, text='Show openCV',
cmnd=self._show_open_cv)
        self.btnShowCV.pack(fill=X)
        sclrbr = Scrlbr(self.frmLeftMenu)
        sclrbr.pack(side=RIGHT, fill=Y)
        self.lstmn = Listbox(self.frmLeftMenu, widthth=40,
yscrollcmnd=sclrbr.set)
        sclrbr.config(cmnd=self.lstmn.yview)

```

```

        #self.lstmn.insert(-1, "mst")
        self.lstmn.bind('<<ListboxSelect>>', self._on_select_lstmn)
        self.lstmn.bind('<Delete>', self._on_delete_lstmn)
        self.lstmn.pack(fill=Y, expand=True)
    def _init_botm(self):
        self.frmBotm = Frm(self, height=20, bordrwidthth=1,
relief=RAISED, padx=4, pady=4)
        self.frmBotm.pack(fill=X)
        self.lbflsts = Lbfl(self.frmBotm, text="")
        self.lbflsts.pack(fill=X)
        def change_prms(self):
            self.crntzbr['prms'] = self.clprms(**self.crntzbr['prms'])
    def _on_select_methdorithm(self, event):
        w = event.widget
        self.crntzbr["methdorithm"] = self.options[w.get()][1]
        self.clprms = self.options[w.get()][2]
        self.crntzbr['prms'] = {}
    def _on_select_lstmn(self, event):
        try:
            w = event.widget
            idx = int(w.crslect()[0])
            value = w.get(idx)
            self.crntzbr = copy.deepcopy(self.spyskdht[idx])
            self.cnvInputZobr.delete("all")
            self.cnvOutputZobr.delete("all")
            self._set_zobr(self.cnvInputZobr, self.crntzbr["vhdzobr"],
IN)
            self._set_zobr(self.cnvOutputZobr, self.crntzbr["vyhzobr"],
OUT)
            # botm
            self.lbflsts["text"] = "Час обробки {0:.2f}
секунд".format(self.crntzbr['time'])
        except IdxError as error:
            print("IdxError!")
    def _on_delete_lstmn(self, event):
        idx = int(event.widget.crslect()[0])

```

```

self.spyskdht.pop(idx)
for i in self.spyskdht:
    print(i)    def _set_zobr(self, cnv, zobr, shwn):
zobrvys, zobrdl= zobr.shape[0], zobr.shape[1]
mxd1, mxvys = cnv.wininfo_widhth(), cnv.wininfo_height()
local_zobr = zobr.copy()
while zobrdl> mxd1 or zobrvys > mxvys:
    zobrdl//= 2
    zobrvys //= 2
else:
    local_zobr = cv2.resize(zobr, (zobrWidhth, zobrvys))
s = PIL.ZobrTk.PhotoZobr(zobr=PIL.Zobr.fromarray(local_zobr))
    cnv.create_zobr(cnv.wininfo_widhth() / 2, cnv.wininfo_height() / 2,
                    zobr=s, anchor=CNTR)
if shwn == IN:
    self.inpcnvzobr = s
elif shwn == OUT:
    self.vyhcnvzobr = s
return
if shwn == 'in':
    self.inpcnvzobr =
PIL.ZobrTk.PhotoZobr(zobr=PIL.Zobr.fromarray(local_zobr))
    cnv.create_zobr(cnv.wininfo_widhth() / 2, cnv.wininfo_height() /
2,
                    zobr=self.inpcnvzobr, anchor=CNTR)
elif shwn == 'out':
    self.vyhcnvzobr =
PIL.ZobrTk.PhotoZobr(zobr=PIL.Zobr.fromarray(local_zobr))
    cnv.create_zobr(cnv.wininfo_widhth() / 2, cnv.wininfo_height() /
2,
                    zobr=self.vyhcnvzobr)
def _save_zobr(self):
    f = fd.asksaveasfilename(initialdir="/", title="Обрати файл",

```

```

        filetypees=((("Файли типу jpeg", "*.jpg"), ("Всі файли",
        "**.*"))))
        if not f:
            return
        svd = self.crntzbr['vyhzobr']
        if len(self.crntzbr['vyhzobr'].shape) == 3:
            svd = cv2.cvtColor(self.crntzbr['vyhzobr'],
cv2.COLOR_RGB2BGR)
            f = f + '.' + self.crntzbr["nmmzobr"].split('.')[ -1]
            cv2.imwrite(f, svd)
        def _load_zobr(self):
            ptt_zobr = fd.askopenfilename(initialdir=getcwd() +
r"\source\zobrs",
                                     title="Оберіть файл",
                                     filetypees=(((" Файли типу jpeg ",
        "*.jpg"), (" Файли типу png ", "*.png"), ("Файли типу bmp ", "*.bmp"),
        ("Всі файли", "**.*"))))
            if len(ptt_zobr) > 0:
                if ptt_zobr.split('.')[ -1] in ['bmp', 'png', 'jpg']:
                    zobr = cv2.imread(ptt_zobr)
                    self.crntzbr["nmmzobr"] = ptt_zobr.split(r'/' )[ -1]
                    self.crntzbr["vhdzobr"] = cv2.cvtColor(zobr,
cv2.COLOR_BGR2RGB)
                    self._set_zobr(self.cnvInputZobr,
self.crntzbr["vhdzobr"], IN)
                    self.vyhcnvzobr = None
                else:
                    print('неправильний тип файлу')
        def _enter(self):
            self.frmBotm['background'] = "red"
            self.lbflsts["text"] = "Виконання"
            self.update()
            methd = self.crntzbr['methdorithm']
            tm0 = time.time()

```

```

        self.crntzbr["vyhzobr"]      =      methd(self.crntzbr["vhdzobr"],
**self.crntzbr["prms"])
        rslttm = time.time() - tm0
        self._set_zobr(self.cnvOutputZobr, self.crntzbr["vyhzobr"], OUT)
        self.crntzbr['time'] = rslttm
        self.spyskdht.append(copy.deepcopy(self.crntzbr))
        self.glbdx += 1
        self.lstmn.insert(END, self.crntzbr['methdorithm'].__name__ + "
" +
                                str(self.glbdx) + " - " +
                                self.crntzbr["nmmzobr"] +
                                ' [ ' + str("{0:.2f}".format(rslttm)) + '
]')

        self.lstmn.selection_clear(0, END)
        self.lstmn.selection_set("кінець")
        self.frmBotm['background'] = self['background']
        self.lbflsts["text"]      =      "Час      обробки      {0:.2f}
секунд".format(rslttm)      self.update()
        def _show_matplotlib(self):
            show_zobrs_mpl((self.crntzbr["vhdzobr"],
self.crntzbr["vyhzobr"]))
        def _show_open_cv(self):
            show_cv(self.crntzbr["vhdzobr"], self.crntzbr["vyhzobr"])
        x, y, z = zobr.shape
        zobr_2d      =      zobr.reshape(x*y,      z)      kseredns_clstr      =
clstr.KSeredns(init='k-seredns++', n_clstrs=k)
        kseredns_clstr.fit(zobr_2d)
        clstcntr = kseredns_clstr.clstcntr_
        clstlbfl = kseredns_clstr.lbfls_      clstcntr = (clstcntr *
255).astype(numpy.uint8)      vyhzobr = clstcntr[clstlbfl].reshape(x, y,
z)
        vyhzobr = vyhzobr.reshape(x,y,z)      return vyhzobr
def kserednsCV(zobr, **kwargs):
    print(kwargs)

```

```

k = kwargs.get('k', 4)
mxtr = kwargs.get('mxtr', 20)
psln = kwargs.get('psln', 1.0)
Z = zobr.reshape((-1, 3))
Z = numpy.float32(Z)
crtrn = (cv2.TERM_CRTRN_EPS + cv2.TERM_CRTRN_MXTR, mxtr, psln)
rt, lbfl, cntr = cv2.kseredns(Z, k, None, crtrn, 10,
cv2.KSEREDNS_RANDOM_CNTRS)
cntr = numpy.uint8(cntr)
rs = cntr[lbfl.flatten()]
rs2 = res.reshape((zobr.shape))
return rs2

def Thrshldng_modif(zobr, **kwargs):
    flg = kwargs.get('flg', 0)
    modif_mthd = kwargs.get('modif_mthd', 0)
    blszf = kwargs.get('blszf', 3)
    c = kwargs.get('c', 0)
    zobr = cv.cvtColor(zobr, cv.COLOR_RGB2GRAY)
    out_zobr = cv.modifThreshold(zobr, 255, modif_mthd, flag, blszf, c)
    return out_zobr

def vodned(zobr, **kwargs):
    print(kwargs)
    kst = krnzst = kwargs.get('krnzst', 3)
    fst = flthd = kwargs.get("flthd", 0)
    zobr_gray = cv2.cvtColor(zobr, cv2.COLOR_BGR2GRAY)
    _, zobr_bin = cv2.threshold(zobr_gray, 0, 255, fs + cv2.THRESH_OTSU)
    zobr_bin = cv2.morphologyEx(zobr_bin, numpy.ones((ks, ks),
dhtype=int))
    if __name__ != "__main__":
        plt.imshow(zobr_bin)
        plt.text(0.5, -10.0, 'Ваші об'єкти повинні бути виділені(255)',
horizontalalignment='cntr',\
                verticalalignment='cntr')
        plt.ion()

```

```

plt.show()
grnc = cv2.dilate(zobr_bin, None, iterations=5)
grnc = grnc - cv2.erode(grnc, None)
dht = cv2.distanceTransform(zobr_bin, 2, 3)
dht = ((dht - dht.min()) / (dht.max() - dht.min())) *
255).astype(numpy.uint8)
_, dht = cv2.threshold(dht, 0.6 * dht.max(), 255, cv2.THRESH_BINARY)
lbfl, ncdc= lbfl(dht)
lbfl[grnc == 255] = 255
lbfl = lbfl * (255 / (ncdc + 1))
lbfl[grnc == 255] = 255
lbfl = lbfl.astype(numpy.int32)
rs = cv2.vodned(zobr, lbfl)
rs = res.astype(numpy.uint8)
result = 255 - rs
return result if __name__ == "__main__":
pth = input('Input path to zobr: ')
pth = pth.replace('\\"', '')
pth = pth.replace("\\'", '')
zobr = vodned(cv2.imread(pth), **{'krnzst': 3, 'flthd': 1})
plt.imshow(zobr)
plt.show()

```

ДОДАТОК В
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Дослідження та програмна реалізація засобів сегментації зображень

Виконав: ст. гр. КНТ-112м

Владислав ПЕРЕГУДА

Керівник:
д.т.н., професор

Андрій ОЛІЙНИК

1

Рисунок В.1 – Слайд 1

Об'єкт дослідження – процес розробки програмного забезпечення для обробки зображень.

Предмет дослідження – програмні засоби для сегментації зображень.

Метою роботи є дослідження та розробка програмного забезпечення для сегментації зображень.

2

Рисунок В.2 – Слайд 2

Задачи роботи:

- виконати аналіз предметної області та програмних засобів для обробки зображень шляхом їх сегментації;
- розробити модель взаємодії користувача з програмним забезпеченням для сегментації зображень;
- здійснити проектування програмного забезпечення для сегментації зображень;
- створити програмне забезпечення для сегментації зображень;
- виконати тестування розробленого програмного забезпечення для сегментації зображень.

3

Рисунок В.3 – Слайд 3

Дослідження програмних засобів для сегментації зображень

Критерій порівняння	Sefexa Image Segmentation Tool	Image Segmenter	CVIPtools
Підтримка широкого набору методів сегментації	+	+	–
Необхідність додаткового забезпечення для функціонування програми	–	+	–
Зручність засобів візуалізації результатів	+–	+	–
Можливість роботи з різними наборами даних	+	+	+–
Автоматизований аналіз великих наборів даних	+	+–	–

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Python	Matlab	C++
Кросплатформенність	+	–	+
Зручна типізація	+	+–	–
Простий синтаксис	+	+–	+
Активна спільнота розробників	+	–	+–
Велика кількість бібліотек для обробки зображень	+	+–	+–
Можливість розширення	+	+–	+

5

Рисунок В.5 – Слайд 5

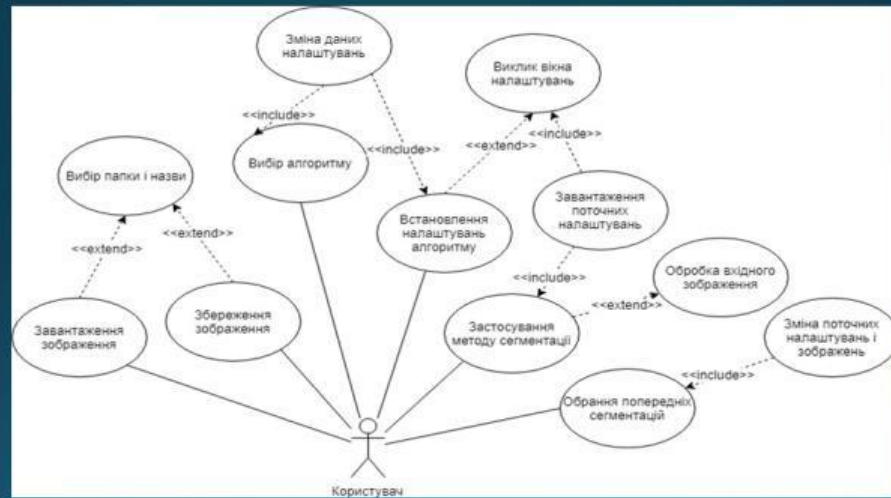
Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовище розробки		
	PyCharm	Microsoft Visual Studio	Sublime Text
Мови програмування	Python	Багато мов, включаючи C++, C#, Python, Java, та інші	Багато мов, включаючи Python та інші
Зручність розробки на мові Python	+	+–	+–
Безкоштовність	+	+–	+–
Наявність розширень для обробки зображень	+	+	–
Візуальний редактор	+	+	–

6

Рисунок В.6 – Слайд 6

Модель взаємодії користувача для сегментації зображень



7

Рисунок В.7 – Слайд 7

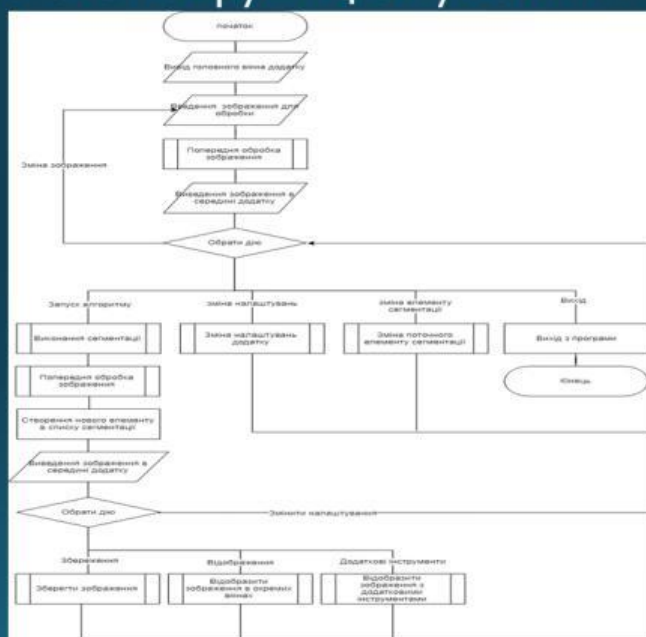
Структура програмного забезпечення для сегментації зображень



8

Рисунок В.8 – Слайд 8

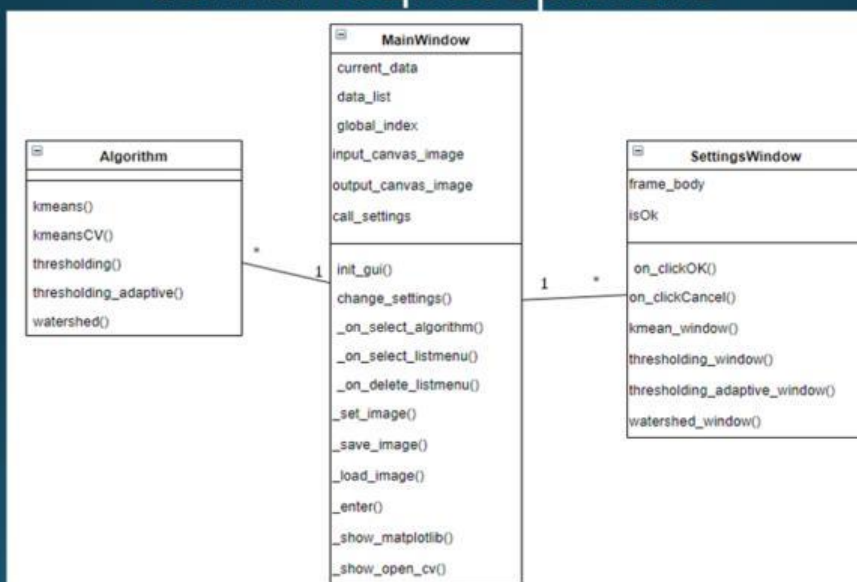
Загальна схема функціонування програми



9

Рисунок В.9 – Слайд 9

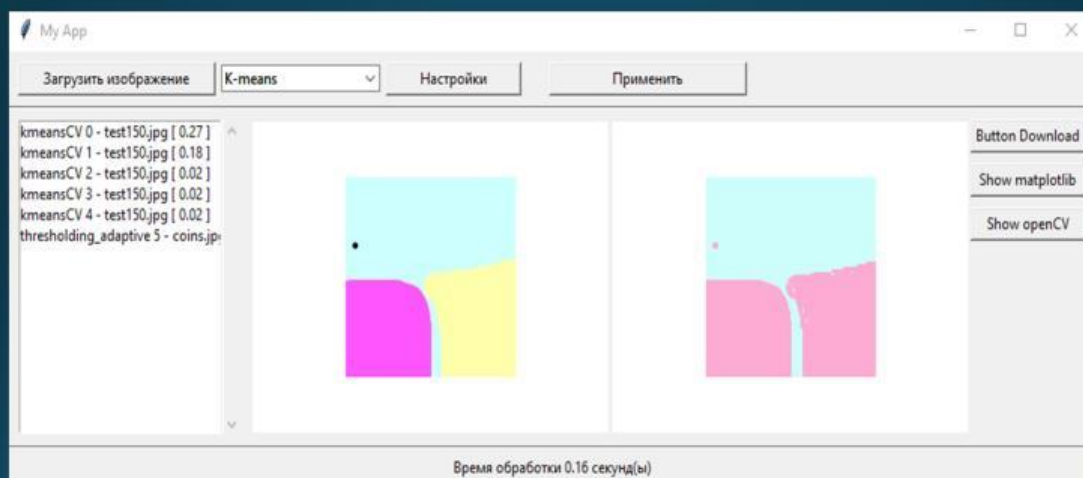
Діаграма класів програмного забезпечення для сегментації зображень



10

Рисунок В.10 – Слайд 10

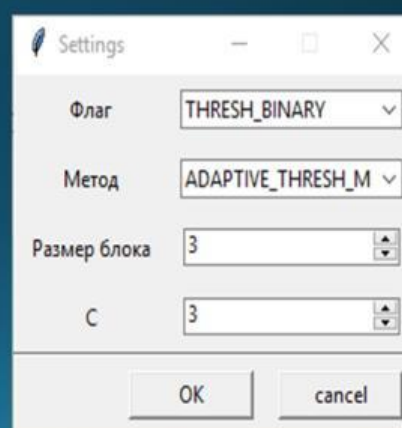
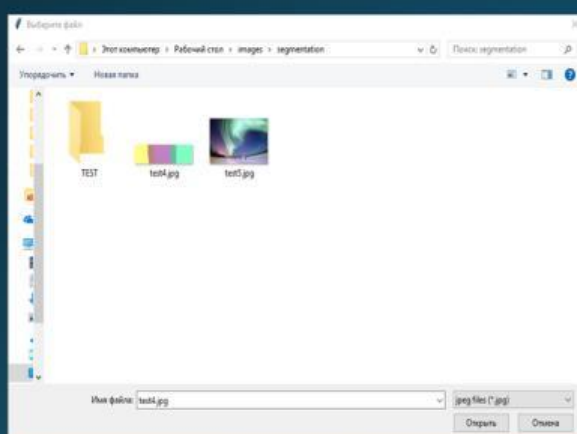
Робота з програмою. Головне вікно



11

Рисунок В.11 – Слайд 11

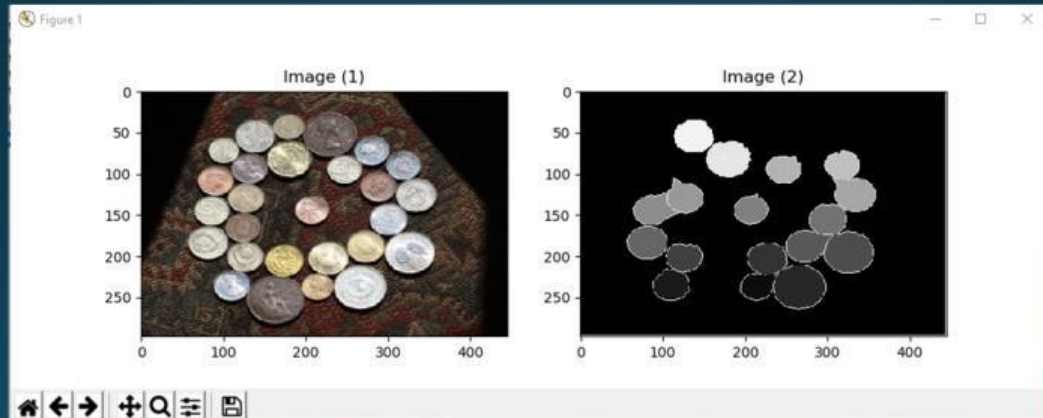
Робота з програмою. Вікно вибору зображення. Налаштування параметрів методу сегментації зображення



12

Рисунок В.12 – Слайд 12

Робота з програмою. Форма перегляду базового та сегментованого зображень



13

Рисунок В.13 – Слайд 13

Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для обробки зображень.

Новизна роботи полягає в тому, що запропоновано модель взаємодії користувача з програмним забезпеченням для сегментації зображень, яка подана у вигляді діаграми прецедентів у нотації UML, та дозволяє розроблювати програмне забезпечення для сегментації зображень.

Практичне значення роботи полягає в тому, що розроблено програмне забезпечення для сегментації зображень.

Результати тестування програмного забезпечення для сегментації зображень показали, що розроблена програма дозволяє розв'язувати завдання виділення об'єктів на зображеннях за допомогою різних методів сегментації.

14

Рисунок В.14 – Слайд 14