

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерних наук \_\_\_\_\_

(повна назва)

Кафедра \_\_\_\_\_ Програмної інженерії \_\_\_\_\_

(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

\_\_\_\_\_ Дослідження і розробка інформаційної технології визначення  
\_\_\_\_\_ розміру взуття за зображенням \_\_\_\_\_  
(тема)

Виконав:

студент 2 курсу, групи ІІЗМ-21-3 \_\_\_\_\_

Сахаров І.Л. \_\_\_\_\_

(прізвище, ініціали)

Спеціальність 121 Інженерія програмного  
забезпечення \_\_\_\_\_

(шифр і назва спеціальності)

Тип програми освітньо-наукова \_\_\_\_\_

(освітньо-професійна або освітньо-наукова)

Керівник проф. Смеляков К.С. \_\_\_\_\_

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_

(підпис)

Дудар З.В. \_\_\_\_\_

(прізвище, ініціали)

Харків 2023

## Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наукКафедра Програмної інженеріїРівень вищої освіти другий (магістерський)Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-наукова програмаОсвітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_

(підпис)

" \_\_\_\_\_ " \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**студента Сахарова Івана Леонідовича

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження і розробка інформаційної технології визначення розміру взуття за зображенням

затверджена наказом по університету від «29» березня 2023 р. № 302Ст

2. Термін подання студентом роботи до екзаменаційної комісії: 10 травня 2023 р.

3. Вихідні дані до проекту: інформаційна технологія визначення розміру взуття за зображенням, пояснювальна записка.

4. Перелік питань, що потрібно опрацювати в роботі: мета роботи; аналіз предметної області; огляд існуючих досліджень; постановка задачі; опис обраних методів порогуювання, розмиття та кластеризації зображення, виявлення контурів; вибір технології розпізнавання об'єктів у режимі реального часу; типовий опис даних; реалізація інформаційної технології визначення розміру взуття; виконання експерименту; обмеження щодо фотографій.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання кваліфікаційної роботи	23.01.2023	виконано
2	Дослідження матеріалів та наукових робіт	24.01.2023 – 05.02.2023	виконано
3	Постановка задачі	06.02.2023 – 08.02.2023	виконано
4	Планування експериментальної частини дослідження	09.02.2023 – 15.02.2023	виконано
5	Розробка програмного забезпечення	16.02.2023 – 05.03.2023	виконано
6	Проведення експериментів	06.03.2023 – 14.03.2023	виконано
7	Підготовка та подача статі публікації	15.03.2023 – 02.04.2023	виконано
8	Оформлення пояснювальної записки	03.04.2023 – 30.04.2023	виконано
9	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту	за 5 днів	виконано
10	Представлення на рецензування	за 3 дні	виконано
11	Представлення кваліфікаційної роботи в ДЕК	за 2 дні	виконано

Дата видачі завдання      23 січня 2023 р.

Студент

\_\_\_\_\_ (підпис)

Керівник роботи

\_\_\_\_\_ (підпис)

\_\_\_\_\_ проф. Смеляков К.С.

\_\_\_\_\_ (посада, прізвище, ініціали)

## РЕФЕРАТ

Кваліфікаційна робота магістра містить: 54 с., 4 табл., 27 рис., 32 джер.

### ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ, КОМП'ЮТЕРНИЙ ЗІР, ВИЗНАЧЕННЯ РОЗМІРУ ВЗУТТЯ, ДВОВИМІРНЕ ЗОБРАЖЕННЯ, PYTHON, OPENCV

Об'єктом дослідження є методи виділення контурів, технології комп'ютерного зору і машинного навчання.

Мета роботи – розробка інформаційної технології визначення розміру взуття за двовимірним зображенням.

Методи дослідження базуються на технології OpenCV.

У результаті роботи проведено аналіз предметної області, особливостей і проблем, що є предметом дослідження. Виконано огляд наукової літератури та існуючих досліджень; аналіз методів порогування, розмиття та кластеризації зображення, виявлення контурів. Використовуючи метод лінійної згортки обрано технологію розпізнавання об'єктів у режимі реального часу, яка буде використовуватися для реалізації мети роботи. На основі схожих робіт, розроблено покращений алгоритм визначення висоти стопи та розміру взуття. Проведено серію експериментів, виходячи з яких сформовано обмеження щодо фотографій.

### INFORMATION TECHNOLOGY, COMPUTER VISION, SHOE SIZE DETERMINATION, TWO-DIMENSIONAL IMAGE, PYTHON, OPENCV

The object of the research are methods of contour selection, computer vision and machine learning technologies.

The purpose of the work is to develop an information technology for determining shoe size based by a two-dimensional image.

Research methods are based on OpenCV technology.

As a result of the work, the subject area, features and problems that are the subject of the study were analyzed. A review of scientific literature and existing studies has been carried out; analysis of thresholding, blurring and image clustering methods,

and contour detection has been performed. Using the linear convolution method, the real-time object recognition technology was chosen to achieve the goal of the work. Based on similar works, an improved algorithm for determining foot height and shoe size was developed. A series of experiments were conducted, based on which restrictions on photos were formed.

Умови публікації пояснювальної записки

Я,

Сахаров Іван Леонідович

(прізвище, ім'я, по батькові)

студент групи ПЗМ-21-3 здобувач вищої освіти на другому (магістерському) рівні

кафедра

програмної інженерії

(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження і розробка інформаційної технології визначення розміру взуття за зображенням

(назва роботи)

що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

## ЗМІСТ

Вступ.....	7
1 Огляд сучасного стану проблеми.....	9
1.1 Аналіз предметної області магістерського дослідження, особливості та проблеми, які є предметом дослідження .....	9
1.2 Огляд існуючих досліджень .....	9
1.3 Постановка задачі .....	13
2 Вибір та обґрунтування методу реалізації мети дослідження .....	15
2.1 Опис обраних методів порогування.....	15
2.1.1 Порогування Оцу .....	15
2.1.2 Адаптивне порогування.....	17
2.2 Опис обраних методів розмиття зображення .....	19
2.2.1 Розмиття за Гаусом .....	19
2.2.2 Медіанна фільтрація .....	22
2.3 Опис обраних методів кластеризації зображення .....	23
2.4 Опис обраних методів виявлення контурів .....	25
2.5 Вибір технології розпізнавання об'єктів у режимі реального часу .....	28
2.6 Розробка плану розробки алгоритму .....	31
3 Розробка інформаційної технології.....	34
3.1 Типовий опис даних.....	34
3.2 Реалізація інформаційної технології визначення розміру взуття.....	35
3.2.1 Реалізація сканера аркуша паперу з фотографії.....	35
3.2.2 Сегментація зображення на основі кольорів.....	41
3.2.3 Вимірювання розміру стопи.....	43
4 Проведення експерименту .....	48
Висновки.....	50
Перелік джерел посилання .....	51
Додаток А Слайди презентації .....	55
Додаток Б Лістинг коду .....	62
Додаток В Звіт результатів перевірки на унікальність тексту .....	68
Додаток Г Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії .....	69

## ВСТУП

У сучасному світі все частіше і актуальніше стає питання розробки і застосування технологій інтелектуального аналізу даних та комп'ютерного зору для вирішення численних завдань. Тому виникає потреба в створенні механізму допомоги сферах життєдіяльності людини. Таким прикладом може бути вирішення проблеми визначення довжини стопи та розміру взуття.

Комп'ютерний зір – це технологія, створена на основі машинного навчання. За допомогою неї машини можуть знаходити, відстежувати, ідентифікувати та класифікувати об'єкти, витягуючи дані і робити аналіз отриманої інформації з зображень. Ця технологія застосовується для розпізнавання об'єктів, відеоаналітики, опису змісту зображень і відео, розпізнавання жестів і рукописного введення, для інтелектуальної обробки всього, що можна побачити людським оком.

Стопа – дуже важлива частина людського тіла, тому що є єдиною, яка контактує з землею, що відіграє життєво важливу роль у навантаженні тіла людини. Взуттєві вироби підтримують і захищають стопу під час стояння, ходьби та стрибків. Медичні дослідження показують, що неправильно підібране взуття може спричинити біль у стопах, неправильну ходу та інші проблеми. Носіння тривалий час непридатного взуття може не тільки спричинити різноманітні захворювання ніг, але й легко спричинити травми.

Метою і задачею роботи є розробка інформаційної технології визначення розміру взуття за двовимірним зображенням, яка буде корисна користувачам при купівлі взуття онлайн у різних системах електронної комерції.

Об'єктом дослідження є методи виділення контурів, технології комп'ютерного зору і машинного навчання.

Методи дослідження базуються на технології OpenCV та включають методи пороговання (адаптивне пороговання та метод Оцу), розмиття зображення (за Гаусом та медіанна фільтрація), k-means для кластеризації зображення та виявлення контурів (за Кенні).

Елементи наукової новизни та/або здійснення інновацій в умовах невизначеності вимог в якості одержаних результатів – розроблена технологія є удосконаленою версією існуючих розробок. Вона дозволяє оброблювати фотографії, зняті при різних дефектах трансформації та освітлення; швидко вилучати скан аркуша паперу формату А4 з зображення та отримувати результат у вигляді повідомлення з зазначеною довжиною строки та розміру взуття за стандартом EU.

Практичне значення одержаних результатів – таке рішення, згодом, можливо інтегрувати у будь-яку систему електронної комерції для спрощення купівлі взуття онлайн.

Результати дослідження опубліковано у збірнику матеріалів 7-ої відкритої міжнародної конференції «Electrical, Electronic and Information Sciences» eStream 2023 у Вільнюсі, Литва [15].

## 1 ОГЛЯД СУЧАСНОГО СТАНУ ПРОБЛЕМИ

1.1 Аналіз предметної області магістерського дослідження, особливості та проблеми, які є предметом дослідження

Найбільше практичне використання серед методів, що застосовуються для розпізнавання зображень об'єктів, знайшли спектральні та синтаксичні методи. Але багато практичних задач, пов'язаних з процесами оперативного управління та контролю, вимагають вирішення проблеми розпізнавання у реальному часі. Контури об'єктів, що розпізнаються, є найбільш простою для використання та найбільш інформативною складовою цих зображень [1].

Як приклад для проведення дослідження обрано систему електронної комерції з продажу взуття. Головною проблемою покупки взуття онлайн є замовлення товару потрібного розміру, так як в кожній країні використовують свою систему вимірювання ступні. Тому є рішення інтегрувати в таку систему можливість надіслати фотографію обведеної на аркуші паперу стопи ноги користувача і інформаційна система автоматично визначила, якому розміру вона відповідає.

Предметом розробки є інформаційні технології й програмні методи створення системи розпізнавання контурів на двовимірних зображеннях у реальному часу.

### 1.2 Огляд існуючих досліджень

На основі останніх досліджень існує багато методів визначення розміру взуття.

Дослідження [4] пропонує наскрізну систему для вимірювання розміру стопи людини. Основними проблемами під час вимірювання розміру стопи було

точне виділення межі стопи та зміни масштабу розміру стопи через його глибину на знятому зображенні.

Запропонована система виділяє ногу з вхідного зображення за допомогою сегментації кольорової маски, яка відокремлює ногу від шуму на фоні. Згідно з цією методологією, шкала варіації розміру стопи була нормалізована з використанням еталонного об'єкта; у цьому випадку це монета місцевої валюти з будь-яким номіналом. Еталонний об'єкт фіксується разом зі ступнею, а потім відокремлюється для оцінки глибини. Щоб точно виміряти розмір стопи, модель лінійної регресії навчається та використовується для прогнозування результатів, а середньоквадратичне відхилення – для перевірки точності прогнозованих результатів.

Таке рішення визначення розміру стопи має ряд недоліків:

- відсутність вилучення контура аркуша паперу на фотографії;
- вхідне зображення повинно бути у вигляді готового скана.

Дослідження [5] пропонує поєднати глибоке навчання та традиційну технологію обробки зображень для вимірювання стопи на основі краю зображення з використанням відповідної інформації про край. Використовуючи стандартний папір формату А4 як фон, отримується зображення ніг.

Коли алгоритм вимірює стопу, немає необхідності проходити процес 3D-реконструкції, і розмір стопи вимірюється швидко та точно. Алгоритм можна розділити на три основні частини: визначення країв аркуша паперу формату А4; сегментація стопи; вимірювання стопи. Алгоритм складається з наступних етапів:

- на вхід подається зображення з аркушем формату А4;
- використання глибокого навчання для виявлення країв зображення;
- бінаризація зображення;
- витяг контура аркуша, щоб отримати координати чотирьох точок;
- кадрування фотографії для отримання зображення стопи на папері формату А4;
- сегментація області стопи і вимірювання розміру стопи.

Головними недоліками такого рішення є:

- неможливість роботи із зображеннями, зняті при слабкому освітленні або наявністю тіней.

- неможливість роботи з обведеним контуром стопи.

Наступні розглянуті дослідження вирішують одну конкретну задачу, які будуть корисні при реалізації інформаційної технології визначення розміру взуття за фотографією у рамках даної кваліфікаційної роботи.

У дослідженні [6] подано простий алгоритм реалізації сканера аркуша. Він складається з трьох етапів: виявлення країв, знаходження контурів та трансформація перспективи. Для визначення країв зображення використовується алгоритм виявлення контурів Кенні. Для кращої точності, виявлення контурів здійснюється наступним чином:

- бінаризація зображення;
- пошук контурів за допомогою функції `findContours()` бібліотеки `OpenCV`;
- малювання знайдених контурів на зображенні.

Трансформація перспективи складається з наступних кроків:

- завантаження зображення;
- обрання 4 точок по порядку: верхня ліва, верхня права, нижня ліва, нижня права;
- малювання кола, щоб показати точні точки, які обрано;
- створення списку із цими 4 точками;
- створення нового набору із 4 точок (ці 4 точки є розміром нового вікна, для відображення зображення);
- застосування перетворення перспективи, щоб створити матрицю;
- деформувати зображення за допомогою оригінального каркасу і щойно створенію матриці.

У статті [9] приведено покращений алгоритм Кенні для виявлення контурів. Алгоритм використовує метод `MeanShift` замість фільтрації Гауса, який зберігає краєву інформацію під час усунення шумів. Також використовується алгоритм

максимальної міжкласової дисперсії (OSTU) для отримання адаптивного оптимального порогу та покращення адаптивності алгоритму.

У статті [7] описано найбільш популярні уніфіковані моделі градаційної корекції затемненого цифрового зображення, сформульовано практичні рекомендації щодо використання розглянутих моделей градаційної корекції при вирішенні актуальних прикладних задач підвищення якості затемнених цифрових зображень.

У статті [8] проведено порівняльний аналіз застосування найбільш затребуваних моделей градаційної корекції сильно освітленого цифрового зображення, здатних до автоматичної адаптації до різних шкал яскравості, розглянуто особливості їх практичного застосування, налаштування експерименту для покращення яскраво освітленої фотографії.

В даному дослідженні тестове зображення модифікується з використанням різних моделей корекції градації та різних параметрів, що допомагає показати практичну цінність таких модифікацій, а коефіцієнт покращення зображення дається для порівняльного аналізу впливу вхідних параметрів на кінцевий результат.

У статті [10] пропонується ефективна модель покращення темного зображення, яка вирішує важливі проблеми погіршення якості шляхом включення гамма-корекції, алгоритму балансу білого, усунення шумів і попередніх методів темного каналу. Техніка гамма-корекції реалізована для вирішення проблеми низької контрастності темних зображень.

Дослідження [11] приведено новий метод багатокольорного балансування для забезпечення константності кольорів. Запропонований метод, який називається  $n$ -кольорним балансуванням, дозволяє не лише ідеально відкоригувати  $n$  цільових кольорів на основі відповідних базових кольорів, але й відкоригувати кольори, відмінні від цих  $n$  кольорів. У результаті, запропонований метод балансування білого може ідеально відкоригувати білий колір.

У статті [12] пропонується простий, але ефективний метод видалення тіней із зображень текстових документів. Метод оцінює глобальні та локальні фонові

кольори за допомогою статистичного аналізу всього зображення та локальних околиць. За допомогою карти тіней зображення без тіней можна відновити за допомогою внутрішньої декомпозиції.

У статті [13] запропоновано двофазну повністю автоматичну схему відновлення зображень, які були пошкоджені розмиттям, імпульсним шумом і шумом Гауса. На першому етапі медіанний фільтр використовується для часткової корекції більшості пікселів, які забруднені імпульсним шумом. Потім зображення відновлюється шляхом вирішення  $\ell_p$ - $\ell_q$  мінімізації. Даний алгоритм може відновлювати зображення, які були забруднені розмиттям і мають до 70% пікселів, забруднених шумом «сіль-перець».

Дослідження [14] пропонує вдосконалений алгоритм сегментації методом k-середніх. Спочатку вихідне зображення перетворюється з простору RGB у простір HSV, потім оригінальне зображення фільтрується та запускається алгоритм пошуку викидів. Нарешті, виконується сегментація зображень кластеризації методом k-середніх.

### 1.3 Постановка задачі

Метою роботи є розробка інформаційної технології визначення розміру взуття за двовимірним зображенням, яка буде корисна користувачам при купівлі взуття онлайн у різних системах електронної комерції.

Розроблений алгоритм на початковому етапі повинен вилучати аркуш паперу з фотографії, перетворюючи його на скан документа; виконувати кластеризацію отриманого скану, використовуючи метод k-means з метою сегментації на основі кольорів; створювати обмежувальну рамку для отримання висоти паперу та стопи ноги на кластерному зображенні з метою визначення розміру взуття.

Використовуючи набір даних, що містить фотографії аркуша формату A4, зняті при різному освітленні з різних кутів, виконати експерименти, результати

яких дозволять сформуванню узагальненого алгоритму визначення розміру взуття за двовимірним зображенням та обмеження щодо фотографій.

Даний алгоритм повинен вирішувати наступні проблеми:

- завантаження фото обведеної стопи ноги;
- обробки фотографій зняті в умовах: різної освітленості, наявності тіней, з будь-яких кутів та поворотів.
- знаходження кордонів аркуша формату А4 на фото;
- знаходження кордонів стопи на аркуші;
- визначення висоти стопи та розміру взуття;
- точність (похибка) виміру; зображення поганої якості (низька роздільна здатність, погане освітлення).

Алгоритм повинен бути розроблений на мові Python, з використанням можливостей бібліотеки OpenCV. Операційна система – macOS 11 або вище, інтегроване середовище розробки – Microsoft Visual Studio Code 2019 або вище.

## 2 ВИБІР ТА ОБҐРУНТУВАННЯ МЕТОДУ РЕАЛІЗАЦІЇ МЕТИ ДОСЛІДЖЕННЯ

### 2.1 Опис обраних методів порогування

В рамках даної роботи розглянуто два методи порогування: метод Оцу та адаптивне порогування.

#### 2.1.1 Порогування Оцу

Процес відділення пікселів переднього плану від фону називається пороговим визначенням (порогуванням). Метод Оцу, запропонований Нобуюкі Оцу, є одним з таких методів, який лежить на основі дисперсії для знаходження порогового значення, де зважена дисперсія між пікселями переднього плану та фону є найменшою. Ключова ідея полягає в тому, щоб перебрати всі можливі значення порогу та виміряти розподіл пікселів фону та переднього плану. Потім знаходиться поріг, де поширення найменше.

Алгоритм ітераційно шукає порогове значення, яке мінімізує дисперсію всередині класу, визначену як зважена сума дисперсій двох класів (фонового та переднього плану). Кольори в градаціях сірого зазвичай між 0-255 (0-1 у випадку з плаваючою точкою). Отже, якщо обрати порогове значення 100, тоді всі пікселі зі значеннями, меншими за 100, стануть фоном, а всі пікселі зі значеннями, більшими або рівними 100, стануть переднім планом зображення [16].

Формула для знаходження дисперсії всередині класу при будь-якому пороговому значенні  $t$  визначається як:

$$\sigma^2 = \omega_{bg}(t)\sigma_{bg}^2(t) + \omega_{fg}(t)\sigma_{fg}^2(t),$$

де  $\omega_{bg}(t)$  та  $\omega_{fg}(t)$  являє собою ймовірність кількості пікселів для кожного класу при пороговому значенні  $t$ , а  $\sigma^2$  представляє дисперсію значень кольорів.

Ймовірність кількості пікселів для кожного класу при пороговому значенні  $t$  визначається наступним чином:

$$\omega_{bg}(t) = \frac{P_{BG}(t)}{P_{all}}, \omega_{fg}(t) = \frac{P_{FG}(t)}{P_{all}},$$

де  $P_{all}$  являє собою загальну кількість пікселів у зображенні,  $P_{BG}(t)$  – кількість фонових пікселів (заднього плану) на порозі  $t$ , а  $P_{FG}(t)$  – кількість пікселів переднього плану на пороговому значенні  $t$ .

Дисперсію значень кольорів можна розрахувати за наведеною нижче формулою:

$$\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{N - 1}$$

де  $x_i$  являє собою значення пікселя в  $i$ -й групі ( $bg$  або  $fg$ ),  $\bar{x}$  – середнє значення пікселів у групі  $bg$  або  $fg$ ,  $N$  – кількість пікселів.

До переваг пороговання методом Оцу можна віднести:

- висока швидкість, оскільки пороговання Оцу працює на гістограмах (які є масивами цілих чи плаваючих елементів довжиною 256);
- простота у використанні кодування.

До недоліків даного методу відносять:

- освітлення на зображенні повинно бути рівномірним;
- гістограма повинна бути бімодальною (як і зображення);
- він не використовує структуру об'єктів або просторову когерентність.

Приклад роботи пороговання Оцу подано на рисунку 2.1.

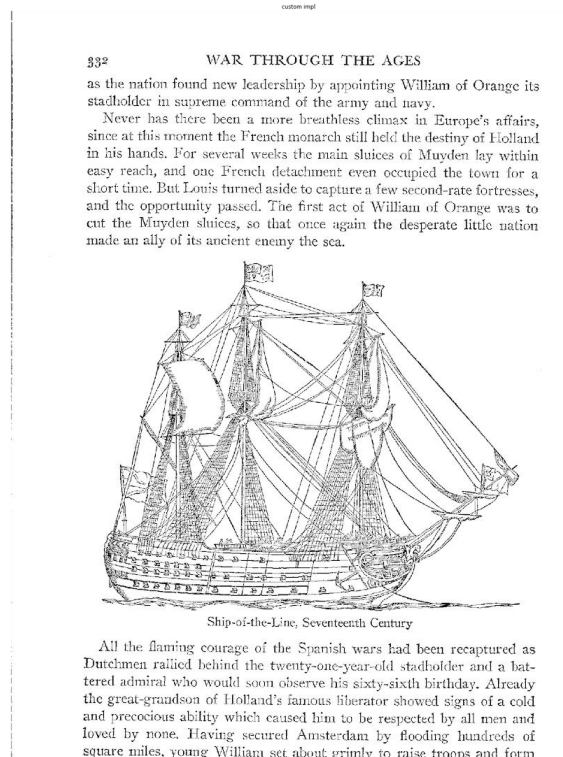
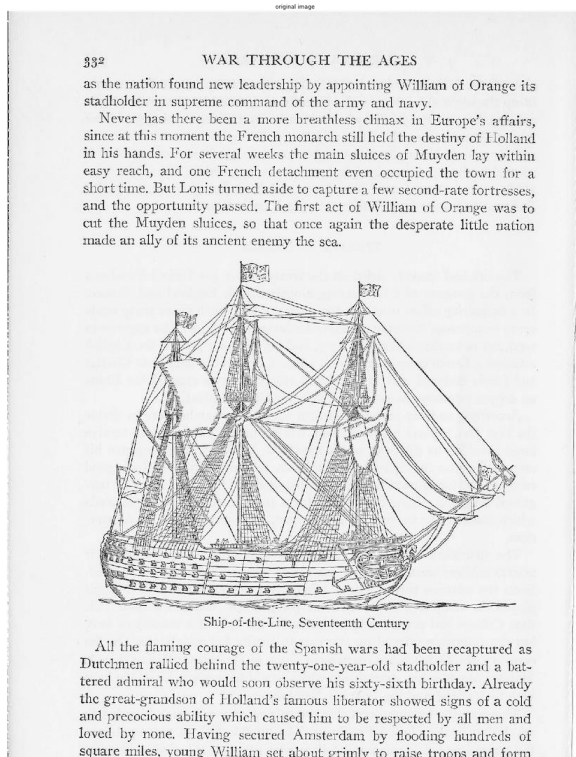


Рисунок 2.1 – Приклад роботи порогування Оцу [16]

## 2.1.2 Адаптивне порогування

Адаптивне порогування враховує одночасно невеликий набір сусідніх пікселів, обчислює порогове значення  $T$  для конкретної локальної області, а потім виконує сегментацію [17].

Адаптивне порогове значення зазвичай приймає зображення у градаціях сірого або кольорове зображення як вхідні дані та, у найпростішій реалізації, виводить двійкове зображення, що представляє сегментацію. Для кожного пікселя на зображенні необхідно обчислити порогове значення. Якщо значення пікселя нижче порогового значення, воно встановлюється на фонове значення, інакше приймається значення переднього плану.

Існує два основні підходи до визначення порогу: підхід Чоу та Канеко та локальне порогове значення. Припущення, що лежить в основі обох методів, полягає в тому, що менші області зображення, швидше за все, матимуть приблизно рівномірне освітлення, тому вони більш придатні для порогового значення. Методи Чоу та Канеко ділять зображення на масив часткових

зображень, що перекриваються, а потім знаходять оптимальний поріг для кожного субзображення, досліджуючи його гістограму. Порогове значення для кожного окремого пікселя визначається шляхом інтерполяції результатів підзображень.

Альтернативним підходом до визначення локального порогу є статистичне дослідження значень інтенсивності локального оточення кожного пікселя. Статистика, яка є найбільш прийнятною, значною мірою залежить від вхідного зображення.

Прості та швидкі функції включають середнє локального розподілу інтенсивності:

$$T = mean,$$

медіанне значення:

$$T = median$$

або середнє мінімального та максимального значень:

$$T = \frac{\max + \min}{2}$$

Розмір околиці має бути достатньо великим, щоб охопити достатню кількість пікселів переднього плану та фону, інакше вибирається поганий поріг. З іншого боку, вибір занадто великих областей може порушити припущення приблизно рівномірного освітлення.

Цей метод потребує менше обчислень, ніж підхід Чоу та Канеко, і дає хороші результати для деяких програм [18].

Приклад застосування адаптивного порогоування зображено на рисунку 2.2.

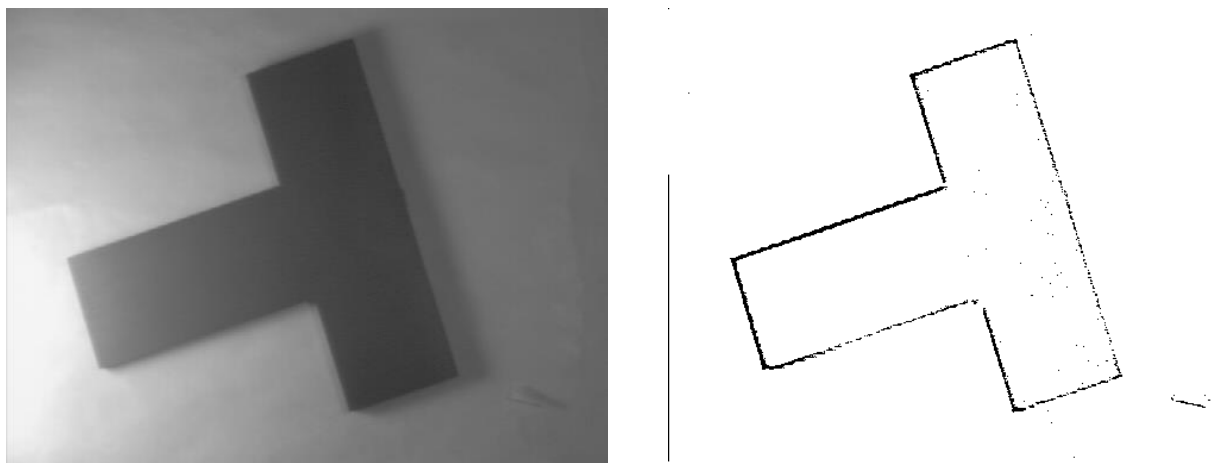


Рисунок 2.2 – Приклад застосування адаптивного порогування [17]

Переваги використання адаптивного порогування:

- за допомогою даного методу можна отримати кращу сегментацію, оскільки порогове значення вибирається шляхом аналізу інтенсивності всіх пікселів зображення, а потім вибирається найбільш підходяще порогове значення, яке може ефективно відокремити передній план від фону;
- цей метод економить час і дозволяє швидко відокремити основний об'єкт від решти зображення [32].

## 2.2 Опис обраних методів розмиття зображення

В рамках даної роботи розглянуто два методи розмиття зображення: розмиття за Гаусом та медіанна фільтрація.

### 2.2.1 Розмиття за Гаусом

Розмиття за Гаусом описує розмиття зображення за допомогою функції Гауса. Це широко використовуваний ефект, як правило, для зменшення шуму зображення та деталей.



Рисунок 2.3 – Приклад зменшення шуму використовуючи ефект Гауса [31]

На рисунку 2.3 зображено приклад зменшення шуму за допомогою фільтру Гауса. Основні причини такого шуму у цифрових зображеннях виникають під час зйомки (наприклад, шум датчика, спричинений недостатнім освітленням, надмірною температурою або передачею). Під час згладжування зображення може виникнути небажаний ефект у вигляді розмивання тонкомасштабних меж і особливостей зображення, оскільки вони корелюють із заблокованими високими частотами. Середнє значення (згортка), медіана та гаусове згладжування є традиційними методами просторової фільтрації для зменшення шуму [31].

Рівняння функції Гауса у двох вимірах має наступний вигляд:

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

де  $x$  – відстань від початку координат по горизонтальній осі,  $y$  – відстань від початку координат по вертикальній осі, а  $\sigma$  – стандартне відхилення розподілу Гауса.

При застосуванні цієї формули у двох вимірах створюється поверхня, контури якої є концентричними колами з розподілом Гауса від центральної точки. Значення цього розподілу використовуються для побудови матриці згортки, яка застосовується до вихідного зображення. Нове значення кожного пікселя встановлюється як середньозважене значення околиці цього пікселя. Значення

вихідного пікселя отримує найбільшу вагу (має найвище значення Гауса), а сусідні пікселі отримують менші ваги, оскільки їх відстань до вихідного пікселя збільшується. Це призводить до розмиття, яке зберігає межі та краї краще, ніж інші, однорідніші фільтри розмивання [19].

Гаусове розмиття часто використовується для задачі визначенням країв, так як більшість алгоритмів виявлення країв чутливі до шуму. Використання цього фільтра спочатку має на меті зменшити рівень шуму в зображенні, що покращує результат наступного алгоритму визначення країв (цей підхід зазвичай називають фільтрацією Лапласа Гаусса або фільтрацією LoG).

Приклад розмиття зображення методом Гауса з різною інтенсивністю подано на рисунку 2.4.

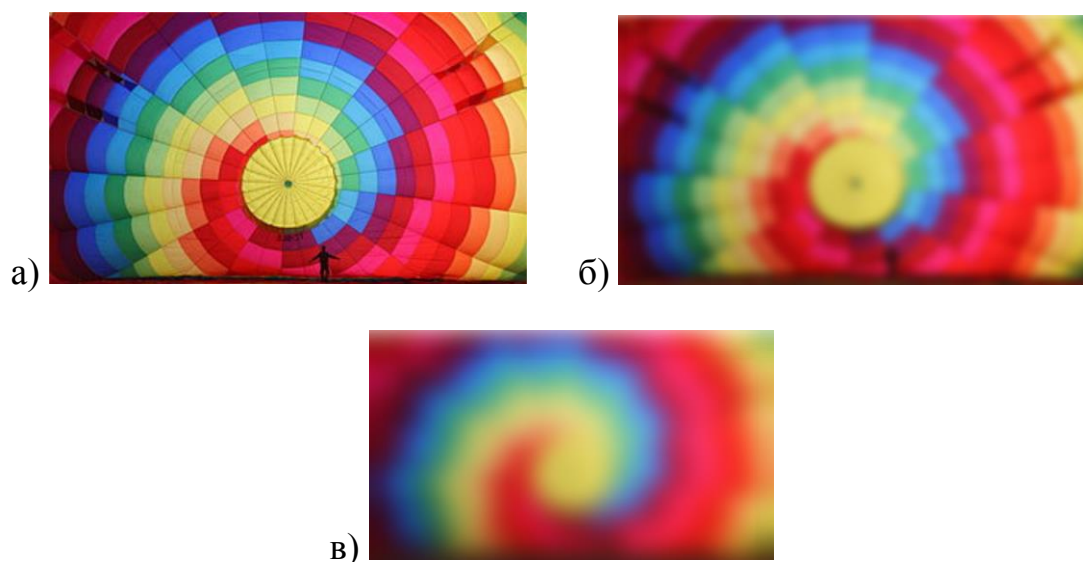


Рисунок 2.4 – Різниця між малим і великим розмиттям за Гаусом: а) оригінальне зображення; б) інтенсивність розмиття = 3; в) інтенсивність розмиття = 10 [19]

Отже, гаусівський фільтр використовується за наступними причинами:

- це допомагає усунути шум від розмитих зображень;
- він використовується для зменшення шуму на зображенні, зробленого за слабким освітлення;
- він може усунути яскраві пікселі;
- це допомагає зробити краї рівними;

- він раціонально симетричний;
- за його допомогою можна визначити ступінь згладжування.

### 2.2.2 Медіанна фільтрація

Медіанна фільтрація – це нелінійний метод, який використовується для видалення шумів із зображень. Його широко використовують, оскільки він дуже ефективний для видалення шуму, зберігаючи краї. Він особливо ефективний для видалення шуму типу «сіль і перець».

На рисунку 2.5 подано приклад використання медіанного фільтра до зашумленого зображення з трьома різними значеннями радіусу вікна фільтрації.

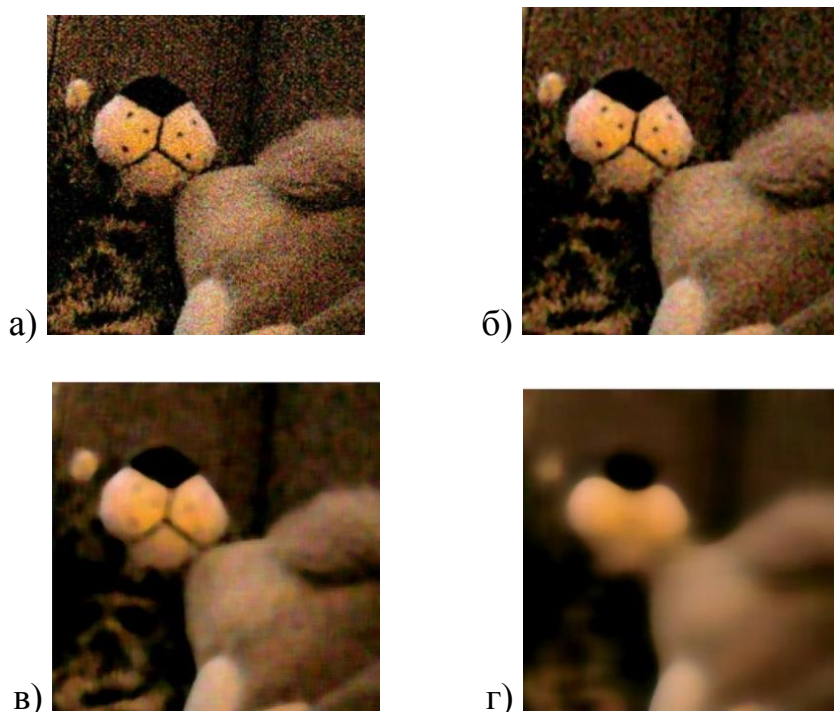


Рисунок 2.5 – Приклад використання медіанного фільтра до зашумленого зображення з трьома різними значеннями радіусу вікна фільтрації: а) оригінал зображення; б) радіус вікна дорівнює 1 пікселю; в) радіус вікна дорівнює 3 пікселі; г) радіус вікна дорівнює 5 пікселів [20]

Медіанний фільтр працює, переміщуючи зображення піксель за пікселем, замінюючи кожне значення середнім значенням сусідніх пікселів. Шаблон сусідів називається «вікном», яке ковзає піксель за пікселем по всьому зображенню. Медіана обчислюється шляхом сортування всіх значень пікселів із вікна за числовим порядком, а потім заміни пікселя, який розглядається, середнім (медіанним) значенням пікселя [20].

### 2.3 Опис обраних методів кластеризації зображення

В даній роботі використовується метод  $k$ -середніх ( $k$ -means) для кластеризації зображення.

Мета даного методу – розділити  $n$  спостережень на  $k$  кластерів таким чином, щоб кожне спостереження належало до кластера із найближчим до нього середнім значенням.

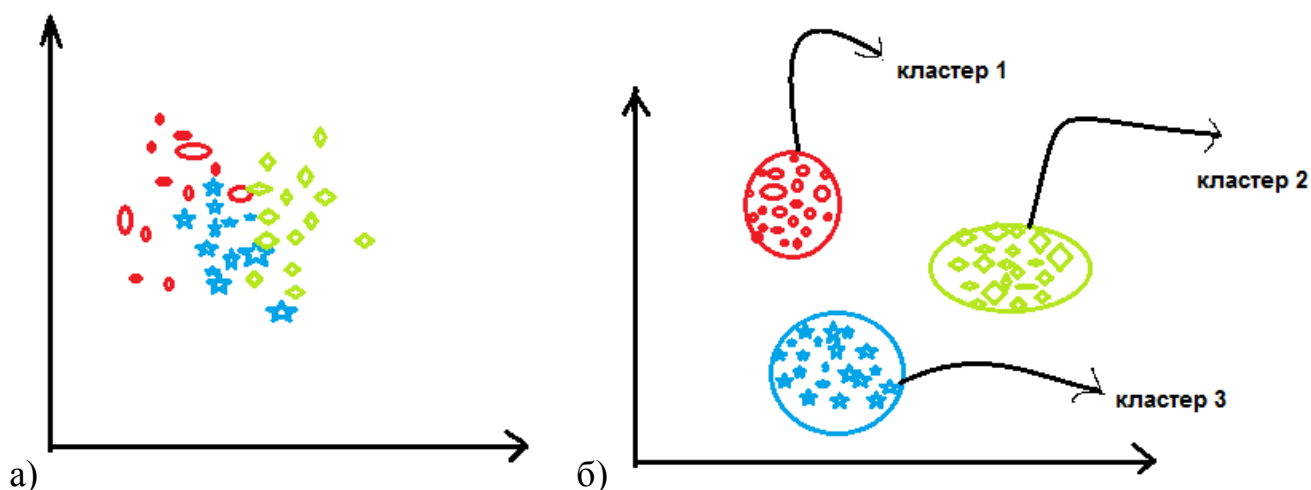


Рисунок 2.6 – Приклад роботи кластеризації методом  $k$ -means: а) дані до застосування алгоритму; б) дані після застосування алгоритму (рисунок виконано самостійно)

На рисунку 2.6 приклад роботи кластеризації методом  $k$ -means. На рисунку 2.6.а показано дані до застосування алгоритму кластеризації  $k$ -середніх. Тут усі три різні категорії переплутані. У реальному світі такі дані неможливо одразу

віднести до конкретної категорії. Рисунок 2.6.б демонструє дані після застосування алгоритму кластеризації  $k$ -середніх. Всі три різні елементи класифікуються за трьома різними категоріями, які називаються кластерами.

Даний метод заснований на мінімізації суми квадратів відстаней між центром його кластера та кожним спостереженням [21]. Така функція має наступний вигляд:

$$\sum_{i=1}^n d(x_i, m_j(x_i))^2$$

де  $d$  – метрика,  $x_i$  –  $i$ -ий об'єкт даних, а  $m_j(x_i)$  – центр кластера, якому на  $j$ -ій ітерації приписаний елемент  $x_i$ .

Принцип алгоритму – пошук таких наборів елементів та центрів кожного кластера при наявності деякої функції  $\Phi(\circ)$ , що виражає якість поточного розбиття множини на  $k$  кластерів. При цьому, сумарне квадратичне відхилення елементів від центрів цих кластерів повинно буде найменшим:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

де  $k$  – число кластерів,  $S_i$  – отримані кластери,  $i = 1, 2, \dots, k$ ,  $\mu_i$  – центри мас векторів  $x_j \in S_i$ .

На початковому етапі роботи алгоритму обираються центри кластерів довільним чином. Для кожного елемента множини ітеративно обраховується відстань від центрів з приєднанням кожного елемента до кластера з найближчим центром. Для кожного з кластерів обчислюються нові значення центрів, щоб мінімізувати функцію  $\Phi(\circ)$ . Після цього повторюється процедура перерозподілу елементів між кластерами [21].

Алгоритм роботи кластеризації методом k-means складається з наступних кроків:

- а) вибір кількості кластерів, які необхідно знайти, тобто  $k$ ;
- б) випадкове призначення точок даних будь-якому з  $k$  кластерів;
- в) обчислення центру кластерів;
- г) обчислення відстані точок даних від центрів кожного з кластерів;
- д) залежно від відстані кожної точки даних від кластера відбувається перепризначення точки даних найближчим кластерам;
- е) повторне обчислення нового центру кластера;
- ж) повторення кроків г, д, е, доки точки даних не змінять кластери або доки не буде досягнене призначеної кількості ітерацій.

Алгоритм кластеризації методом k-means має наступні переваги:

- він дуже простий у застосуванні;
- його можна масштабувати до великого набору даних;
- підтримує узагальнення кластерів різних форм і розмірів.

До недоліків даного методу можна віднести:

- результат класифікації залежить від початкових позицій центрів кластерів;
- дуже чутливий до викидів, що може призвести до викривлювання середнього;
- кількість кластерів має бути визначена дослідником заздалегідь [30].

#### 2.4 Опис обраних методів виявлення контурів

В даній роботі використано оператор Кенні для виявлення контурів. Даний метод розроблений у 1986 році Джоном Ф. Кенні. Він складається з 5 кроків:

- а) зменшення шуму;
- б) розрахунок градієнта;
- в) немаксимальне придушення;
- г) подвійна порогова обробка;

д) відстеження країв за гістерезисом.

Алгоритм заснований на зображеннях у відтінках сірого. Тому необхідною умовою є перетворення зображення в градації сірого перед виконанням вищезазначених кроків [22].

Один із способів позбутися шуму на зображенні – застосування розмиття за Гаусом, щоб згладити його. Для цього використовується техніка згортки зображення з ядром Гауса (3x3, 5x5, 7x7 тощо). Розмір ядра залежить від очікуваного ефекту розмиття (чим менше ядро, тим менш помітним є розмиття). Рівняння для ядра фільтра Гауса розміром  $(2k+1) \times (2k+1)$  задається наступним чином:

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k + 1)$$

Обчислення градієнта визначає інтенсивність і напрямки країв шляхом обчислення градієнта зображення за допомогою операторів визначення країв. Краї відповідають зміні інтенсивності пікселів. Найпростішим способом їх виявлення є застосування фільтрів, які висвітлюють цю зміну інтенсивності в горизонтальному (x) і вертикальному (y) напрямках. Коли зображення згладжене, похідні  $I_x$  та  $I_y$  відносно x і y обчислюються. Це можна зробити шляхом згортання I з ядрами Собеля  $K_x$  і  $K_y$  відповідно:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Потім обчислюється величина G і нахил  $\theta$  градієнта:

$$|G| = \sqrt{I_x^2 + I_y^2},$$

$$\theta(x, y) = \arctan\left(\frac{I_y}{I_x}\right)$$

В ідеалі кінцеве зображення повинно мати тонкі краї. Отже, потрібно виконати немаксимальне придушення, щоб зробити краї тоншими. Кожен піксель має 2 основні критерії (напрямок краю в радіанах та інтенсивність пікселя у межах від 0 до 255). На основі цих вхідних даних процес немаксимального придушення складається з наступних кроків:

- створити матрицю, ініціалізовану на 0, такого ж розміру, як вихідна матриця інтенсивності градієнта;
- визначити напрямок краю на основі значення кута з матриці кутів;
- перевірити, чи піксель у тому самому напрямку має вищу інтенсивність, ніж піксель, який зараз обробляється.

Крок подвійного порогоування спрямований на визначення 3 типів пікселів: сильних, слабких і нерелевантних:

- сильні пікселі – пікселі, які мають настільки високу інтенсивність, що можна бути впевненим, що вони сприяють остаточному краю;
- слабкі пікселі – пікселі, які мають недостатнє високе значення інтенсивності щоб вважатися сильними, але все ж недостатньо мале, щоб вважатися нерелевантними для виявлення краю;
- нерелевантними до краю вважаються інші пікселі.

Подвійне порогове значення означає, що:

- високий поріг використовується для ідентифікації сильних пікселів (інтенсивність вище верхнього порігу);
- низький поріг використовується для визначення нерелевантних пікселів (інтенсивність нижча за нижній поріг);
- усі пікселі, інтенсивність яких знаходиться між обома порогамі, позначаються як слабкі, а механізм гістерезису (наступний крок) допоможе ідентифікувати ті, які можна вважати сильними, і ті, які вважаються нерелевантними [22].

Гістерезис складається з перетворення слабких пікселів на сильні, виходячи з порогових результатів на минулому етапі.

На рисунку 2.7 наведено приклад роботи алгоритму виявлення контурів методом Кенні.



Рисунок 2.7 – Приклад роботи алгоритму виявлення контурів методом Кенні: а) оригінальне зображення; б) зображення після обробки [22]

## 2.5 Вибір технології розпізнавання об'єктів у режимі реального часу

Використовуючи метод лінійної згортки обрано технологію розпізнавання об'єктів у режимі реального часу, яка буде використовуватися для розробки алгоритму визначення розміру взуття за двовимірним зображенням.

Серед альтернатив виділено:

- технологія Vision (A1);
- технологія Core ML (A2);
- технологія OpenCV (A3).

Для задачі виділено критерії, які здійснюють найбільший вплив на очікуваний результат. Такими критеріями є:

- складність реалізації (K1), вага критерію = 0,2;
- сучасність (K2), вага критерію = 0,4;
- здатність до змін (K3), вага критерію = 0,1;
- швидкість роботи (K4), вага критерію = 0,3.

Графічне зображення дерева альтернатив відображено на рисунку 2.8.

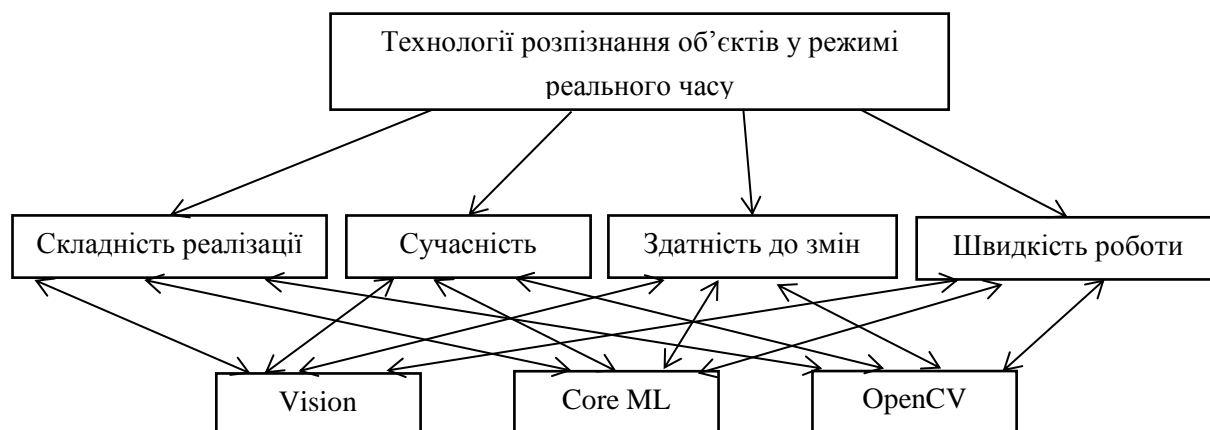


Рисунок 2.8 – Дерево альтернатив (рисунок виконано самостійно)

Матриця попарних порівнянь критеріїв наведено у таблиці 2.1.

Таблиця 2.1 – Матриця попарних порівнянь критеріїв (таблиця виконана самостійно)

	К1	К2	К3	К4
A1	просто	оновлюється раз у місяць	немає	достатньо швидко
A2	складно	оновлюється раз на рік	є	повільно
A3	просто	оновлюється раз на квартал	є	швидко
w	0,2	0,4	0,1	0,3

Далі проведено нормалізацію значень критеріїв, результат який наведено у таблиці 2.2.

Таблиця 2.2 – Нормалізація значень критеріїв (таблиця виконана самостійно)

	К1	К2	К3	К4
A1	1	1	0	0,75
A2	0	0	1	0
A3	1	0,75	1	1
w	0,2	0,4	0,1	0,3

Вирішення багатокритеріальної задачі вибори методом лінійної згортки передбачає вибір альтернативи  $a_i \in A$ , так щоб максимізувати наступний вираз:

$$\sum_{j=1}^m w_j X_j(a_i)$$

Тоді для альтернативи 1:  $1*0,2 + 1*0,4 + 0*0,1 + 0,75*0,3 = 0,825$ .

Для альтернативи 2:  $0*0,2 + 0*0,4 + 1*0,1 + 0*0,3 = 0,1$

Для альтернативи 3:  $1*0,2 + 0,75*0,4 + 1*0,1 + 1*0,3 = 0,9$

Результати обчислень для вибору кращої альтернативи зведено у таблицю 2.3. У результаті найкращою технологією розпізнавання об'єктів у режимі реального часу для вирішення завдання визначення розміру взуття за фотографією обведеної стопи є OpenCV (альтернатива 3).

Таблиця 2.3 – Результати обчислень для вибору кращої альтернативи (таблиця виконана самостійно)

	K1	K2	K3	K4	Σ
A1	0,2	0,4	0	0,225	0,825
A2	0	0	0,1	0	0,1
A3	0,2	0,3	0,1	0,3	0,9

Отже, OpenCV (Open Source Computer Vision Library) – це бібліотека на мові Python, яка дозволяє виконувати завдання з обробки зображень та комп'ютерного зору. Вона надає широкий спектр можливостей, включаючи виявлення об'єктів, розпізнавання облич та відстеження.

Ця бібліотека базується на оптимізованому C/C++ і підтримує Java та Python разом з C++ через інтерфейс. Бібліотека має більше 2500 оптимізованих алгоритмів, включаючи велику колекцію алгоритмів комп'ютерного зору і машинного навчання, як класичних, так і найсучасніших.

За допомогою OpenCV стає легко виконувати складні завдання, такі як ідентифікація і розпізнавання осіб, ідентифікація об'єктів, класифікація дій людини на відео, відстеження руху камери, відстеження рухомих об'єктів, витяг 3D-моделей об'єктів, генерація 3D хмар точок зі стереокамер, об'єднання

зображень для отримання цілої сцени з високою роздільною здатністю і багато іншого [24].

До основних функцій OpenCV належать:

- робота із структурами даних – OpenCV використовує матриці та діапазони, вектори та скаляри для зберігання та роботи із зображеннями; дозволяють орієнтуватися на зображення, проводити математичні перетворення та виконувати інші дії;

- видозміна зображень – за допомогою OpenCV можна виконувати попередню підготовку картини перед її розшифровкою, наприклад: обрізати, виконувати збільшення чи зменшення, обертати;

- додавання ефектів – картинку можна зробити у відтінках сірого або чорно-білою; це важливо для алгоритмів розпізнавання, які працюють із знебарвленими зображеннями; можна змінювати тон кольору, розмивати, згладжувати або геометрично змінювати картинку.

- малювання поверх зображення – на картинку можна нанести лінії та геометричні фігури, зробити підпис, наприклад, щоб виділити знайдене програмою обличчя;

- розпізнавання об'єктів – OpenCV використовує контури об'єктів, сегментацію за кольорами, вбудовані методи розпізнавання, які можна налаштувати залежно від об'єкта і чутливості алгоритму;

- робота з відеороликами – зчитування роликів з використанням кодеків, аналізувати, що в них відбувається та відстежувати рухи і елементи. Це особливо корисно при створенні програмного забезпечення для камери відеоспостереження або програмуванні робота, що рухається [24].

## 2.6 Розробка плану розробки алгоритму

Процес розробки інформаційної технології визначення розміру стопи та взуття включає три етапи:

- реалізація сканера аркуша паперу з фотографії;

- сегментація зображення на основі кольорів;
- вимірювання розміру стопи та визначення розміру взуття.

На етапі реалізації сканера аркуша паперу з фотографії повинні виконуватися наступні кроки:

- а) завантаження фото;
- б) передобробка зображення:
  - 1) конвертування зображення з колірного простору BGR у відтінки сірого (Grayscale);
  - 2) використання розмиття по Гаусу для видалення гаусівського шуму із зображення;
- в) виявлення контурів аркуша паперу методом Кенні;
- г) позначення країв аркуша паперу точками;
- д) вилучення аркуша паперу з фотографії за знайденими точками;
- е) постобробка зображення (перетворення вилученого зображення у скан):
  - 1) корекція кольору зображення з використанням методу балансування білого;
  - 2) нормалізація зображення (зміна інтенсивності пікселів зображення);
  - 3) конвертування зображення з колірного простору BGR у відтінки сірого (Grayscale);
  - 4) застосування гамма-корекції;
  - 5) застосування адаптивного порогу.

Процес сегментації зображення на основі кольорів передбачає виконання наступних кроків:

- конвертація зображення з колірного простору RGB в HSV;
- застосування гаусівського розмиття;
- кластеризація зображення за допомогою методу k-means;
- конвертація результату з HSV в BGR.

Етап вимірювання розміру стопи та визначення розміру взуття складається з наступних кроків:

- а) попередня обробка зображення:
  - 1) застосування медіанної фільтрації;
  - 2) конвертування зображення з колірного простору BGR у відтінки сірого (Grayscale);
  - 3) застосування порогування Оцу;
- б) використання заливки для визначення зовнішньої форми об'єкта;
- в) об'єднання залитого зображення з контуром об'єкта;
- г) інвертування кольорів;
- д) обведення контурів об'єкта;
- е) знаходження довжини стопи.
- ж) знаючи розмір листа А4, за допомогою пропорції, розраховується довжина стопи у см та визначається розмір взуття за європейським стандартом.

### 3 РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

#### 3.1 Типовий опис даних

Типові приклади зображень наведено на рисунку 3.1.

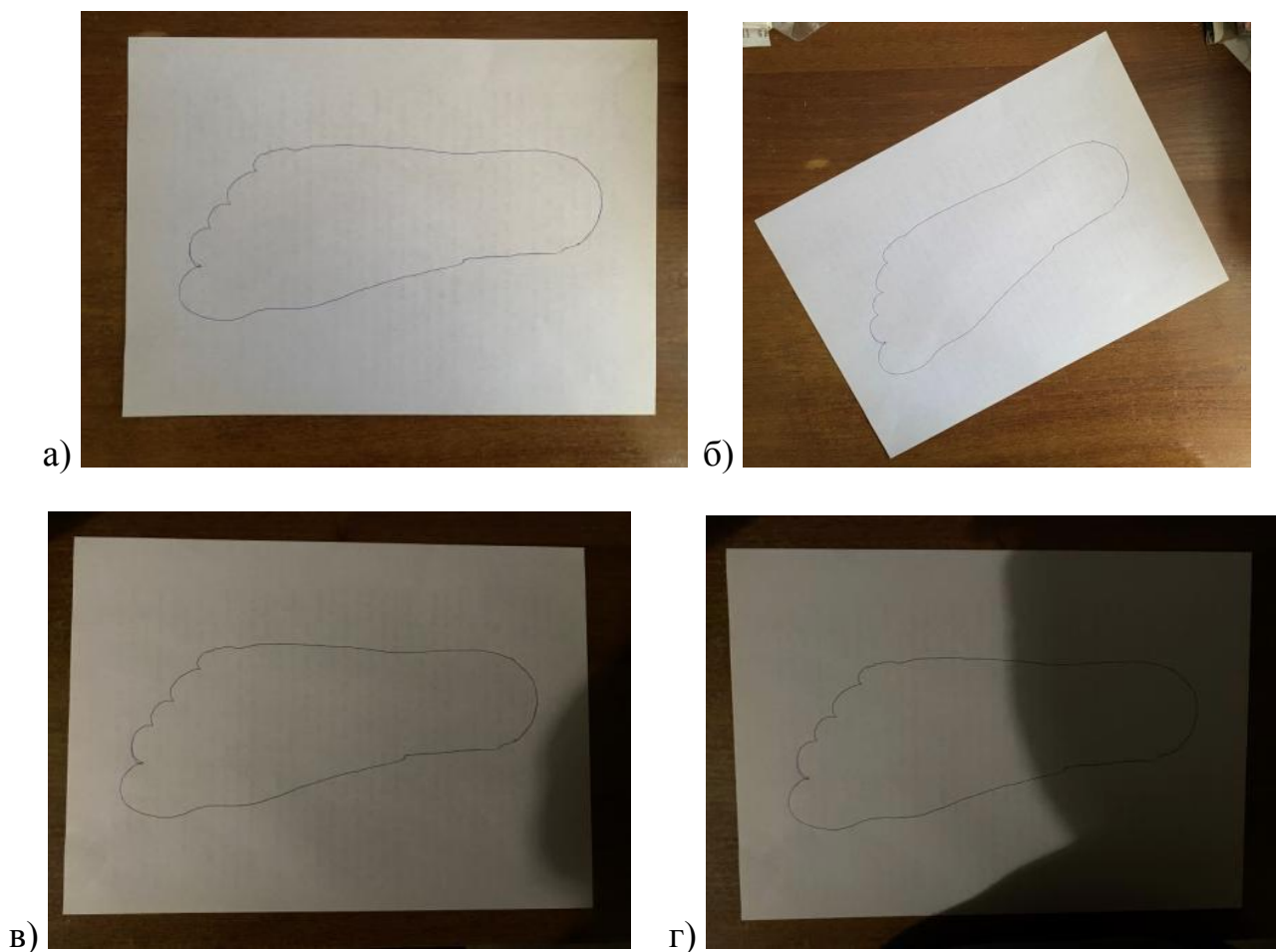


Рисунок 3.1 – Приклади зображень розглянутого набору даних: а) фото без дефектів (добре освітлення, немає тіней, майже відсутні дефекти трансформації); б) фото з добрим освітленням, але з дефектом трансформації; в) приклад фотографії при вечірньому освітленні з наявністю тіні на аркуші паперу; г) фотографія при слабкому освітленні та наявністю тіні, що перекриває майже половину аркуша паперу (усі фото зроблені самостійно)

Усі фотографії вони мають один або декілька дефектів, а саме:

- фото зняті при різних кутах та поворотах;
- фото зняті при слабкому освітленні;
- на деяких фото виявлено тіні.

### 3.2 Реалізація інформаційної технології визначення розміру взуття

Використовуючи можливості бібліотеки OpenCV, розроблено алгоритму згідно плану, описаного у розділі 2.6.

#### 3.2.1 Реалізація сканера аркуша паперу з фотографії

Першим етапом реалізації інформаційної технології визначення висоти стопи та розміру взуття є реалізація сканера аркуша паперу з фотографії. Для цього виконано ряд дій.

Спочатку виконується завантаження фото та зміна розміру фото до 500 пікселів за висотою із збереженням співвідношення сторін. Далі, здійснюється попередня обробка зображення, що включає в себе:

- конвертування зображення з колірного простору BGR у відтінки сірого (Grayscale);
- використання розмиття по Гаусу для видалення гаусівського шуму із зображення;
- виявлення країв аркуша паперу методом Кенні.

Результат такої передобробки зображено на рисунку 3.2.

На наступному етапі виконується знаходження всіх контурів на зображенні, а саме пошук контуру з чотирма сторонами за допомогою функції `approxPolyDP()`, позначення кутів аркуша паперу точками. Дана функція використовує алгоритм Рамера-Дугласа-Пекера. Результат знаходження контурів подано на рисунку 3.3.

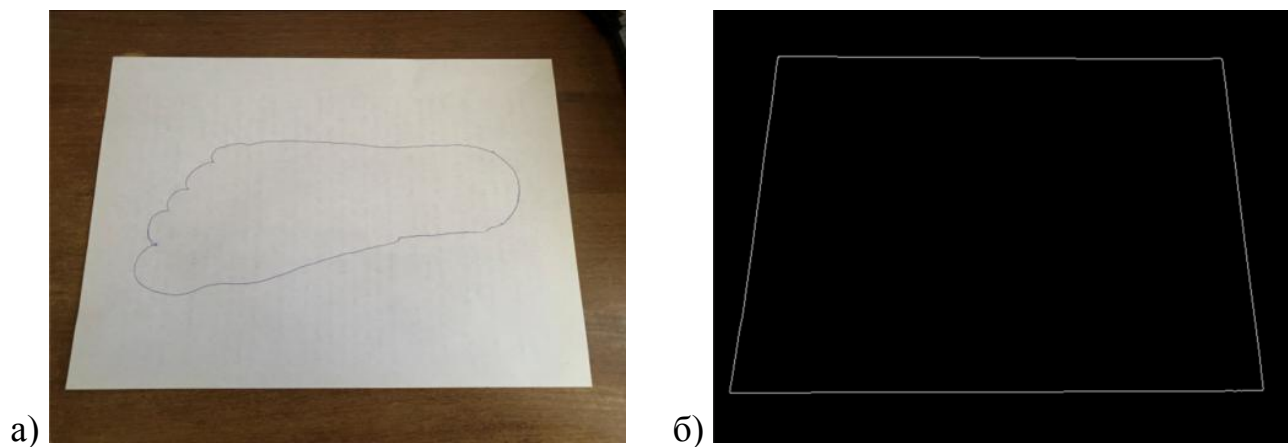


Рисунок 3.2 – Попередня обробка зображення для знаходження контурів:  
а) оригінал фото; б) результат після передобробки (рисунок виконано самостійно)

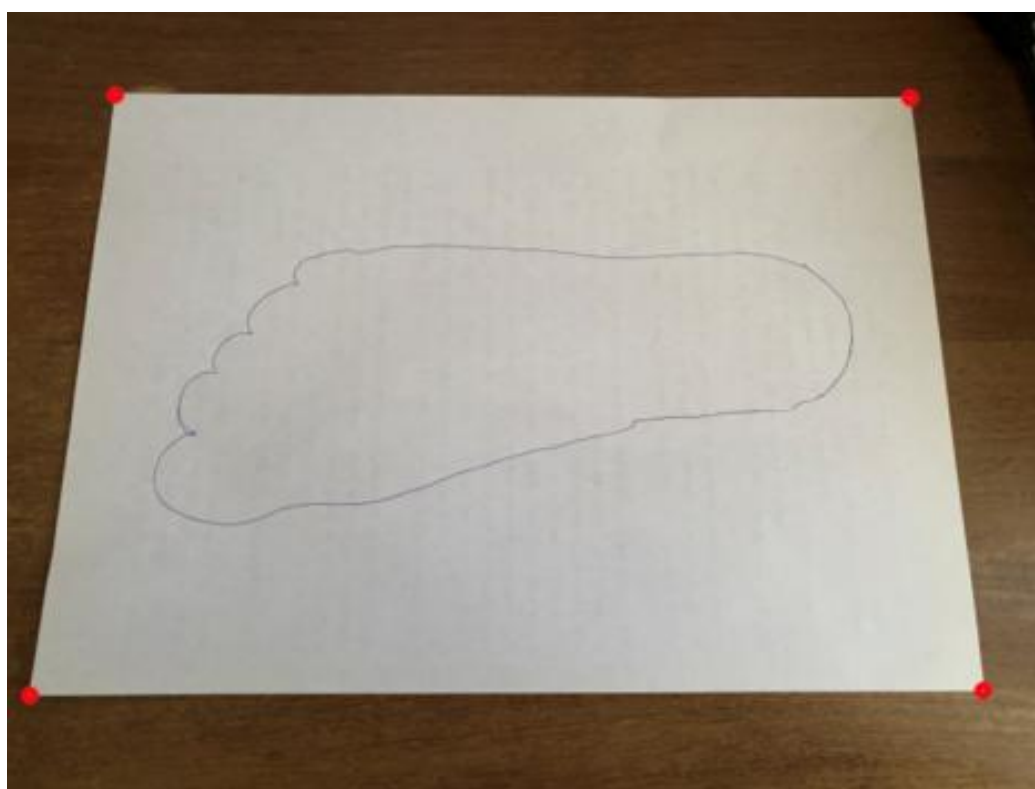


Рисунок 3.3 – Результат знаходження контурів аркуша паперу на фото, позначення кутів червоними точками (рисунок виконано самостійно)

Далі, застосується чотириточкове перетворення зображення для вилучення аркуша паперу з фото. Для цього відбувається витяг прямокутника, утворений чотирма точками на минулому етапі (рисунок 3.4).

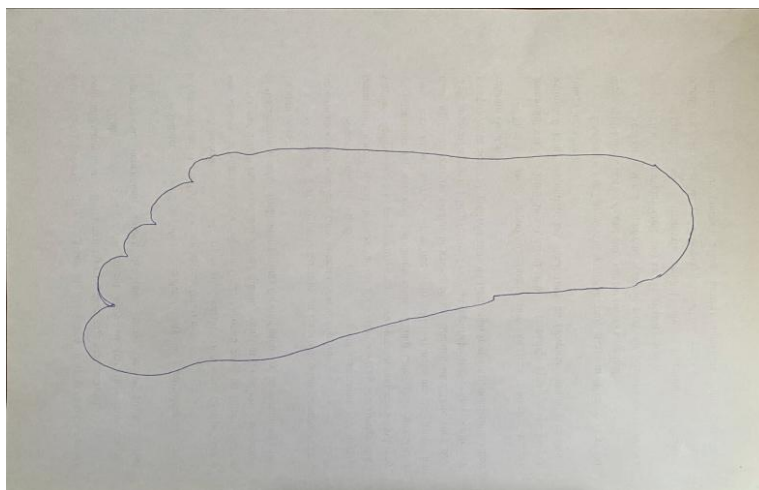


Рисунок 3.4 – Результат вилучення аркуша паперу з фото (рисунок виконано самостійно)

Наступним великим кроком стала постобробка зображення. На даному етапі передбачено вилучення тіней з зображення та перетворення вилученого зображення у скан.

Спочатку відбувається корекція кольору зображення з використанням методу балансування білого. Метою цього алгоритму є нормалізація кожного каналу зображення до певного значення з цього каналу. Нормалізація зображення відбувається за допомогою певних перцентильних значень у кожному каналі [23]. Припустивши, що біла пляма встановлена на рівні 99% перцентилі значень інтенсивності зображення, можна побачити, що отримане зображення має менш сірий відтінок (рисунок 3.5).

Далі застосовано методи розширення (оператор морфологічного перетворення, який використовується для збільшення розміру або товщини об'єкта переднього плану на зображенні) [25] та медіанне розмивання для придушення контура стопи. Це дає досить хороше фонове зображення, яке містить усі тіні та/або знебарвлення. Далі, обчислюється різниця між оригіналом і фоном, який отримано та виконується інвертування кольорів. Застосовано нормалізацію зображення (зміна інтенсивності пікселів). Результат такої обробки зображено на рисунку 3.6.

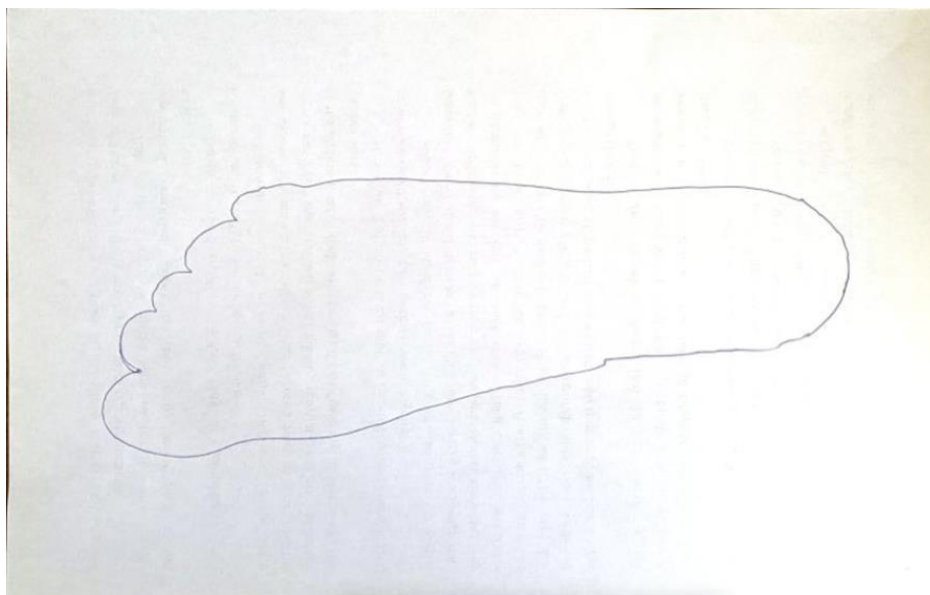


Рисунок 3.5 – Результат застосування методу балансування білого (рисунок виконано самостійно)

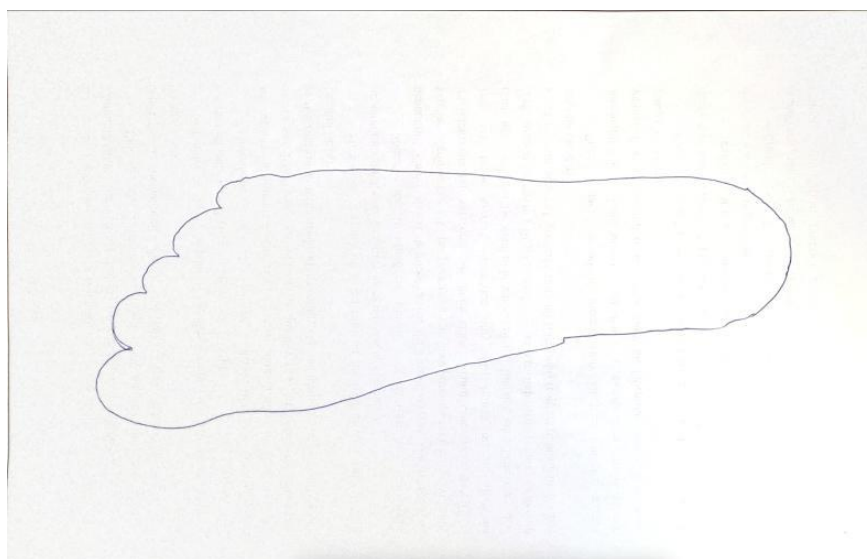


Рисунок 3.6 – Результат нормалізації зображення (рисунок виконано самостійно)

Згодом виконано конвертування зображення з колірному простору BGR у відтінки сірого (Grayscale) та застосування гамма-корекції. Гамма-корекція також відома як перетворення степеневого закону. Це означає, що інтенсивність пікселів зображення може бути масштабована від діапазону  $[0, 255]$  до  $[0, 1,0]$ . Звідти виходить гамма-кориговане зображення, застосовуючи таке рівняння:

$$O = I^{\frac{1}{G}},$$

де  $I$  – вхідне зображення,  $G$  – гамма значення.

Потім вихідне зображення  $O$  масштабується до діапазону  $[0, 255]$ . Значення гамми  $< 1$  зсунуть зображення до темнішого краю спектра, а значення гамми  $> 1$  зроблять зображення світлішим [26]. Результат такої обробки зображено на рисунку 3.7.

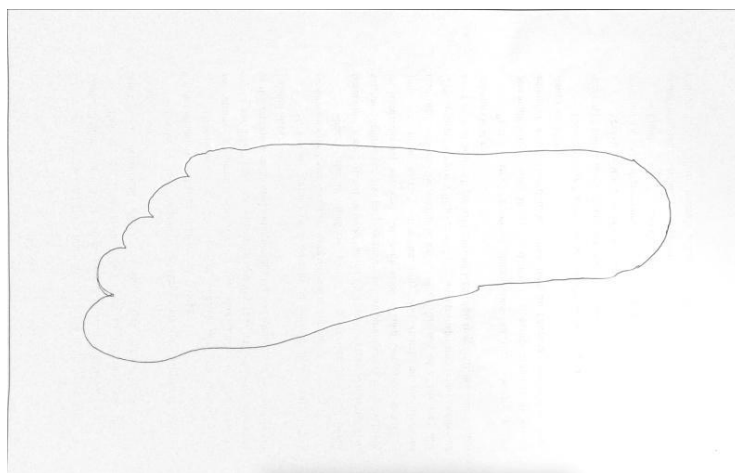


Рисунок 3.7 – Результат застосування гамма-корекції (рисунок виконано самостійно)

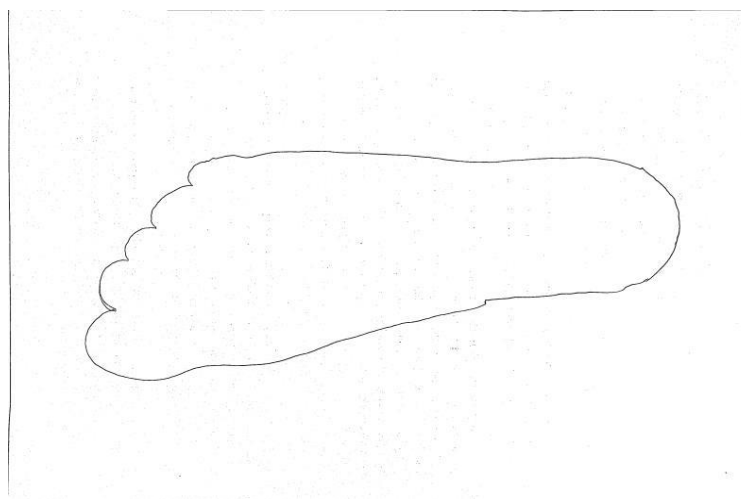


Рисунок 3.8 – Результат застосування адаптивного порогоування (фінальний результат) (рисунок виконано самостійно)

На останньому етапі застосовується адаптивний поріг (Adaptive Threshold), щоб надати повноцінного вигляду скана документа (рисунок 3.8).

Даний метод сканування чудово працює з фотографіями, зроблені при доброму освітленні, але при поганому у більшості випадків можна отримати помилку `UnboundLocalError` на етапі виявлення країв аркуша паперу методом Кенні. У такому випадку процес попередньої обробки зображення включає наступні кроки:

- корекція кольору з використанням методу балансування білого;
- конвертація фото з RGB в HSV;
- нормалізація зображення;
- конвертація фото з HSV у Grayscale;
- застосування методу Adaptive Threshold;
- використання Gaussian Blur для видалення гаусівського шуму з зображення.

Результат такої попередньої обробки зображення подано на рисунку 3.9, а рисунок 3.10 демонструє результат знаходження контурів аркуша паперу та позначення кутів червоними точками.

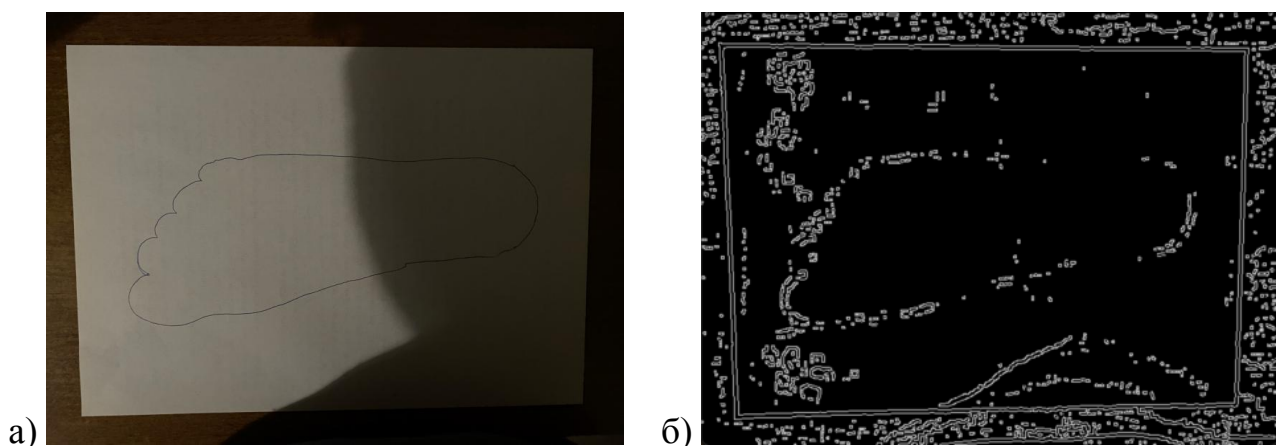


Рисунок 3.9 – Попередня обробка зображення зі слабким освітленням для знаходження контурів: а) оригінал фото; б) результат після передобробки (рисунок виконано самостійно)

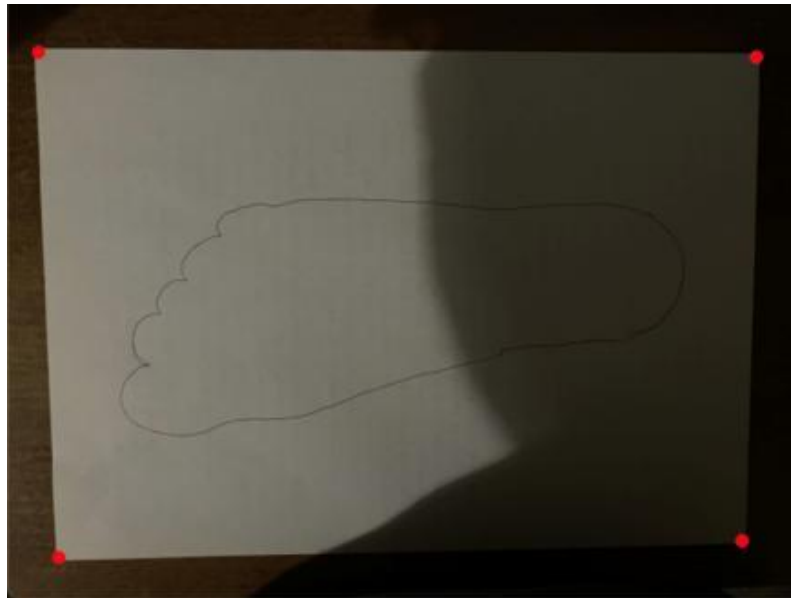


Рисунок 3.10 – Результат знаходження контурів аркуша паперу на фото зі слабким освітленням, позначення кутів червоними точками (рисунок виконано самостійно)

### 3.2.2 Сегментація зображення на основі кольорів

Другим етапом реалізації інформаційної технології визначення висоти стопи та розміру взуття стало сегментація зображення на основі кольорів.

У комп'ютерному зорі сегментація зображення – це процес поділу зображення на кілька сегментів. Мета сегментації зображення полягає в тому, щоб змінити подання зображення на щось, що є більш значущим і легшим для аналізу [27]. У рамках даного дослідження це допоможе відділити кольори аркуша паперу та контура стопи.

Перед початком сегментації виконано попередню обробку зображення, а саме:

- конвертація зображення у колірну схему HSV;
- застосування розмиття Гауса.

Великою причиною конвертації зображення у колірну схему HSV є те, що вона ідеально підходить для сегментації зображення на основі кольорів. Колірний простір HSV складається з 3 матриць: «тон» («hue»), «насиченість» («saturation»), і «значення» («value»). У OpenCV діапазон значень для «тону», «насиченості» та

«значення» становить відповідно 0-179, 0-255 і 0-255. «Тон» представляє колір (тобто, є сумішню всіх каналів RGB і краще розрізняє всі окремі кольори), «насиченість» представляє кількість, до якої відповідний колір змішується з білим, а «значення» представляє кількість, до якої відповідний колір змішується з чорним [28]. Результат такої обробки зображено на рисунку 3.11.

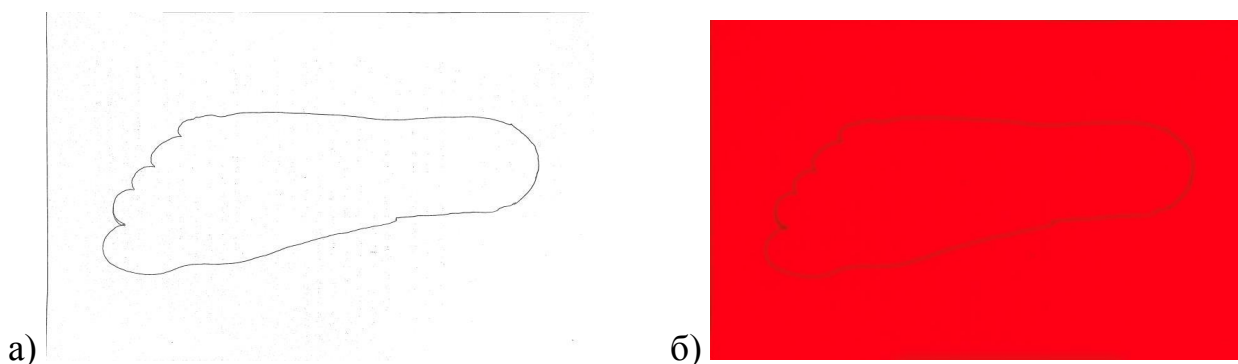


Рисунок 3.11 – Попередня обробка зображення для застосування сегментації на основі кольорів: а) скан аркуша паперу до обробки; б) результат після обробки (рисунок виконано самостійно)



Рисунок 3.12 – Результат сегментації зображення на основі кольорів за допомогою методу кластеризації k-means (кількість кластерів дорівнює 2) (рисунок виконано самостійно)

Сегментація зображення на основі кольорів відбувається за допомогою методу кластеризації  $k$ -means. Результат кластеризації цим методом відображено на рисунку 3.12. Кількість кластерів (значення  $k$ ) дорівнює 2.

На останньому етапі виконано конвертування зображення з колірної схеми HSV у BGR (рисунок 3.13). Тепер можна побачити, як на чисто-білому аркуші паперу зображено темно-сірий контур стопи.



Рисунок 3.13 – Фінальний результат сегментації зображення на основі кольорів (рисунок виконано самостійно)

### 3.2.3 Вимірювання розміру стопи

Для визначення розміру взуття береться зображення після кластеризації методом  $k$ -means (приклад – рисунок 3.13).

Спочатку на цьому етапі виконується попередня обробка зображення, що включає застосування медіанного розмиття та застосування порогування Оцу. Результат такої обробки подано на рисунку 3.14.

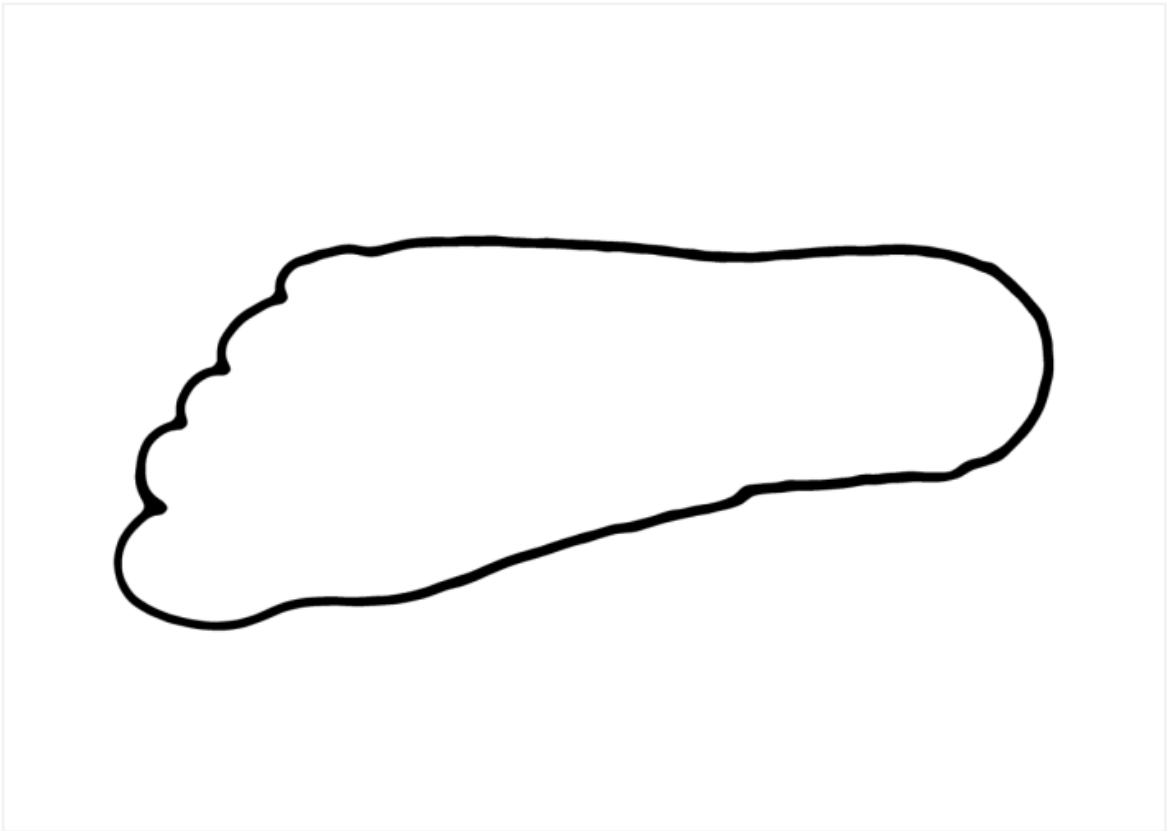


Рисунок 3.14 – Результат попередньої обробки зображення для визначення розміру взуття (рисунок виконано самостійно)

Далі, здійснюється заливка для визначення зовнішньої форми об'єкта функцією `floodFill` з бібліотеки `OpenCV`. Цей алгоритм, по суті, шукає сусідів з однаковим кольором, використовуючи початковий піксель як еталон. Якщо знаходиться колір як параметр, функція змусить усі пікселі, знайдені пошуком, мати цей колір [29]. `floodFill` запитує зображення, маску (не використовується = немає), вихідний піксель і колір для фарбування початкового елемента та подібних сусідів. Маючи це під рукою, потрібно лише йти піксель за пікселем, перевіряючи його колір, якщо він білий, ми змінюємо якийсь сірий колір, а потім збільшуємо сірий колір, щоб наступний об'єкт мав інший сірий тон [2]. Результат застосування `floodFill` зображено на рисунку 3.15.

Після застосування функції `floodFill` виконано об'єднання залитого зображення з оригінальним зображенням (рисунок 3.16).

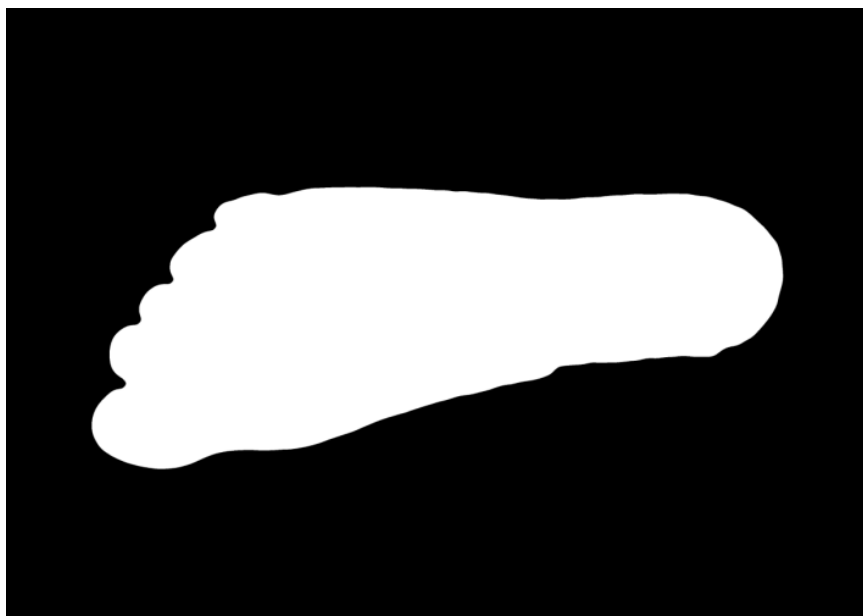


Рисунок 3.15 – Застосування floodFill до зображення (рисунок виконано самостійно)

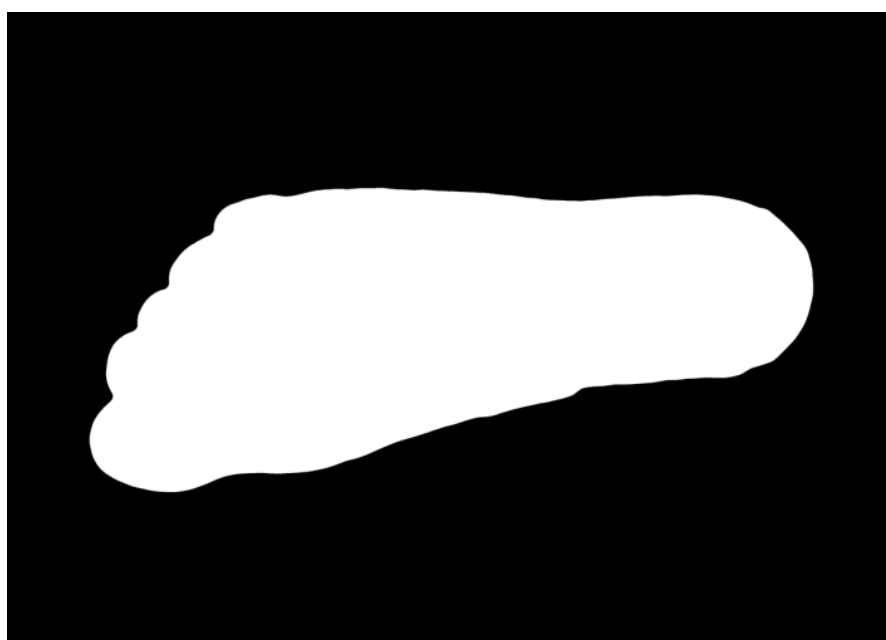


Рисунок 3.16 – Об'єднання залитого зображення з оригінальним зображенням (рисунок виконано самостійно)

На наступному кроці відбувається інвертування кольорів, пошук та обведення контурів об'єкта за допомогою функції `boundingRect`. Результат обведення контурів стопи зображено на рисунку 3.17.



Рисунок 3.17 – Обведення контурів стопи (рисунок виконано самостійно)

На останньому етапі здійснюється вилучення ширини `boundingRect` об'єкта. Знаючи розмір листа А4 (см), ширину зображення і ширину об'єкта, за допомогою пропорції, знайдено ширину стопи в см (рисунок 18). Використовуючи таблицю 3.1, визначено розмір взуття.

Таблиця 3.1 – Таблиця розмірів взуття за європейським стандартом (таблиця виконана самостійно)

Діапазон у см	Розмір в EU
22.1 — 22.5	35.5
22.5 — 22.9	36
22.9 — 23.3	37
23.3 — 23.7	37.5
23.7 — 24.1	38
24.1 — 24.8	39
24.8 — 25.1	39.5

## Продовження таблиці 3.1

25.1 — 25.4	40
25.4 — 25.8	40.5
25.8 — 26	41
26 — 26.7	41.5
26.7 — 27.3	42
27.3 — 27.7	43
27.7 — 27.9	43.5
27.9 — 28.6	44
28.6 — 29.2	44.5
29.2 — 29.5	45
29.5 — 29.8	46

```
MacBook-Pro-Ivan:Diploma ivansakharov$ /usr/local/bin/python3.9 /  
paper width = 3773  
bounding box width = 3048  
feet width in cm = 23.993002915451893  
Size = EU 38
```

Рисунок 3.18 – Результат вимірювання стопи та визначення розміру взуття  
(рисунок виконано самостійно)

## 4 ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ

Типові приклади зображень наведено на рисунку 3.1.

Проведено серію експериментів, за результатами яких сформовано алгоритм вимірювання стопи та визначення розміру взуття. На рисунку 4.1 можна побачити ці результати. В даному випадку проведено 100 вимірювань, тобто використано 100 різних фотографій одного аркуша паперу формату А4, на якому відображено контур стопи дорослої людини.

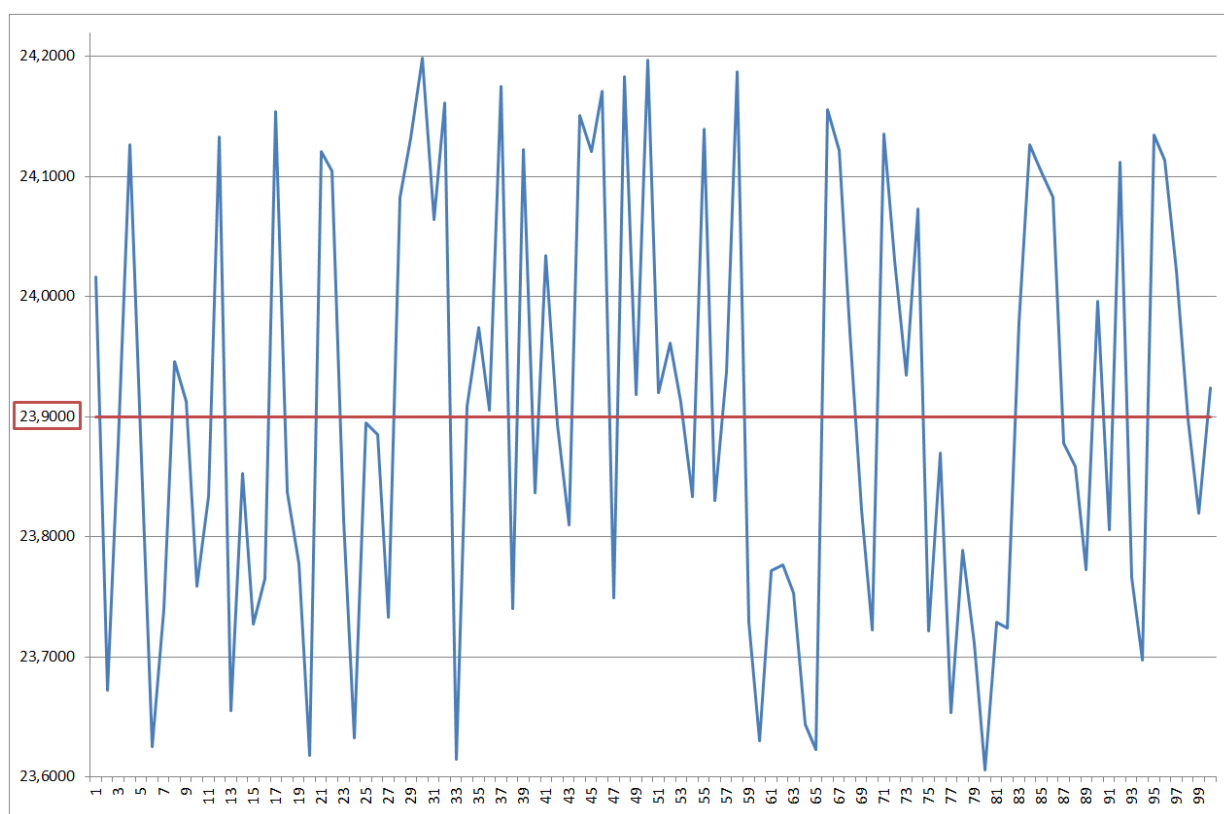


Рисунок 4.1 – Результати вимірювань довжини стопи (кількість вимірювань дорівнює 100) (рисунок виконано самостійно)

Реальна довжина стопи становить 23,9 см (на графіку це позначено червоною лінією). На основі цих вимірювань, можна зробити висновки, що якість фото та наявність дефектів впливають на результат визначення висоти стопи. Так, погане освітлення та наявність тіні, в залежності від розміру, може давати невелику похибку. Найбільша похибка становить  $\pm 0,3$  см.

У результаті, можна виділити наступні обмеження щодо фотографій:

- аркуш папер повиний бути рівний, не м'ятий;
- поверхня повинна бути контрастною, щоб не виникло помилки на етапі сегментації білого аркуша паперу;
- поверхня повинна бути добре освітленою;
- відсутність тіней або мала тінь, що не перекриває кут аркуша паперу;
- ступня не повинна бути намальована під кутом, максимально рівно.

На рисунку 4.2 подано поганий приклад фото аркуша паперу та до яких результатів це призводить.

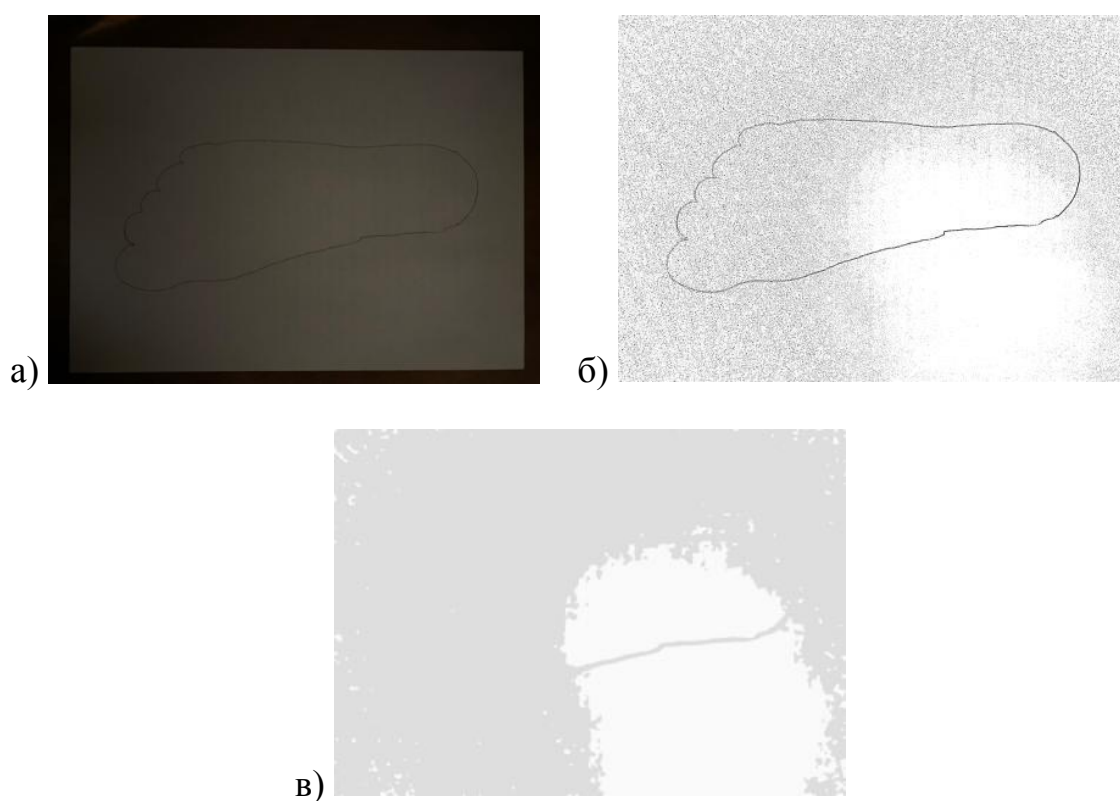


Рисунок 4.2 – Приклад поганого фото аркуша паперу: а) вхідне фото; б) результат перетворення фотографії на скан аркуша паперу; в) результат сегментації зображення (рисунок виконано самостійно)

## ВИСНОВКИ

У процесі виконання роботи проведено аналіз предметної області, що включає огляд існуючих досліджень визначення розміру взуття за двовимірним зображенням.

На основі схожих робіт, розроблено покращений алгоритм визначення висоти стопи та розміру взуття. Він передбачає від користувача тільки обвести стопу на аркуші паперу формату А4. Алгоритм розроблено на мові Python, з використанням можливостей бібліотеки OpenCV.

Процес розробки інформаційної технології включав три етапи. На першому етапі, використовуючи методи виділення контурів Кенні, адаптивного порогування, розмиття за Гаусом, реалізовано сканер, що перетворює фотографію на скан документа. Далі виконано сегментацію скана на основі кольорів за допомогою методу кластеризації k-means. І на останньому етапі, використовуючи методи медіанного розмиття порогування Оцу, визначено довжину стопи.

У якості набору даних використано реальні фото аркуша паперу формату А4, на якому зображено контури стопи, зняті у різних умовах та мають дефекти освітлення та трансформації. Проведено серію експериментів, виходячи з яких сформовано обмеження щодо фотографій.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

### Видання 1-3 авторів

1. J. Solem. Programming Computer Vision with Python: Tools and algorithms for analyzing images. O'Reilly Media, Inc., 2012. 264 p.
2. Gary Bradski, Adrian Kaehler, "Learning OpenCV. Computer vision with the OpenCV library, " O'Reilly Media; 1st edition (February 7, 2017), 1022 pages.

### Багаточастинний документ

3. Методичні вказівки до виконання кваліфікаційної роботи магістра за спеціальністю 121 – Інженерія програмного забезпечення (освітньо-наукова програма – Інженерія програмного забезпечення) для усіх форм навчання /Упор.: З.В. Дудар, В.В. Голян, В.І Каук, В.Ю. Нечволод, І.А. Ревенчук – Харків: ХНУРЕ, 2022 – 81с.

### Статті, конференції, тези доповідей

4. M. A. Shahid, M. A. Iftikhar, Z. A. Gondal, M. Adnan and S. Rathore, "Object Size Measurement through Images: An Application to Measuring Human Foot Size," 2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2018, pp. 298-302, doi: 10.1109/FIT.2018.00059.
5. Huawen Wang, Fuchang Liu and Ran Fan, "A research on foot size measurement algorithm based on image," 2021 Journal of Physics: Conference Series, Volume 1903 012004, 2021 International Conference on Applied Mathematics, Modelling and Intelligent Computing (CAMMIC 2021) 26-28 March 2021, Guilin, China
6. Sachin Parihar, S Ayushman, Rohan Deshpande, Shantanu Pal, Sourabh Yadav, Prof. Akshita Sharma "Document Scanner Application Using Python," 2020 International Research Journal of Modernization in Engineering Technology and Science, Volume:02/Issue:05/May-2020, e-ISSN: 2582-5208
7. K. Smelyakov, A. Chupryna, M. Hvozdiev and D. Sandrkin, "Gradational Correction Models Efficiency Analysis of Low-Light Digital Image," 2019 Open

Conference of Electrical, Electronic and Information Sciences (eStream), 2019, pp. 1-6, doi: 10.1109/eStream.2019.8732174.

8. K. Smelyakov, M. Hvozdiev, A. Chupryna, D. Sandrkin and V. Martovytskyi, "Comparative Efficiency Analysis of Gradational Correction Models of Highly Lighted Image," 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), 2019, pp. 703-708, doi: 10.1109/PICST47496.2019.9061356.

9. D. Ji, Y. Liu and C. Wang, "Research on Image Edge Detection Based on Improved Canny Operator," 2022 3rd International Conference on Information Science, Parallel and Distributed Systems (ISPDS), Guangzhou, China, 2022, pp. 229-232, doi: 10.1109/ISPDS56360.2022.9874064.

10. M. T. Hasan, T. Ahmed, R. R. Paul, M. A. Alam and R. Rab, "A Comprehensive Approach to Enhance Dark Image Implementing Image Processing Techniques," 2022 International Conference on Smart Information Systems and Technologies (SIST), Nur-Sultan, Kazakhstan, 2022, pp. 1-6, doi: 10.1109/SIST54437.2022.9945730.

11. T. Akazawa, Y. Kinoshita and H. Kiya, "Multi-Color Balance For Color Constancy," 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2021, pp. 1369-1373, doi: 10.1109/ICIP42928.2021.9506528.

12. J. -R. Wang and Y. -Y. Chuang, "Shadow Removal of Text Document Images by Estimating Local and Global Background Colors," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 1534-1538, doi: 10.1109/ICASSP40776.2020.9053378.

13. M. Alotaibi, A. Buccini and L. Reichel, "Restoration of Blurred Images Corrupted by Impulse Noise via Median Filters and  $\ell_p$ - $\ell_q$  Minimization," 2021 21st International Conference on Computational Science and Its Applications (ICCSA), Cagliari, Italy, 2021, pp. 112-122, doi: 10.1109/ICCSA54496.2021.00025.

14. R. Liang, L. Han, P. Liu and M. Liu, "Research on Improved K-means Image Segmentation Algorithm Based on HSV Space," 2022 International Conference on

Image Processing, Computer Vision and Machine Learning (ICICML), Xi'an, China, 2022, pp. 116-122, doi: 10.1109/ICICML57342.2022.10009682.

15. I. Sakharov, K. Smelyakov, O. Bohomolov, "Research and Development of Information Technology for Determining Shoe Size by Image," 2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2023, pp. 1-6.

*Електронний ресурс*

16. Otsu's method for image thresholding explained and implemented. URL: <https://muthu.co/otsus-method-for-image-thresholding-explained-and-implemented/> (дата звернення: 30.01.2023).

17. Adaptive Thresholding with OpenCV (cv2.adaptiveThreshold). URL: <https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/> (дата звернення: 30.01.2023).

18. Adaptive Thresholding. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm> (дата звернення: 30.01.2023).

19. Image Filters: Gaussian Blur. URL: <https://aryamansharda.medium.com/image-filters-gaussian-blur-eb36db6781b1> (дата звернення: 31.01.2023).

20. Image Filtering Median Filtering. URL: <https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Image%20Filtering.pdf> (дата звернення: 01.02.2023).

21. Image Segmentation using K Means Clustering. URL: <https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/> (дата звернення: 01.02.2023).

22. Canny Edge Detection Step by Step in Python — Computer Vision. URL: <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123> (дата звернення: 02.02.2023).

23. Image Processing with Python: Color Correction using White Balancing Methods. URL: <https://jmanansala.medium.com/image-processing-with-python-color-correction-using-white-balancing-6c6c749886de> (дата звернення: 02.02.2023).

24. OpenCV documentation. URL: <https://docs.opencv.org/4.x/d1/dfb/intro.html> (дата звернення: 02.02.2023).

25. Explain Dilation of an image How to dilate an image in OpenCV. URL: <https://www.projectpro.io/recipes/what-is-dilation-of-image-dilate-image-opencv> (дата звернення: 02.02.2023).

26. OpenCV Gamma Correction. URL: <https://pyimagesearch.com/2015/10/05/opencv-gamma-correction/> (дата звернення: 02.02.2023).

27. Image Segmentation using K Means Clustering. URL: <https://www.geeksforgeeks.org/image-segmentation-using-k-means-clustering/> (дата звернення: 03.02.2023).

28. Color Detection & Object Tracking. URL: <https://www.opencv-srf.com/2010/09/object-detection-using-color-seperation.html> (дата звернення: 11.02.2023).

29. Introduction to OpenCV with Python. URL: <https://medium.com/@elvisdias/introduction-to-opencv-with-python-i-9fc72082f230> (дата звернення: 02.02.2023).

30. A Simple Explanation of K-Means Clustering. URL: <https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/> (дата звернення: 12.02.2023).

31. What is Gaussian blur in image processing?. URL: <https://www.educative.io/answers/what-is-gaussian-blur-in-image-processing> (дата звернення: 02.02.2023).

32. Adaptive thresholding. URL: <https://www.codingninjas.com/codestudio/library/adaptive-thresholding> (дата звернення: 02.02.2023).