

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

ВОДЗІНСЬКА Віта Миколаївна

**Модуль багатопотокового розпаралелення процесу
опрацювання зображень / A module for multi-threaded
parallelization of the image processing.**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
Водзінська Віта Миколаївна

Науковий керівник
к.т.н., доц. О.Й. Піцун

ТЕРНОПІЛЬ-2023

РЕЗЮМЕ

Кваліфікаційна робота на тему «Модуль багатопотокового розпаралелення процесу опрацювання зображень» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 55 сторінок пояснюючої записки, 19 рисунків, 12 таблиць, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розроблення програмного модуля багатопотокового розпаралелення процесу опрацювання зображень.

Методи дослідження в області багатопотокового розпаралелення включають методи і логічну структурування зображень, методи структурного програмування, теорію елементи математичної логіки.

Метою розробки модуля багатопотокового розпаралелення є створення ефективного і швидкого інструменту для обробки зображень. Модуль повинен забезпечувати розпаралелене виконання обробки зображень, що дозволить збільшити продуктивність і скоротити час обробки. Додатковою метою є забезпечення простоти використання модуля та підтримка різних форматів зображень.

Розроблено модуль багатопотокового розпаралелення обробки зображень реалізує такі основні функції: завантаження зображень, обробка зображень; збереження зображень; керування параметрами обробки; Візуалізація результатів.

Ключові слова: БАГАТОПОТОКОВОГО РОЗПАРАЛЕЛЕННЯ ОБРОБКИ ЗОБРАЖЕНЬ, МОДУЛЬ, РОЗПАРАЛЕЛЕННЯ, БАГАТОПОТОКОВОГО.

RESUME

The qualification work on the topic "Module of multi-threaded parallelization of the image processing process" from specialty 123 "Computer engineering" of the bachelor's degree contains 55 pages of explanatory note, 19 figures, 12 tables, 3 appendices. The amount of graphic material is 2 sheets of A3 format.

The purpose of the qualification work is to develop a software module for multi-threaded parallelization of the image processing process.

Research methods in the field of multi-threaded parallelization include methods and logical structuring of images, methods of structural programming, theory and elements of mathematical logic.

The goal of the development of the multi-threaded parallelization module is to create an efficient and fast tool for image processing. The module should provide parallel execution of image processing, which will increase productivity and reduce processing time. An additional goal is to make the module easy to use and support different image formats.

The multi-threaded parallelization module for image processing has been developed and implements the following basic functions: image loading, image processing; saving images; management of processing parameters; visualization of results.

Keywords: MULTI-THREAD PARALLELIZATION OF IMAGE PROCESSING, MODULE, PARALLELIZATION, MULTI-THREAD.

ЗМІСТ

Перелік умовних скорочень.....	9
Вступ.....	10
1. Аналіз багатопотокового розпаралелення процесу опрацювання зображень	13
1.1 Аналіз багатопотокового опрацювання зображень.....	13
1.2 Огляд модулів багатопотокового розпаралелення процесів.....	14
1.3 Опис програмних засобів реалізації багатопотоковості	20
1.4 Постановка задачі кваліфікаційної роботи	22
Для досягнення мети роботи слід виконати такі завдання:	22
2. Проектування модуля багатопотокового розпаралелення процесу опрацювання зображень	23
2.1 Підходи до розпаралелення обробки зображень	23
2.2 Проектування структури модуля багатопотокової обробки зображення ...	24
2.3 Аналіз існуючих алгоритмів багатопотокового оброблення зображень	26
3. Програмна реалізація модуля багатопотокового опрацювання зображень	34
3.1 Програмні засоби реалізації модуля	34
3.2 Реалізація модуля багатопотокового опрацювання зображення	42
3.3 Тестування роботи модуля багатопотокового опрацювання	44
4 Техніко-економічне обґрунтування	49
4.1 Розрахунок витрат на розробку програмного забезпечення.....	49
4.2 Визначення прогнозованої (договірної) ціни КС	53
4.3 Визначення економічної ефективності	57
Висновок.....	61
Список використаних джерел.....	62
Додаток А Довідка про використання	69
Додаток Б Світлокопії виданих публікацій	70

					КР.КІ.8351493.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив		Водзінська В.М			МОДУЛЬ БАГАТОПОТОКОВОГО РОЗПАРАЛЕЛЕННЯ ПРОЦЕСУ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ	Літ.	Арк.	Акрушів
Перевір.		Піцун О.Й.					7	
Консульт.		Савка Н.Я.				ЗУНУ, ФКІТ, КІ-41		
Н. Контр.		Мельник Г.М.						
Затвердив		Дубчак Л.О.						

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПК-персональний комп'ютер

МН-машинне навчання

QA - Quality Assurance

VR – Virtual reality

ООП-об'єктно орієнтоване програмування

ПЗ- програмне забезпечення

ОС – операційна система

IDE - Інтегроване середовище розробки (Integrated Development Environment)

DL - Глибоке навчання (Deep Learning)

NN- Нейронна мережа (Neural Network)

CNN - Згортова нейронна мережа (Convolutional Neural Network)

RNN - Рекурентна нейронна мережа (Recurrent Neural Network)

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

ВСТУП

Модуль багатопотокового розпаралелення процесу опрацювання зображень є актуальним і широко використовується для прискорення обробки зображень за допомогою паралельних обчислень.

Метою модуля багатопотокового розпаралелення процесу опрацювання зображень є покращення ефективності та швидкодії обробки зображень за допомогою використання паралельних обчислювальних потоків. Цей модуль може забезпечити швидкі та ефективні обчислення для опрацювання великого обсягу зображень, що дозволяє прискорити процес обробки та забезпечити більш оперативну відповідь на вимоги сучасних застосувань, таких як комп'ютерне зорове сприйняття, обробка зображень у реальному часі, медична діагностика та багато інших.

З ростом обсягу та складності зображень, що обробляються, та зростання вимог до швидкості обробки, модуль багатопотокового розпаралелення процесу опрацювання зображень стає дедалі більш актуальним. Він дозволяє ефективно використовувати ресурси багатоядерних та розподілених систем, що відкриває нові можливості для обробки зображень у великому масштабі.

Цей модуль також відповідає сучасним тенденціям у розвитку обчислювальних систем, де паралельні обчислення стають все більш поширеними. Багатопотокове розпаралелення дозволяє використовувати потужність сучасних процесорів та графічних прискорювачів для швидкого та ефективного опрацювання зображень.

Отже, модуль багатопотокового розпаралелення процесу опрацювання зображень має велику актуальність у сучасному світі, де швидкість, ефективність та точність обробки зображень відіграють ключову роль у різних сферах, включаючи медицину, медіа, комп'ютерне зорове сприйняття та багато інших додатків. Основна проблема полягає в тому, що обробка зображень вимагає велику кількість обчислень, таких як фільтрація, згортка, збільшення

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

чіткості, зменшення шуму тощо. Виконання цих обчислень послідовно в одному потоці займає багато часу. Однак, застосування багатопотоковості дозволяє розподілити обчислювальні завдання між кількома потоками і виконувати їх паралельно. Це призводить до значного збільшення швидкодії обробки зображень. Таким чином, використання багатопотокового розпаралелення процесу обробки зображень є важливим для розв'язання проблеми повільної обробки зображень та досягнення швидкодії обробки в реальному часі. Це особливо корисно у сферах, де важливо оперативно отримувати результати обробки зображень, таких як медична діагностика, розпізнавання обличчя, аналіз відео тощо.

Для досягнення мети роботи слід виконати такі завдання:

- розібратися з вимогами до модуля багатопотоковості опрацювання зображень;
- дослідити різні технології та бібліотеки, які надають можливості багатопотоковості;
- спланувати структуру модуля багатопотоковості опрацювання зображень;
- провести ретельне тестування модуля багатопотоковості опрацювання зображень;
- розробити користувацький інтерфейс;
- розробити багаторівневий проект модуля багатопотоковості опрацювання зображень.

Об'єкт дослідження. Розробка модуля багатопотоковості опрацювання зображень

Предмет дослідження. Модуль багатопотоковості опрацювання зображень

У першому розділі було проаналізовано вимоги для роботи модуля багатопотоковості опрацювання зображень. Здійснено аналіз існуючих модулів які їх переваги недоліки що варто взяти за основу власної розробки. Також

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

проаналізовано існуючі бібліотеки для реалізації програмного модуля багатопотокового опрацювання зображень.

У другому розділі роботи розробили основні вимоги до роботи модуля багатопотоковості опрацювання зображень. З метою кращого відображення роботи модуля, використали діаграми, які ілюструють його функціональність.

На основі вимог, провели аналіз різних алгоритмів та їх використання в модулі. Для цього, дослідили різні алгоритми та їх можливості визначили ті, що найкраще відповідають потребам даного модуля. Загалом, аналіз дав нам чітку картину про те, яким повинен бути наш модуль багатопотоковості опрацювання зображень і які кроки необхідно вжити для його розробки.

У третьому розділі роботи описали середовище розробки для модуля багатопотоковості опрацювання зображень. Розглянули різні середовища розробки, і проаналізували їх переваги і недоліки.

Далі, було проаналізовано різні мови програмування і їх функціонал, щоб визначити, яку мову використати для розробки модуля. З урахуванням потреб і вимог, було визначено, найбільш підходящу для розробки модуля.

Також розробили повноцінний модуль багатопотоковості опрацювання зображень. Було розроблено відповідні алгоритми та інтегрували їх у розробці модуля.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

1. АНАЛІЗ БАГАТОПОТОКОВОГО РОЗПАРАЛЕЛЕННЯ ПРОЦЕСУ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ

1.1 Аналіз багатопотокового опрацювання зображень

Стан та аналіз даної теми, а саме багатопотокового розпаралелення процесу опрацювання зображень, є важливим для визначення рівня розвитку і досліджень в цій галузі. Можна перерахувати деякі основні моменти стану даної теми.

Багатопотокове розпаралелення процесу опрацювання зображень має широкі застосування в різних галузях, включаючи комп'ютерне зорове розпізнавання, обробку відео, медичну діагностику, аналіз зображень в сфері безпеки та багато інших. Сучасні процесори мають велику кількість ядер та підтримують паралельні обчислення, що дозволяє використовувати багатопотоковість для прискорення обробки зображень. У розробці програм для багатопотокового розпаралелення процесу опрацювання зображень широко використовуються різні бібліотеки та фреймворки, такі як OpenCV, TensorFlow, PyTorch, NumPy та інші. Ці інструменти надають зручні інтерфейси для паралельної обробки зображень.

Одним із викликів у багатопотоковому розпаралеленні процесу опрацювання зображень є ефективне розподілення завдань між потоками та управління синхронізацією даних. Крім того, обробка зображень може вимагати великої кількості пам'яті та обчислювальних ресурсів. В галузі багатопотокового розпаралелення процесу опрацювання зображень продовжуються дослідження щодо оптимізації алгоритмів, підвищення продуктивності, покращення якості обробки та зниження обчислювальних витрат.

У загальному, стан даної теми свідчить про постійний розвиток та високий інтерес до багатопотокового розпаралелення процесу опрацювання зображень. Використання цієї технології дозволяє досягати швидкодії обробки

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

зображень, що має велике значення для багатьох галузей, де швидка та ефективна обробка зображень є важливою вимогою.

Аналіз стану та функціональних можливостей багатопотокового розпаралелення процесу опрацювання зображень враховує як наявні досягнення, так і потенційні можливості для подальшого розвитку. Це допомагає визначити напрямки подальшого дослідження та вдосконалення цієї технології.

1.2 Огляд модулів багатопотокового розпаралелення процесів

Огляд і аналіз існуючих рішень є важливим пунктом в дослідженні даного модуля. Для даного пункту буде доцільно зробити огляд існуючих програм, наукових статей та багато іншого. Почнемо з програм які підтримують цей модуль. До цього списку належать такі програми:

- Adobe Photoshop;
- GIMP;
- Lightroom;
- ImageJ

Почнемо з короткого опису переваг і недоліків цих програм. І першою програмою яку будемо досліджувати є Adobe Photoshop. Цей професійний графічний редактор має функціонал для обробки зображень, який може використовувати багатопотоковість для прискорення обчислень. На рисунку 1.1 зображено інтерфейс програми.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

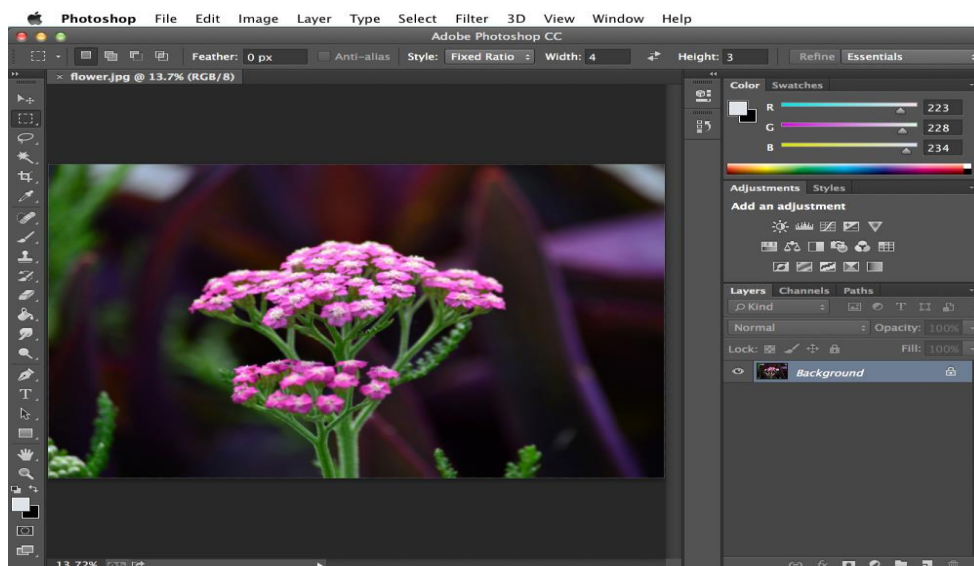


Рисунок 1.1–інтерфейс Adobe Photoshop

Ось деякі основні переваги і недоліки:

Переваги:

- Широкий функціонал для редагування та обробки зображень.
- Можливість використання багатопотоковості для швидкої обробки зображень.
- Професійні інструменти для колориметрії та корекції зображень.

Недоліки:

- Висока вартість ліцензії для повної версії програми.
- Вимоги до апаратного забезпечення, що можуть бути важкими для менш потужних систем.

Наступним кроком є дослідження GIMP, інтерфейс програми зображено на рисунку 1.2

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

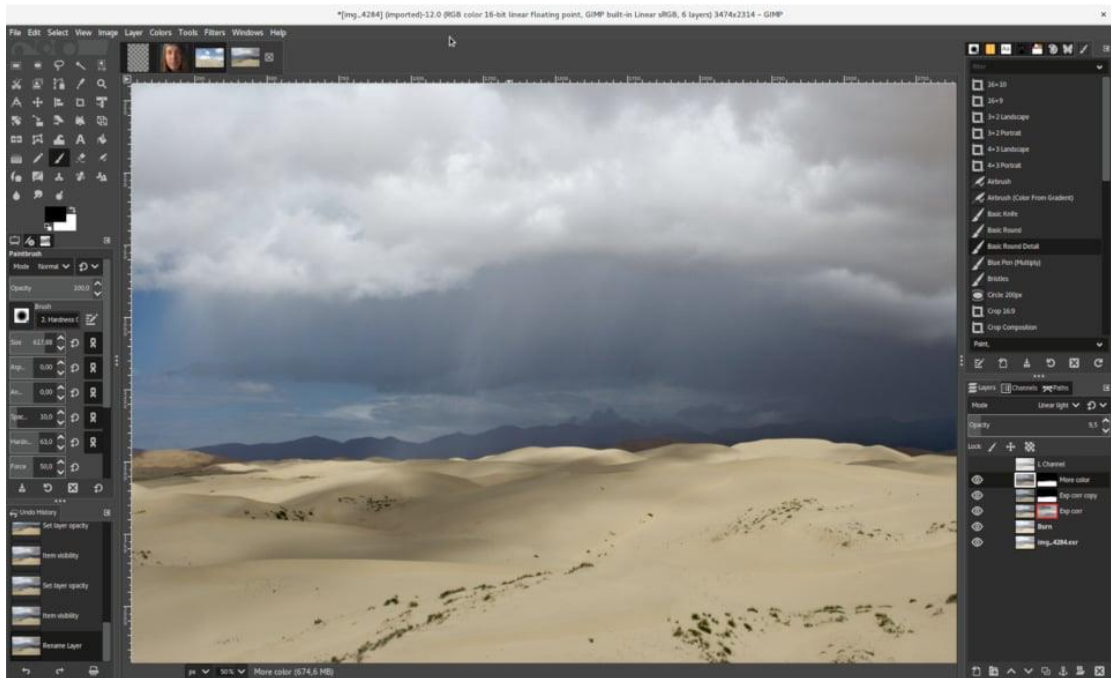


Рисунок 1.2—інтерфейс GIMP

GIMP (GNU Image Manipulation Program) - це безкоштовне програмне забезпечення з відкритим кодом, яке надає широкі можливості для редагування та обробки зображень, також має можливість багатопотокового розпаралелення обробки зображень. Ось деякі переваги та недоліки GIMP:

Переваги GIMP:

- GIMP є вільним програмним забезпеченням з активною спільнотою розробників, що означає, що ви можете переглядати та змінювати його вихідний код відповідно до своїх потреб.
- GIMP доступний для безкоштовного використання без обмежень часу або функціональності. Ви можете використовувати його на будь-якій платформі, включаючи Windows, macOS та Linux.
- GIMP має велику кількість плагінів та скриптів, які дозволяють розширити його функціональність та надати нові можливості для обробки зображень.
- GIMP надає широкий спектр інструментів для редагування та обробки зображень, включаючи корекцію кольору, ретушування, створення малюнків, маск, шарів та багато іншого.

Недоліки GIMP:

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

– Для новачків може знадобитися трохи часу, щоб зрозуміти та звикнути до інтерфейсу GIMP, оскільки він може відрізнятися від інших графічних редакторів.

– Деякі функції, які можуть бути присутні в комерційних програмах, можуть бути відсутніми або менш розвинутими в GIMP.

– Деякі функції GIMP можуть бути вимогливими до системи та вимагати достатньо потужного обладнання для оптимальної роботи.

В цілому, GIMP є потужним та безкоштовним інструментом для редагування та обробки зображень. Він може задовольнити потреби як початківців, так і досвідчених користувачів, забезпечуючи широкий функціонал та можливості розширення.

Дослідимо ще одну популярну програму для роботи із зображенням яка називається Lightroom (рисунок 1.3)

Це програмне забезпечення для обробки та каталогізації фотографій, яке також використовує багатопотоковість для прискорення обчислень.

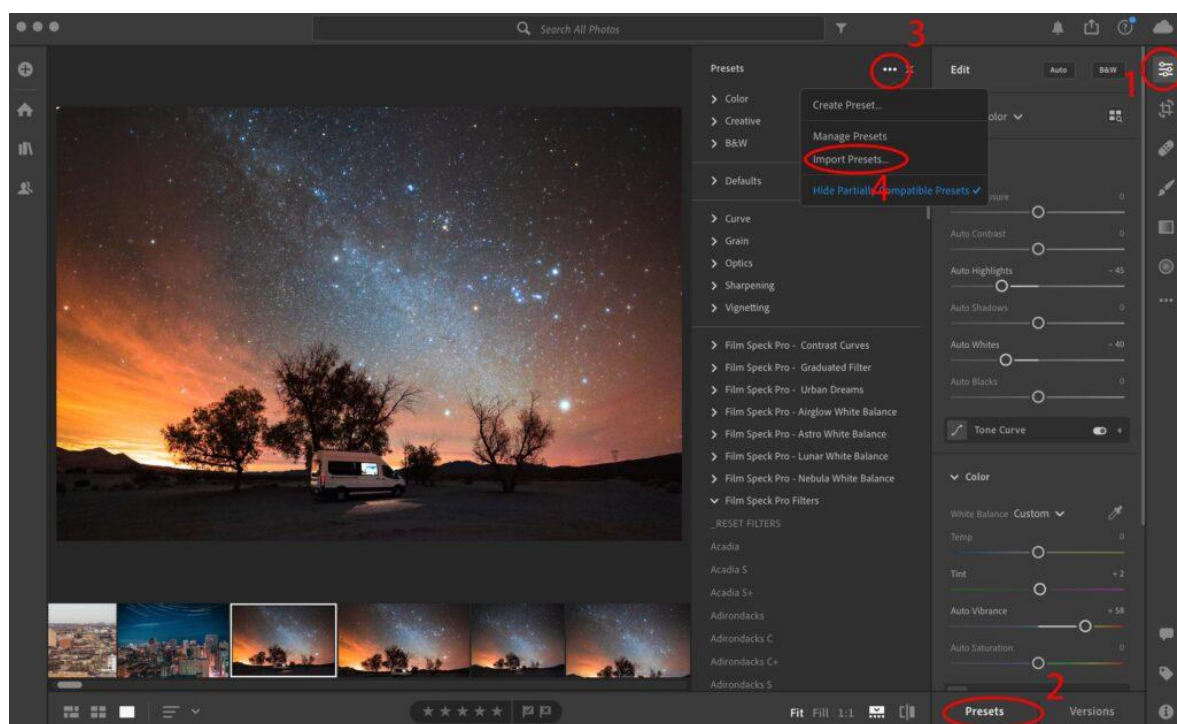


Рисунок 1.3– інтерфейс Lightroom

Lightroom - це програмне забезпечення, розроблене компанією Adobe, спеціально для обробки та каталогізації фотографій

Переваги Lightroom:

– Lightroom має широкий спектр функцій та інструментів, що дозволяють редагувати, коригувати колір, вирівнювати тони, виконувати обрізку та ретушування фотографій.

– Lightroom підтримує багатопотокову обробку, що дозволяє прискорити час обробки зображень шляхом одночасного виконання декількох операцій.

– Інтерфейс Lightroom є зручним та простим у використанні, що дозволяє швидко навчатися та працювати з програмою.

– Lightroom інтегрований з іншими програмами Adobe, такими як Photoshop, що дозволяє легко переміщуватися між програмами та обмінюватися зображеннями.

Недоліки Lightroom:

– Lightroom вимагає підписки або придбання ліцензії, що може бути витратним для деяких користувачів.

– Lightroom може вимагати потужного комп'ютера для оптимальної роботи з великими обсягами фотографій.

– Lightroom спеціалізується на обробці фотографій, тому деякі функції, які можуть бути присутніми в інших програмах, можуть бути обмеженими або відсутніми.

У кінцевому рахунку, Lightroom є потужним інструментом для обробки та каталогізації фотографій, але вибір програмного забезпечення залежатиме від ваших потреб та бюджету.

І остання програма яку будемо досліджувати є ImageJ (рисунок 1.4). Це безкоштовне програмне забезпечення для аналізу зображень та обробки зображень, яке має можливість використовувати багатопотоковість для оптимального використання ресурсів.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

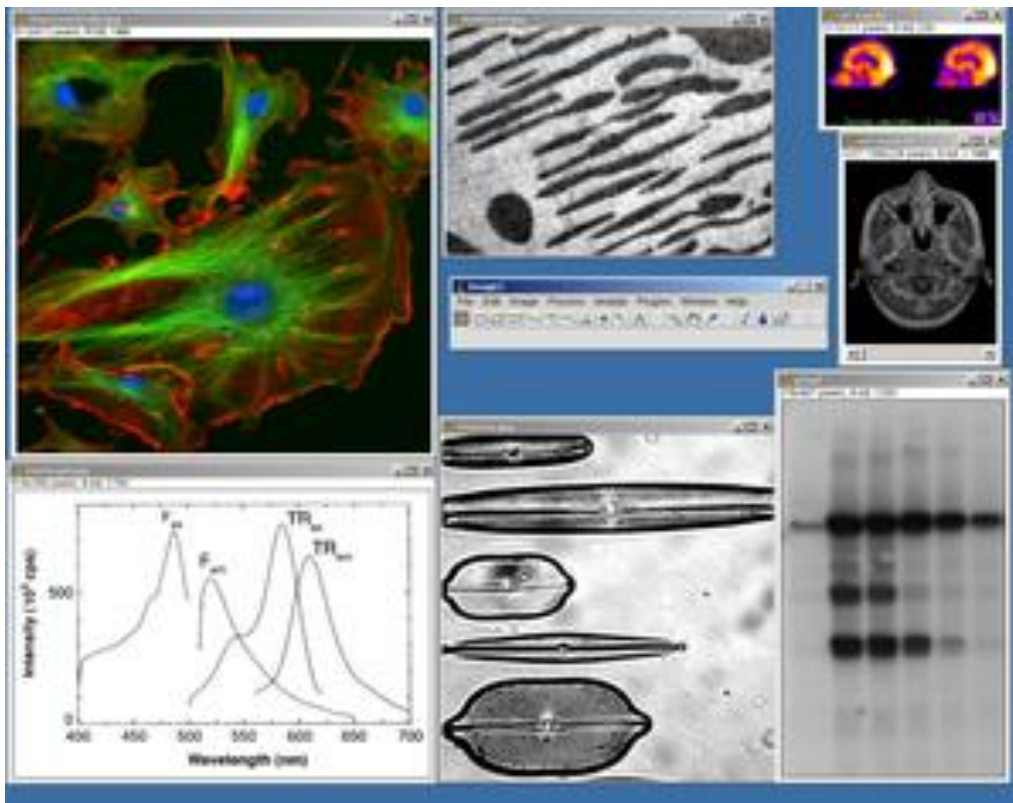


Рисунок 1.4-інтерфейс ImageJ

Переваги:

- безкоштовне програмне забезпечення з відкритим кодом;
- можливість редагувати та обробляти зображення з використанням багатопотоковості;
- розширюваність за допомогою плагінів та скриптів.

Недоліки:

- Може мати менший функціонал порівняно з комерційними програмами.
- Інтерфейс може бути менш зручним для користувачів, які не знайомі з інструментами обробки зображень.

У кожної програми є свої переваги та недоліки, і вибір програмного забезпечення залежатиме від конкретних потреб та пріоритетів користувача. Наприклад, Adobe Photoshop є потужним та професійним інструментом, але вимагає великих витрат на ліцензію. GIMP та ImageJ, незважаючи на своє відкрите джерело, можуть мати обмежений функціонал порівняно з

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

комерційними аналогами. Lightroom спеціалізується на обробці фотографій та має зручний інтерфейс для роботи з ними.

У кінцевому рахунку, вибір програмного забезпечення для багатопотокового розпаралелення процесу опрацювання зображень буде залежати від особистих потреб, навичок та бюджету.

1.3 Опис програмних засобів реалізації багатопотоковості

Засоби реалізації багатопотокового розпаралелення процесу опрацювання зображень можуть включати як мови програмування так і фреймворки і бібліотеки. Спершу опишемо мови програмування. Ось опис деяких мов програмування, які часто використовуються для багатопотокового розпаралелення процесу опрацювання зображень:

Python - це інтерпретована, високорівнева мова програмування, яка надає широкі можливості для обробки зображень. Вона має багатопотокову підтримку за допомогою модулів, таких як `threading` і `concurrent.futures`. Python також має багато бібліотек для обробки зображень, таких як OpenCV, Pillow та scikit-image, які підтримують багатопотокову обробку.

Java - це об'єктно-орієнтована мова програмування, яка має вбудовану підтримку для багатопотоковості. Вона має класи та методи для створення та управління потоками виконання. Java також має багато бібліотек, таких як OpenCV та ImageJ, які надають функціональність для обробки зображень та підтримують багатопотокову обробку.

C++ - це мова програмування загального призначення, яка надає низькорівневий доступ до ресурсів комп'ютера. Вона має потужну підтримку багатопотоковості за допомогою бібліотеки `std::thread` та інших механізмів. C++ також має багато бібліотек для обробки зображень, таких як OpenCV та CImg, які надають можливості багатопотокової обробки.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

CUDA (Compute Unified Device Architecture) - це платформа та модель програмування для обчислень на графічних процесорах (GPU). Вона надає можливість використовувати сотні або тисячі ядер GPU для паралельного виконання обчислень. CUDA має спеціальну бібліотеку для обробки зображень, відому як CUDA Toolkit, яка дозволяє розпаралелити процеси обробки зображень на GPU.

Кожна з цих мов має свої переваги та обмеження. Вибір мови програмування залежить від потреб, рівня досвіду та вподобань.

Також в цьому пункті можна описати деяких фреймворків, які можна використовувати для багатопотокового розпаралелення процесу опрацювання зображень:

OpenCV є одним з найпопулярніших фреймворків для обробки зображень. Він надає широкий набір функцій для завантаження, обробки та аналізу зображень. OpenCV також має підтримку багатопотоковості, що дозволяє розпаралелити обробку зображень на різні потоки та використовувати багатоядерні процесори для прискорення обчислень.

TensorFlow є одним з найпопулярніших фреймворків для машинного навчання та обробки зображень. Він надає багато функцій для обробки зображень, таких як зчитування, обрізка, зміна розміру, нормалізація та інші. TensorFlow також має підтримку багатопотоковості та може використовувати багатоядерні процесори або графічні процесори для прискорення обчислень.

PyTorch є іншим популярним фреймворком для машинного навчання та обробки зображень. Він надає багато функцій для роботи з зображеннями, таких як обрізка, зміна розміру, поворот, фільтрація та багато інших. PyTorch також має підтримку багатопотоковості та може використовувати багатоядерні процесори або графічні процесори для прискорення обчислень.

Keras є високорівневим фреймворком для машинного навчання, який побудований на основі TensorFlow. Він надає простий та інтуїтивний інтерфейс для обробки зображень та створення моделей нейронних мереж. Keras також має підтримку багатопотоковості, яка дозволяє розпаралелити обробку

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

зображень на різні потоки та використовувати багатоядерні процесори для прискорення обчислень.

Кожен з цих фреймворків має свої переваги та особливості. Вибір фреймворку залежить також від потреб, рівня досвіду та вподобань.

В першому розділі було проведено дослідження актуальності модуля багатопотокового опрацювання зображень, а саме було проаналізовано в яких сферах може застосовуватись даний модуль і які переваги він надасть в майбутньому.

Також було проведено порівняння існуючих програмних засобів які опрацьовують зображення за допомогою багатопотоковості, було описано їх плюси і мінуси і також було описано функції які можна б було додати до роботи цих програмних модулів щоб покращити їх роботу.

Було описано існуючі програмні засоби для багатопотоковості, а саме мови програмування які її підтримують і бібліотеки. Коротко описано їх переваги і недоліки і також було вказано рівень навчальних знань для роботи з ними.

1.4 Постановка задачі кваліфікаційної роботи

Для досягнення мети роботи слід виконати такі завдання:

- Розібратися з вимогами до модуля багатопотоковості опрацювання зображень,.
- Дослідити різні технології та бібліотеки, які надають можливості багатопотоковості.
- Спланувати структуру модуля багатопотоковості опрацювання зображень,
- Провести ретельне тестування модуля багатопотоковості опрацювання зображень,
- Розробити користувацький інтерфейс

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

2. ПРОЕКТУВАННЯ МОДУЛЯ БАГАТОПОТОКОВОГО РОЗПАРАЛЕЛЕННЯ ПРОЦЕСУ ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ

2.1 Підходи до розпаралелення обробки зображень

В сучасному світі існує кілька підходів до розпаралелення обробки зображень.

Розпаралелення на рівні пікселів- цей підхід передбачає розділення обробки зображення на окремі операції над кожним пікселем. Наприклад, виконання фільтрації, розмиття або зміни кольору на кожному пікселі окремо. Для розпаралелення можуть використовуватись різні технології, такі як OpenMP або CUDA, які дозволяють розподілити роботу між різними обчислювальними пристроями, такими як багатопотокові процесори або графічні процесори. Цей підхід підходить для задач, де обробка кожного пікселя зображення незалежна від інших пікселів.

Розпаралелення на рівні об'єктів- в цьому підході зображення розподіляється на окремі об'єкти або регіони, і кожен об'єкт обробляється окремо. Наприклад, виявлення об'єктів на зображенні або розпізнавання образів. Цей підхід може бути використаний для розпаралелення обробки об'єктів між різними обчислювальними пристроями або потоками. Для цього можуть використовуватись бібліотеки комп'ютерного зору, такі як OpenCV або TensorFlow, які надають засоби для обробки об'єктів на зображенні.

Розпаралелення за допомогою багатопотоковості- цей підхід використовує багатопотоковість для розподілу обробки зображень між різними потоками в одному процесорі або між різними процесорами. Кожен потік може обробляти окремі частини зображення або виконувати окремі операції. Це може бути досягнуто за допомогою бібліотек, таких

як OpenCV або Pillow, які надають можливість створювати багатопотокові програми для обробки зображень.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Розпаралелення на рівні задач - в цьому підході обробка зображень розподіляється між різними вузлами обчислювальної мережі або хмарними ресурсами. Кожен вузол або ресурс може обробляти окремі частини зображення або виконувати окремі задачі. Цей підхід корисний для великого обсягу обробки зображень або при виконанні складних алгоритмів, де розподіл роботи між різними обчислювальними ресурсами забезпечує більшу швидкодію і ефективність обробки.

Кожен з цих підходів має свої переваги і недоліки, і вибір підходу залежить від вимог проекту, характеристик оброблюваних зображень, доступних ресурсів та інших факторів. Найкращий підхід може бути вибраний шляхом аналізу і порівняння різних підходів у контексті конкретного проекту.

2.2 Проектування структури модуля багатопотокової обробки зображення

Проектування структури модуля багатопотокового розпаралелення процесу опрацювання зображень включає в себе наступні модулі:

Ядро модуля—це центральна частина модуля, яка відповідає за управління потоками та обробку зображень. Вона включає в себе основні функції, які розподіляють завдання між потоками, виконують обробку зображень та забезпечують координацію між ними. Ядро може бути реалізоване за допомогою класу або модуля, який містить необхідні методи та атрибути.

Потоків—це компонент модуля, який відповідає за створення та керування потоками обробки зображень. Він може включати в себе функції для створення нових потоків, виконання завдань у паралельних потоках, синхронізації та координації роботи потоків.

Обробки зображень – це компонент модуля, який містить функції та алгоритми для реалізації різних операцій обробки зображень. Він може

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

включати в себе методи для зчитування зображень з файлів, виконання операцій обробки, збереження оброблених зображень та інші функції, які допомагають користувачам маніпулювати зображеннями.

Інтерфейс користувача—це компонент модуля, який забезпечує взаємодію з користувачем. Він може включати в себе графічний інтерфейс, командний рядок або АРІ для програмної взаємодії з модулем. Інтерфейс користувача дозволяє користувачам вибирати операції обробки, налаштовувати параметри та контролювати процес обробки зображень.

Тестування та налагодження—це важливий компонент проектування модуля, який включає в себе написання тестових сценаріїв для перевірки функціональності та надійності модуля. Тестування допомагає виявити та виправити помилки та проблеми в роботі модуля перед його використанням в реальних проектах.

Документація та довідка включає в себе написання документації, яка описує функціональність, використання та налаштування модуля. Документація повинна бути зрозумілою та повністю охоплювати всі основні аспекти модуля, щоб користувачі могли ефективно використовувати його в своїх проектах.

Ці компоненти співпрацюють разом для створення модуля багатопотокового розпаралелення процесу опрацювання зображень, що забезпечує його ефективну та зручну роботу для користувачів.

Проектування структури модуля багатопотокового розпаралелення процесу опрацювання зображень також може включати такі елементи:

Масштабування—це компонент, який дозволяє модулю адаптуватися до збільшення обсягу роботи та кількості зображень, які потрібно обробляти. Модуль масштабування може включати в себе механізми автоматичного розподілу завдань між різними машинами або серверами, що дозволяє підвищити продуктивність та швидкість обробки зображень.

Оптимізації дозволяє оптимізувати швидкість та ефективність роботи модуля. Він може включати в себе алгоритми оптимізації, які допомагають зменшити час обробки зображень, ефективно використовуючи ресурси системи та оптимізуючи розподіл завдань між потоками.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

Безпеки забезпечує захист від несанкціонованого доступу та зламу системи. Модуль безпеки може включати в себе механізми автентифікації, авторизації та шифрування даних, що допомагають зберегти конфіденційність та цілісність зображень під час їх обробки.

Моніторингу та логування відстежує та реєструє дії та події, що відбуваються в модулі. Модуль моніторингу та логування може включати в себе функції для запису журналів, моніторингу продуктивності, аналізу помилок та інших подій, що допомагають управляти та вдосконалювати роботу модуля. Виконання цих компонентів можна відобразити в діаграмі послідовності на рисунку 2.1

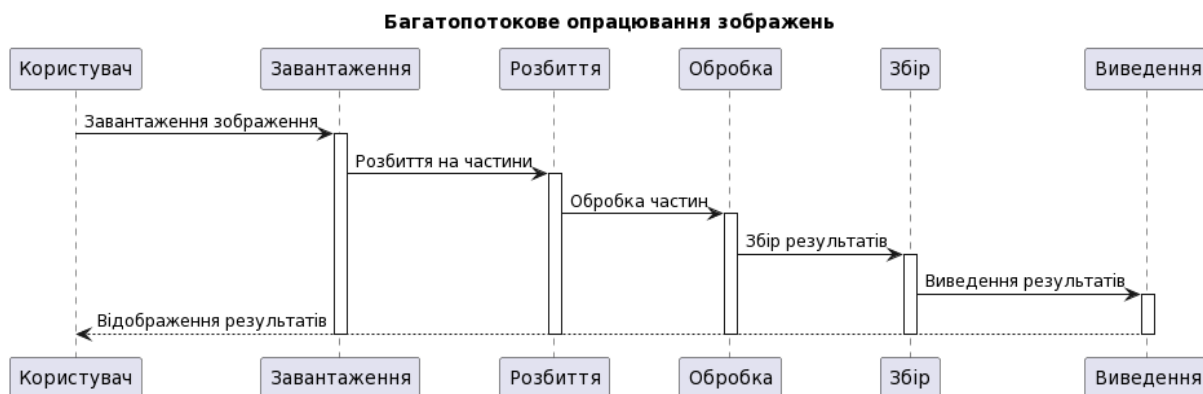


Рисунок 2.1—діаграма послідовності

Ці додаткові компоненти доповнюють основну структуру модуля багатопотокового розпаралелення процесу опрацювання зображень та розширюють його можливості для виконання різних завдань обробки зображень.

2.3 Аналіз існуючих алгоритмів багатопотокового оброблення зображень

Існує кілька алгоритмів та підходів, які можна використовувати для реалізації модуля багатопотокового розпаралелення процесу опрацювання

зображень. Ось деякі з них:

- Розділення завдань:
- Пул потоків:
- Модель "виробник-споживач":
- Модель акторів:

Тепер опишемо кожен з них по порядку:

Підхід розділення завдань полягає у розбитті процесу опрацювання зображень на окремі завдання, які можуть бути виконані паралельно. Наприклад, зчитування зображень може бути виконане окремо в кожному потоці, а потім обробка кожного зображення також виконується паралельно. Це дозволяє зменшити загальний час обробки.

Алгоритм «Пул потоків» використовує пул потоків, в якому певна кількість потоків створюється наперед і готова до виконання завдань. Коли з'являється нове завдання, воно надсилається до вільного потоку в пулі для обробки. Після виконання завдання потік повертається в пул і стає доступним для інших завдань. Це дозволяє забезпечити ефективне використання ресурсів та контроль над кількістю одночасно виконуваних завдань.

Підхід "виробник-споживач" використовує два види потоків - виробників та споживачів. Виробники генерують нові завдання обробки зображень та розміщують їх у черзі. Споживачі беруть завдання з черги та виконують їх обробку. Цей підхід дозволяє гнучко керувати кількістю виробників та споживачів залежно від навантаження та швидкості обробки.

Підхід «Модель акторів» базується на принципі взаємодії незалежних акторів, які обмінюються повідомленнями та виконують свої завдання. Кожен актор може бути відповідальним за обробку конкретного зображення або групи зображень. Вони можуть взаємодіяти між собою, передавати повідомлення та координувати обробку зображень. Цей підхід дозволяє створювати гнучкі та розподілені системи обробки зображень.

Ці алгоритми та підходи можуть бути використані окремо або комбіновані для реалізації модуля багатопотокового розпаралелення процесу

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

опрацювання зображень, залежно від вимог та особливостей конкретного проекту.

Використання мови програмування Python для багатопотокового оброблення зображень може бути зручним завдяки наявності багатьох бібліотек, які надають підтримку паралельного обчислення та обробки зображень. Ось кілька алгоритмів, які можуть бути використані для багатопотокового оброблення зображень у Python:

Бібліотека `concurrent.futures` входить до стандартної бібліотеки Python і надає простий спосіб виконання багатопотокових та багатопроецесних обчислень. Вона базується на концепції пулу потоків або процесів, де можна подати задачі на виконання та отримати результати.

Основні переваги використання бібліотеки `concurrent.futures` для багатопотокового обчислення:

Бібліотека надає простий та зрозумілий інтерфейс для використання багатопотоковості та багатопроецесності. Вона має однаковий API для роботи з потоками та процесами, що спрощує перехід від одного до іншого.

Бібліотека дозволяє легко розподіляти завдання між різними потоками або процесами, що забезпечує масштабованість та покращену продуктивність. Можна використовувати пул потоків або процесів для ефективного використання ресурсів.

Бібліотека дозволяє керувати виконанням завдань та отримувати результати. Можна використовувати функцію `submit()` для подання завдання на виконання та отримувати об'єкт `Future`, з якого можна отримати результат після завершення обчислення.

Недоліки використання бібліотеки `concurrent.futures` для багатопотокового обчислення:

Бібліотека має досить обмежені можливості для керування потоками або процесами. Вона не надає таких розширених можливостей, як синхронізація потоків, спільний доступ до ресурсів або взаємодія між потоками.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

Бібліотека не надає вбудованої підтримки для розподіленого обчислення, що може бути обмеженням у випадках, коли потрібно виконати обчислення на різних машинах.

Бібліотека спрощена і має обмежений набір функціональних можливостей. Якщо потрібні більш розширені функції, можливо, доведеться використовувати інші бібліотеки або розробляти власний код.

Загалом, бібліотека `concurrent.futures` є зручним інструментом для простих багатопотокових обчислень, але вона має свої обмеження і може не влаштовувати для складних випадків. Перед використанням слід ретельно розглянути вимоги проекту та вибрати найбільш підходящий інструмент. `OpenCV` є потужною бібліотекою для обробки зображень у Python. Вона надає функції для маніпулювання та обробки зображень. Ви можете використати ці функції для створення багатьох потоків, які паралельно обробляють різні частини зображень. Наприклад, ви можете розбити зображення на різні регіони або блоки та обробляти їх у окремих потоках.

`NumPy` є популярною бібліотекою для наукових обчислень у Python. Вона має вбудовану підтримку багатопотоковості через векторизовані операції. Ви можете використовувати ці векторизовані операції для ефективного обчислення на великих масивах зображень у багатьох потоках одночасно.

`Dask` є бібліотекою для паралельних обчислень у Python. Вона дозволяє розподіляти обробку зображень на різні обчислювальні вузли або ресурси та керувати потоками обчислень. `Dask` дозволяє створювати обчислювальні графи, які описують послідовність операцій над зображеннями, і розподіляти ці операції між потоками чи процесами для паралельного виконання.

`Dask` є бібліотекою для обчислень у Python, яка дозволяє розподіляти та виконувати обчислення на багатьох потоках або в розподіленому середовищі. Вона надає можливості для ефективної обробки даних та виконання складних обчислень у багатопотоковому режимі.

Основні переваги використання `Dask` для багатопотокового обчислення:

`Dask` дозволяє обробляти великі обсяги даних та виконувати складні обчислення у багатопотоковому середовищі. Вона може автоматично

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

розподіляти завдання між різними потоками або вузлами, що дозволяє використовувати всю потужність доступних ресурсів.

Dask надає можливість паралельного виконання обчислень, що дозволяє виконувати кілька операцій одночасно на різних даних. Це забезпечує покращену продуктивність та швидкість обчислень.

Dask може бути легко інтегрована з існуючими кодами на Python. Вона підтримує багато інших популярних бібліотек для обробки даних, таких як NumPy та Pandas, що дозволяє використовувати їх функціонал у багатопотоковому режимі.

Недоліки використання Dask для багатопотокового обчислення:

Для ефективного використання Dask може бути необхідно мати доступ до достатньої кількості процесорних ядер та оперативної пам'яті. В іншому випадку, продуктивність може бути обмежена.

Dask має складнішу синтаксис та вимагає деякої підготовки для його використання. Розробникам може знадобитися деякий час, щоб оволодіти всіма можливостями та особливостями цієї бібліотеки.

Загалом, Dask є потужним інструментом для багатопотокового обчислення у Python, який надає можливості для ефективного обробки даних та виконання складних обчислень у багатопотоковому середовищі. Однак, перед його використанням варто врахувати вимоги до ресурсів та складність використання.

Це лише кілька прикладів алгоритмів та бібліотек, які можуть бути використані для багатопотокового оброблення зображень у Python.

Також існує ще багато бібліотек для розробки модуля багатопотоковго розпаралелення процесу опрацювання зображень. Одна із таких бібліотек є multiprocessing.

Модуль `multiprocessing` в мові програмування Python надає можливості для багатопотокового розпаралелення процесів. Він дозволяє створювати та керувати процесами у мові Python, що дозволяє використовувати багатопотоковість для ефективного обробки задач.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Ярусно-паралельну форму представлення розпаралелення процесу обчислення кількісних характеристик наведено на рисунку 2.2

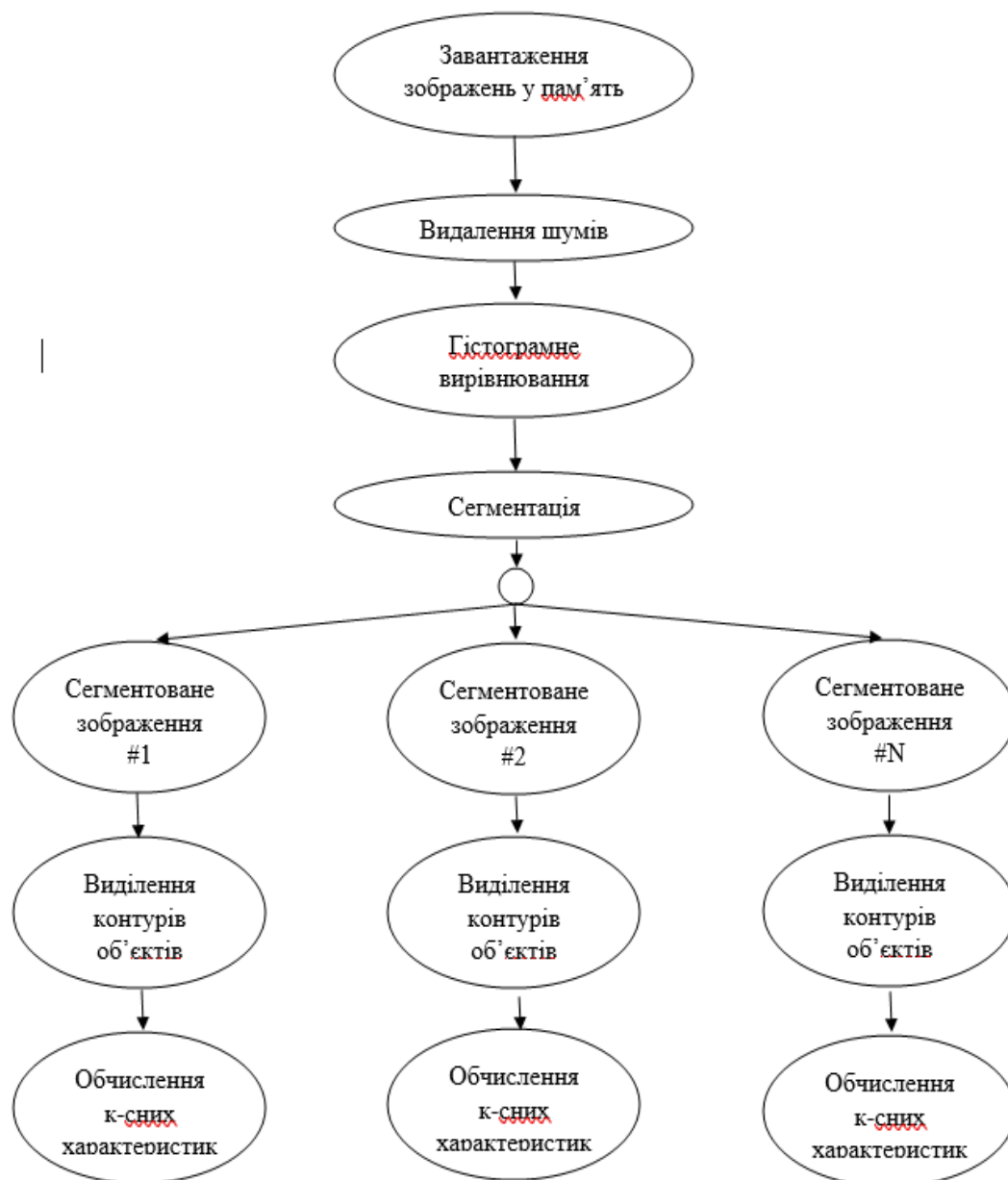


Рисунок 2.2 – Ярусно-паралельна форма представлення розпаралелення процесу обчислення кількісних характеристик.

Деякі з основних функцій та можливостей модуля `multiprocessing` включають:

Дозволяє створювати нові процеси за допомогою класу `Process`. Кожен процес має свій власний набір ресурсів та виконується незалежно від інших процесів. Модуль `multiprocessing` надає функціональність для керування

процесами, таку як запуск, призупинення, відновлення та припинення процесів. Це дозволяє гнучко керувати виконанням багатопотокових задач.

Процеси можуть взаємодіяти між собою за допомогою спільної пам'яті або передачі повідомлень. Модуль `multiprocessing` надає класи та інструменти для зручного обміну даними та комунікації між процесами.

Модуль `multiprocessing` надає можливість створювати пул процесів, що дозволяє автоматично розподіляти задачі між доступними процесами. Це особливо корисно для обробки багатопотокових завдань, коли кількість процесів може бути динамічною. Модуль `multiprocessing` надає механізми для синхронізації та блокування доступу до спільних ресурсів між процесами. Це допомагає уникнути проблем конкурентного доступу до даних.

Використання модуля `multiprocessing` дозволяє ефективно розпаралелити процеси обробки зображень, що дозволяє прискорити виконання завдань та покращити продуктивність програми.

2.4 Постановка завдання модуля багатопотокового опрацювання зображень

Постановка завдання для модуля багатопотокового опрацювання зображень має важливе значення адже за тим які завдання будуть стояти перед розробником буде легше спроектувати програмний модуль.

Мета даного модуля це розробити модуль, який забезпечує багатопотокове опрацювання зображень, забезпечити паралельне виконання операцій над зображеннями для покращення продуктивності та ефективності.

Функціональні вимоги:

- Забезпечити можливість завантаження зображень з різних джерел (локальний файл, URL, камера тощо).
- Реалізувати операції оброблення зображень, такі як зміна розміру, зміна кольору, фільтри, обрізка тощо.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

- Дозволити виконання операцій над зображеннями у багатьох потоках паралельно.
- Забезпечити синхронізацію доступу до спільних ресурсів (наприклад, пам'ять) для уникнення гонок даних та блокувань.
- Забезпечити збереження оброблених зображень у відповідних форматах (наприклад, JPEG, PNG).
- Нефункціональні вимоги:
 - Забезпечити ефективність та швидкість опрацювання зображень у багатопотоковому середовищі.
 - Забезпечити стійкість та надійність модуля, враховуючи можливі помилки оброблення зображень або некоректні дані.
 - Забезпечити гнучкість та розширюваність модуля для додавання нових операцій та функцій обробки зображень.
- Ця постановка завдання визначає основні цілі та вимоги до модуля багатопотокового опрацювання зображень. Вона може бути доповнена або змінена залежно від конкретних потреб проекту.

У даному розділі описано приклади сучасних алгоритмів багатопотоковості, виконано ярусно-паралельну форму представлення розпаралелення процесу обчислення кількісних характеристик.

. А також описані основні завдання роботи алгоритму багатопотокоості для майбутньої розробки. Було представлено за допомогою діаграми послідовності роботу багатопотоковості в цій розробці. Також було представлено опис роботи модуля multiprocessing.

Також було детально описано завдання для виконання майбутньої розробки, а саме описано основні функціональні і нефункціональні вимоги.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

3.ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ БАГАТОПОТОКОГО

ОПРАЦЮВАННЯ ЗОБРАЖЕНЬ

3.1 Програмні засоби реалізації модуля

Вибір засобів для реалізації модуля багатопотокового розпаралелення процесу опрацювання зображень залежить від різних факторів, таких як:

- вимоги проекту;
- мови програмування;
- бібліотеки і фреймворки;
- ресурси;
- Зручність використання.

Ці фактори повинні бути враховані при виборі засобів для реалізації модуля багатопотокового розпаралелення процесу опрацювання зображень.

Також при виборі засобів реалізації треба враховувати вимоги до мов програмування. Вимоги до мови програмування включають ряд аспектів, які впливають на якість і ефективність написаного коду. Ось деякі з них:

Читабельність—код повинен бути легко зрозумілим і читабельним для інших розробників. Використовуйте зрозумілі імена змінних, функцій та класів, розміщуйте коментарі для пояснення складних частин коду та дотримуйтесь стандартів форматування.

Масштабованість – код має бути легко розширюваним та змінюваним для відповіді на змінні вимоги. Використовуйте модульну архітектуру та дотримуйтесь принципів SOLID для забезпечення гнучкості та розширюваності.

Надійність – код повинен бути стійким до помилок і небезпек. Використовуйте відповідний обробник винятків, перевіряйте вхідні дані та робіть перевірки безпеки для уникнення можливих проблем.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		34

Код повинен бути написаний з урахуванням тестування. Використовуйте модульні тести для перевірки окремих частин коду та функціональні тести для перевірки повного функціоналу програми.

Код має бути легко розширюваним для додавання нового функціоналу або зміни існуючого. Використовуйте патерни проектування та добрі практики, щоб забезпечити гнучкість та розширюваність коду.

Наступним кроком є обрати середовище для розробки модуля. Існує декілька популярних середовищ для розробки на мові програмування Python. Ось декілька з них:

PyCharm—це інтегроване середовище розробки (IDE) для Python, розроблене компанією JetBrains. Воно надає широкі можливості для розробки, налагодження та тестування програм на Python. PyCharm має велику кількість корисних функцій, таких як автодоповнення, вбудований термінал, візуальний дебагер та підтримку системи керування версіями.

Anaconda—це платформа для наукового обчислення та розробки на Python. Вона включає в себе велику кількість наукових пакетів та інструментів, таких як NumPy, Pandas, Matplotlib та інші. Anaconda також має власне середовище розробки, що називається Anaconda Navigator, яке надає зручний інтерфейс для управління пакетами та проектами.

Jupyter Notebook—це веб-середовище для виконання коду Python в інтерактивному режимі. Jupyter Notebook дозволяє створювати та виконувати код у вигляді "клітинок", що дозволяє краще організувати та документувати процес розробки. Воно часто використовується для наукових обчислень, візуалізації даних та створення звітів.

Visual Studio Code—це легкий та розширюваний текстовий редактор, який має підтримку Python. Він надає широкий вибір розширень для покращення функціональності, таких як автодоповнення, вбудований термінал, візуальний дебагер та багато іншого. Visual Studio Code популярний серед розробників Python завдяки своїй простоті використання та потужним можливостям.

Для розробки даного модуля було використано середовище Visual Studio Code. Це середовище було обрано за рахунок зручного інтерфейсу і зручності в

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

підключені бібліотек і плагінів. За такими критеріями можна було також обрати PyCharm, але за рахунок більш частого використання середовища Visual Studio Code вибір був зроблений. Основним критерієм для вибору середовища став зручний інтерфейс і легка робота з середовищем. Інтерфейс середовища зображено на рис.3.1

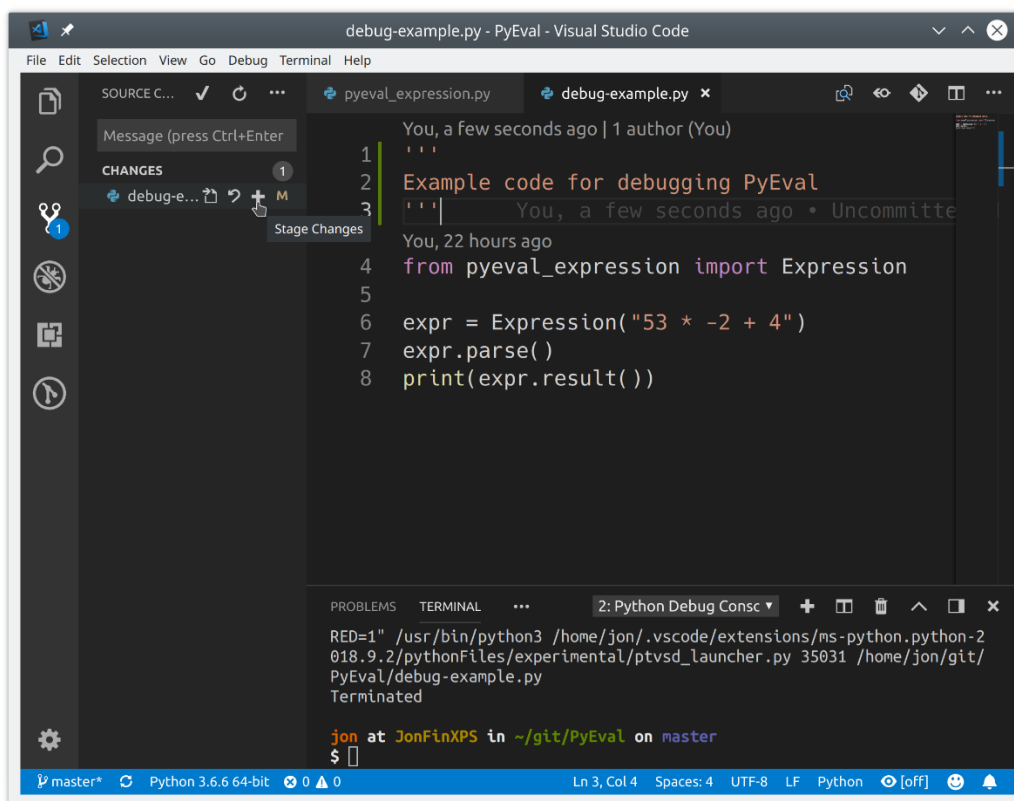


Рисунок 3.1–інтерфейс середовища Visual Studio Code

Згідно вище вказаних вимог і вибору середовища було обрано мову програмування Python.

Python є дуже підходящою мовою програмування для розробки модуля багатопотокового розпаралелення процесу опрацювання зображень. Python має розширений набір бібліотек для роботи з зображеннями, таких як PIL (Pillow), OpenCV, scikit-image та інші, що дозволяє зручно та ефективно обробляти зображення.

Крім того, Python має модуль `multiprocessing`, який був описаний в 2 розділі. Це дозволяє ефективно використовувати ресурси комп'ютера та прискорювати обробку зображень за допомогою багатопотоковості.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Python також має зрозумілу та просту синтаксис, що полегшує розробку та зрозуміння коду. Він є популярною мовою серед спільноти розробників, тому є велика кількість ресурсів та документації для вивчення та вирішення потенційних проблем.

Однак, варто зазначити, що Python може бути дещо повільнішим порівняно з низькорівневими мовами програмування, такими як C++ або Java. Це може впливати на продуктивність обробки зображень, особливо якщо ви маєте справу з великими обсягами даних. Однак, завдяки багатопотоковості та використанню ефективних бібліотек для обробки зображень, цей недолік можна зменшити.

Отже, Python є гнучкою та зручною мовою програмування для розробки модуля багатопотокового розпаралелення процесу опрацювання зображень, забезпечуючи широкі можливості для роботи з зображеннями та використання ресурсів комп'ютера.

Багатопотоковість є важливою концепцією у програмуванні і дозволяє виконувати багато завдань паралельно у межах одного програмного додатку. Java та C++ - популярні мови програмування, які надають підтримку багатопотоковості.

Java надає вбудовану підтримку багатопотоковості через пакет `java.lang.Thread`. Цей пакет дозволяє створювати та керувати потоками виконання у програмі. За допомогою класу `Thread`, ви можете створити новий потік та виконати код у цьому потоці. Java також надає ключове слово `synchronized` для синхронізації доступу до спільних ресурсів, що дозволяє уникнути конфліктів під час одночасного доступу декількох потоків до одного ресурсу.

C++ також має підтримку багатопотоковості через стандартну бібліотеку `std::thread`. Ця бібліотека дозволяє створювати та керувати потоками виконання у програмі. Ви можете створити новий потік за допомогою класу `std::thread` та виконати функцію у цьому потоці. Крім того, C++ надає механізми синхронізації, такі як `std::mutex` та `std::lock_guard`, для захисту спільних ресурсів від одночасного доступу з боку декількох потоків.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо некоректно використовувати багатопотоковість, можуть виникати проблеми, такі як гонки даних (data races) і блокування (deadlocks). Тому важливо правильно синхронізувати та керувати доступом до спільних ресурсів у багатопотокових програмах.

В обох мовах є багато інших інструментів та концепцій, пов'язаних з багатопотоковістю, таких як семафори, умовні змінні, бар'єри тощо. Розуміння цих концепцій допоможе вам ефективно використовувати багатопотоковість у своїх програмах і досягти більшої продуктивності та ефективності виконання.

Для реалізації модуля багатопотокового опрацювання зображень можна використовувати різні програмні засоби та бібліотеки. Ось кілька з них:

- бібліотека `OpenCV` - надає широкий спектр функцій для оброблення зображень, включаючи багатопотокові можливості.

Для багатопотокового опрацювання зображень зазвичай використовують бібліотеку OpenCV. OpenCV (Open Source Computer Vision Library) - це відкрита бібліотека комп'ютерного зору та машинного навчання, яка надає різноманітні функції для роботи з зображеннями та відео.

OpenCV має вбудовану підтримку багатопотокового опрацювання, що дозволяє паралельно обробляти декілька зображень або кадрів відео. Це може значно покращити швидкодію обробки зображень, особливо коли маємо справу з великою кількістю даних.

Для реалізації багатопотокового опрацювання зображень в OpenCV можна використовувати класи `cv::ParallelLoopBody` та `cv::parallel_for_`. Клас `cv::ParallelLoopBody` використовується для визначення функції, яка буде виконуватися паралельно для кожного елемента в циклі. Функція `cv::parallel_for_` дозволяє розподілити цикл обробки зображень між доступними потоками.

Основні переваги використання OpenCV для багатопотокового опрацювання зображень:

OpenCV надає оптимізовані алгоритми та операції, що дозволяють швидко опрацьовувати зображення на багатопотокових системах.

OpenCV має зрозумілий та зручний інтерфейс програмування, що дозволяє легко використовувати багатопотокове опрацювання зображень.

Дозволяє зручно розподіляти ресурси системи та використовувати доступні потоки для оптимальної обробки зображень.

Недоліки використання OpenCV для багатопотокового опрацювання зображень:

При використанні багатопотоковості необхідно правильно синхронізувати доступ до спільних ресурсів, щоб уникнути гонок та інших проблем з паралельним виконанням.

Швидкодія багатопотокового опрацювання залежить від кількості доступних потоків та апаратних можливостей системи. Деякі системи можуть мати обмеження на кількість одночасно виконуваних потоків.

У загальному, OpenCV є потужною бібліотекою для багатопотокового опрацювання зображень, яка надає різні можливості та функціональність для роботи з зображеннями та відео. Залежно від конкретних потреб та вимог проекту, OpenCV може бути ефективним інструментом для роботи модуля багатопотоковості опрацювання зображень.

- бібліотека `Pillow` - простий у використанні інструмент для роботи з зображеннями, має підтримку багатопотоковості.

Pillow є популярною бібліотекою для роботи з зображеннями в мові програмування Python. Вона надає широкий набір функцій для обробки та маніпуляції зображеннями, включаючи зміну розміру, налаштування якості, маніпуляції пікселями та багато іншого.

Щодо багатопотокового опрацювання зображень з використанням Pillow, важливо враховувати, що сама бібліотека не має вбудованої підтримки багатопотоковості. Однак, ви можете використовувати стандартні інструменти Python для багатопотоковості, такі як модуль `concurrent.futures`, для паралельного опрацювання зображень з використанням Pillow.

Основні переваги використання Pillow для багатопотокового опрацювання зображень:

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Pillow має простий та зрозумілий інтерфейс програмування, що дозволяє легко використовувати його функції для обробки зображень у багатопотоковому середовищі.

Надає широкий спектр функцій для роботи з зображеннями, що дозволяє виконувати різноманітні операції, такі як зміна розміру, обрізка, фільтрація та інші, у багатопотоковому режимі.

Pillow підтримує багато різних форматів зображень, включаючи JPEG, PNG, TIFF, BMP та інші, що дозволяє легко працювати з різними типами зображень у багатопотоковому середовищі.

Недоліки використання Pillow для багатопотокового опрацювання зображень:

Потрібно використовувати сторонні інструменти та методи для забезпечення безпечної багатопотоковості при роботі з Pillow.

Багатопотокове опрацювання зображень може вимагати багато ресурсів, особливо якщо обробляються великі зображення або виконуються складні операції. Важливо забезпечити достатні ресурси для ефективного виконання багатопотокових завдань з використанням Pillow.

Враховуючи переваги та недоліки, Pillow може бути хорошим вибором для багатопотокового опрацювання зображень, якщо враховані вимоги та обмеження використання бібліотеки.

- бібліотека `NumPy` - має потужні функції для маніпуляції зображеннями та обчислень, може використовуватися з багатопотоковістю.

NumPy є однією з найпопулярніших бібліотек для наукових обчислень у мові програмування Python. Вона надає ефективні структури даних, такі як масиви та матриці, і велику кількість функцій для операцій з ними. Проте, за замовчуванням, NumPy не має вбудованої підтримки багатопотоковості.

Однак, існують різні способи використання NumPy з багатопотоковістю для покращення продуктивності обчислень. Ось деякі з них:

NumPy надає функції, такі як `numpy.vectorize` та `numpy.frompyfunc`, які дозволяють векторизувати функції та виконувати їх паралельно на багатоядерних системах.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

NumPy може бути поєднаний з іншими бібліотеками, такими як Dask або mpi4py, для виконання обчислень на розподілених системах або використання MPI для паралельного обчислення на кластерах.

Ви також можете використовувати багатопотокові бібліотеки, такі як threading або multiprocessing, для розпаралелювання обчислень з використанням NumPy. Наприклад, ви можете розділити масив на менші частини та обчислювати їх у окремих потоках або процесах.

Переваги використання NumPy для багатопотоковості:

NumPy відомий своєю високою швидкістю обчислень, що робить його привабливим вибором для багатопотокового обчислення, де продуктивність є ключовим фактором.

NumPy має простий та зрозумілий інтерфейс програмування, що дозволяє легко виконувати операції з масивами та матрицями у багатопотоковому середовищі.

Недоліки використання NumPy для багатопотоковості:

Якщо не правильно виконати синхронізацію потоків або процесів, можуть виникнути конфлікти, що можуть призвести до некоректних результатів обчислень.

Деякі алгоритми та обчислення можуть мати обмеження щодо паралельного виконання через апаратні особливості системи. В таких випадках, багатопотоковість може не приводити до значного покращення продуктивності.

Загалом, використання NumPy для багатопотокового обчислення може бути ефективним, але вимагає обережного планування та управління потоками для досягнення найкращої продуктивності та уникнення конфліктів.

- бібліотека 'Java Advanced Imaging (JAI)' - надає набір інструментів для оброблення зображень в Java, включаючи підтримку багатопотоковості.

Java Advanced Imaging (JAI) є бібліотекою для роботи з обробкою зображень у мові програмування Java. Вона надає розширений набір функцій для обробки та маніпулювання зображеннями, включаючи підтримку багатопотоковості.

Основні переваги використання JAI для багатопотокового обчислення:

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

JAІ надає можливість паралельної обробки зображень, дозволяючи виконувати кілька операцій одночасно на різних ділянках зображення. Це забезпечує покращену продуктивність та швидкість обробки.

Недоліки використання JAІ для багатопотокового обчислення:

JAІ є досить складною бібліотекою з великою кількістю функцій та налаштувань. Це може ускладнити розробку та налагодження багатопотокового коду.

Як і в інших бібліотеках для багатопотокового обчислення, продуктивність JAІ може бути обмежена апаратними характеристиками системи, такими як кількість ядер процесора та доступна оперативна пам'ять.

Загалом, JAІ є потужною бібліотекою для роботи з обробкою зображень у багатопотоковому середовищі. Вона надає широкий функціонал та можливості для паралельного обчислення, але вимагає від розробників вміння працювати зі складними функціями та вирішувати проблеми багатопотоковості.

Ці програмні засоби надають різні можливості для реалізації багатопотокового опрацювання зображень у вашому модулі. Вибір конкретного засобу залежатиме від ваших потреб, мови програмування, яку ви використовуєте, та функціональності, яку ви хочете реалізувати.

3.2 Реалізація модуля багатопотокового опрацювання зображення

Після реалізації всіх передніх кроків можна перейти до початку розробки модуля. Перше що має бути виконано це встановлення всіх бібліотек і залежностей. Цю дію виконуємо за допомогою команди «pip install Pillow».

Ця команда застосовується для встановлення такої бібліотеки як PIL (Python Imaging Library), за допомогою цієї бібліотеки ви можете здійснювати різноманітні операції з зображеннями. Більшість з цих операцій використана в нашому модулі. А саме:

- Відкриття /закриття зображення;

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

- Зміна розміру зображення зі збереженням пропорцій і без
- Поворот зображення
- Зміна яскравості
- Обрізка зображення.

Приклад частини цього модуля зображено на рисунку 3.2

```
def process_image(image_path):
    # Виконуємо обробку зображення тут
    image = Image.open(image_path)

    # Зміна розміру зображення
    resized_image = image.resize((400, 600))

    blurred_image = resized_image.filter(ImageFilter.BLUR)
    # Зберігаємо оброблене зображення
    processed_image_path = "processed_" + image_path
    blurred_image.save(processed_image_path)
```

Рисунок 3.2– код обробки зображення

Наступною частиною коду є головна функція яка має список зображень, створює пул процесів, запускає обробку зображення, завершує обробку зображень. Дана функція зображена Рисунок 3.3

```
if __name__ == '__main__':
    # Список зображень, які потрібно обробити
    image_paths = ["image1.jpg", "image2.jpg"]

    # Створюємо пул процесів з кількістю потоків, рівною кількості яде
    pool = multiprocessing.Pool()

    # Запускаємо обробку зображень в багатопотоковому режимі
    pool.map(process_image, image_paths)

    # Завершуємо роботу пулу потоків
    pool.close()
    pool.join()
```

Рисунок 3.3– Головна функція

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		43

Також в кодї застосовано ще інші операції обробки зображення, але більшість функцій зображена на Рисунок 3.2. якщо описувати код то можна сказати що основну роботу виконує модуль multiprocessing за допомогою бібліотеки PIL

Основною частиною коду я вважаю розробку інтерфейсу для даної програми Рисунок 3.4 відображає частину коду для розробки інтерфейсу програми

```
if __name__ == '__main__':  
    image_paths = []  
  
    root = tk.Tk()  
    root.title("Image Processing")  
  
    select_button = tk.Button(root, text="Вибрати зображення", command=select_image)  
    select_button.pack()  
  
    image_listbox = tk.Listbox(root)  
    image_listbox.pack()  
  
    process_button = tk.Button(root, text="Обробити зображення", command=process_images)  
    process_button.pack()  
  
    root.mainloop()
```

Рисунок 3.4- Код для відображення інтерфейсу

3.3 Тестування роботи модуля багатопотокового опрацювання

Перше що відображається при тестуванні ми бачимо інтерфейс користувача Рисунок 3.5

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

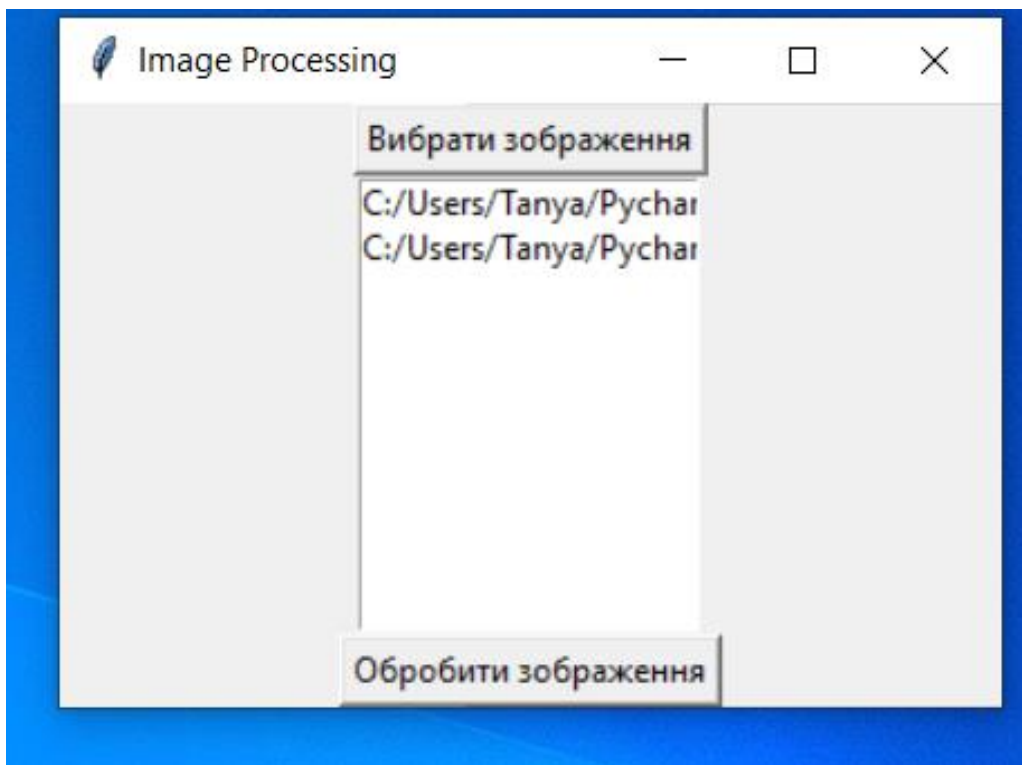


Рисунок 3.5–інтерфейс програми

Наступною дією користувача є вибір зображення, в даній програмі можна обирати більше ніж одне зображення. Для виборі зображення натискаємо кнопку «Вибрати зображення», після чого нам відкривається папка з фото

Рисунок 3.5

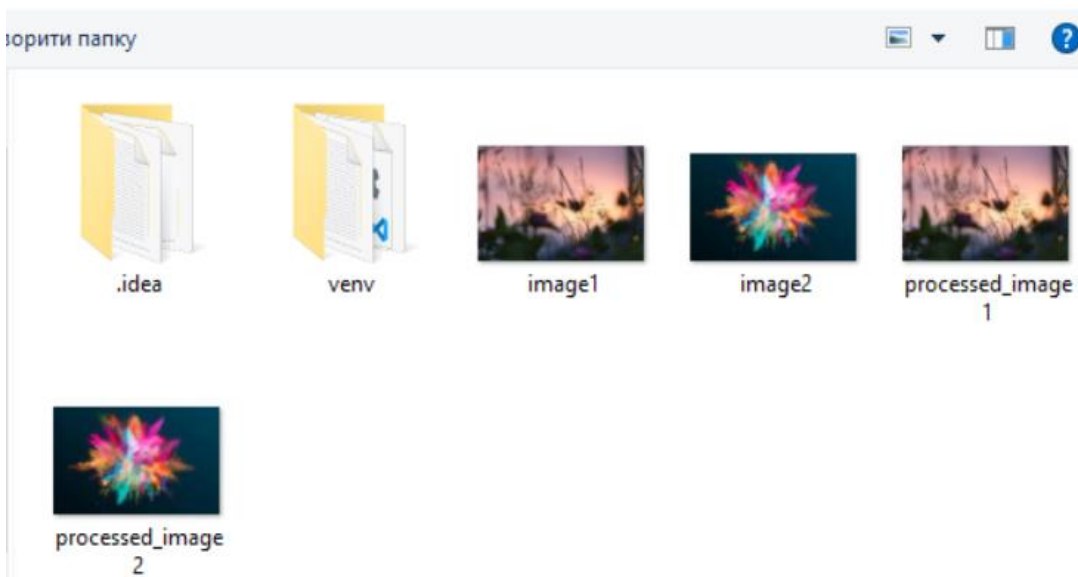


Рисунок 3.5–Папка з рисунками

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

В цій папці будуть знаходитись як неопрацьованні так і опрацьованні зображення.

Вибравши зображення які треба опрацьовати їх шлях відобразиться в інтерфейсі програми. Рисунок 3.6

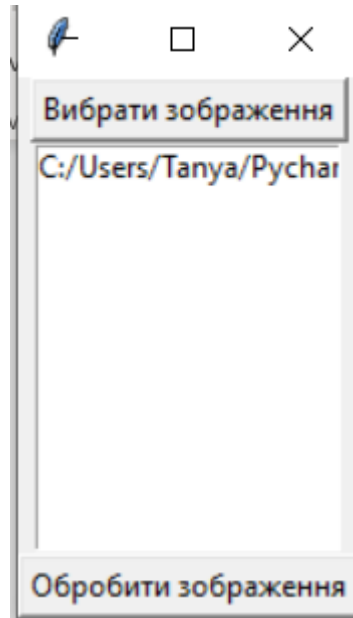


Рисунок 3.6–шлях до зображення

Після чого буде опрацьоване на рисунку 3.7 зображенно не опрацьоване зображення а на рисунку 3.8 вже опрацьоване.

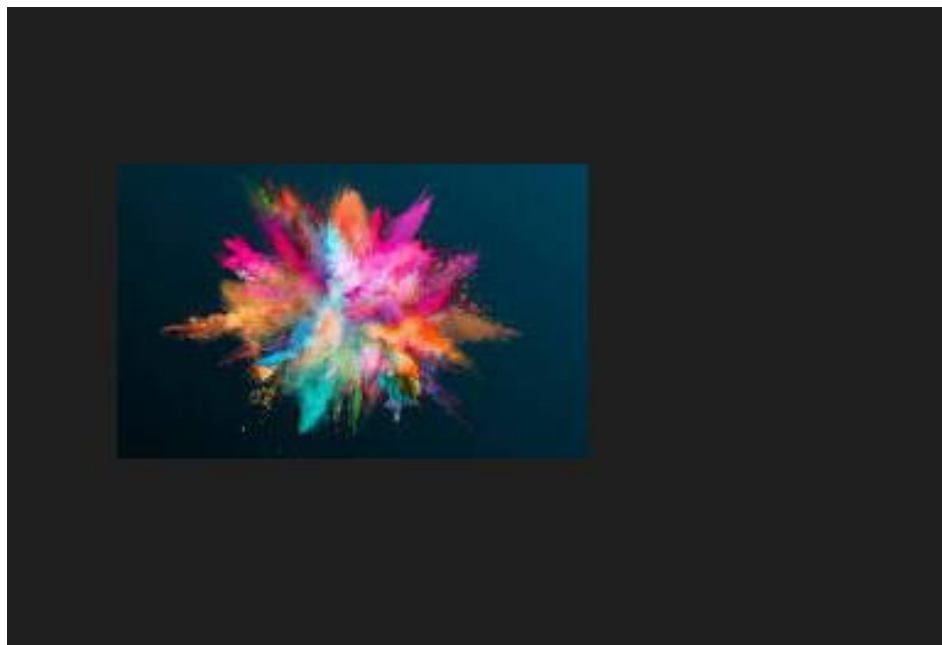


Рисунок 3.7–неопрацьоване зображення

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

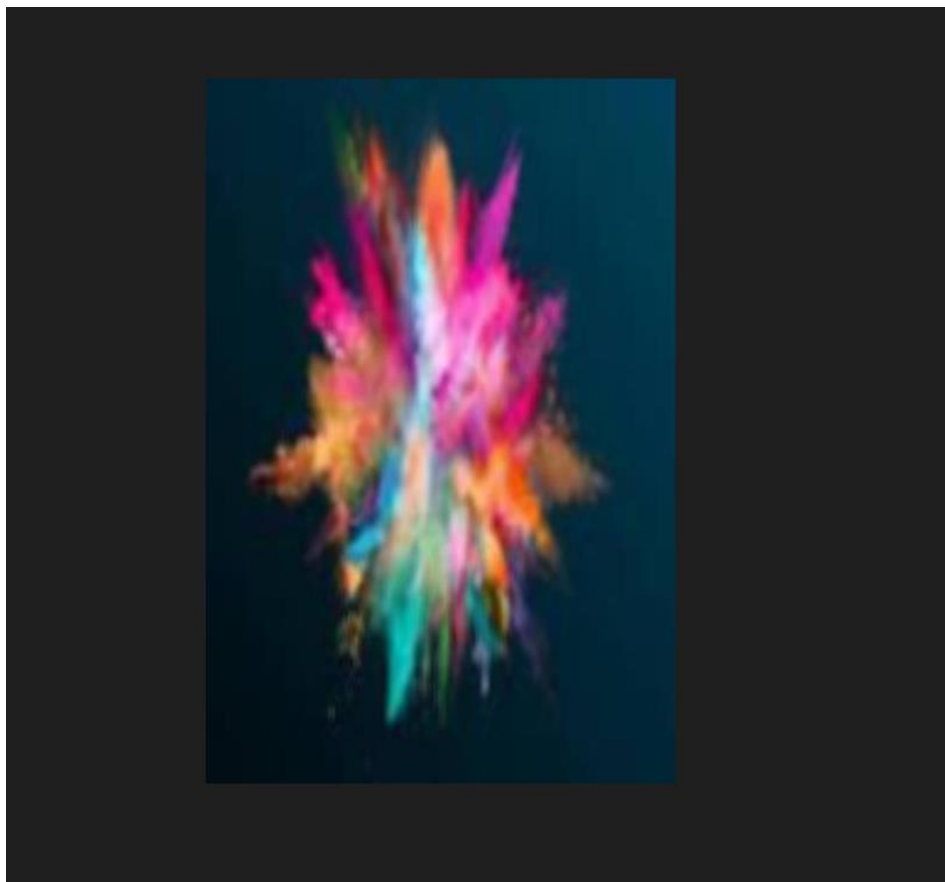


Рисунок 3.8—опрацьоване зображення

Результати проведених досліджень наведено на рисунку 3.9

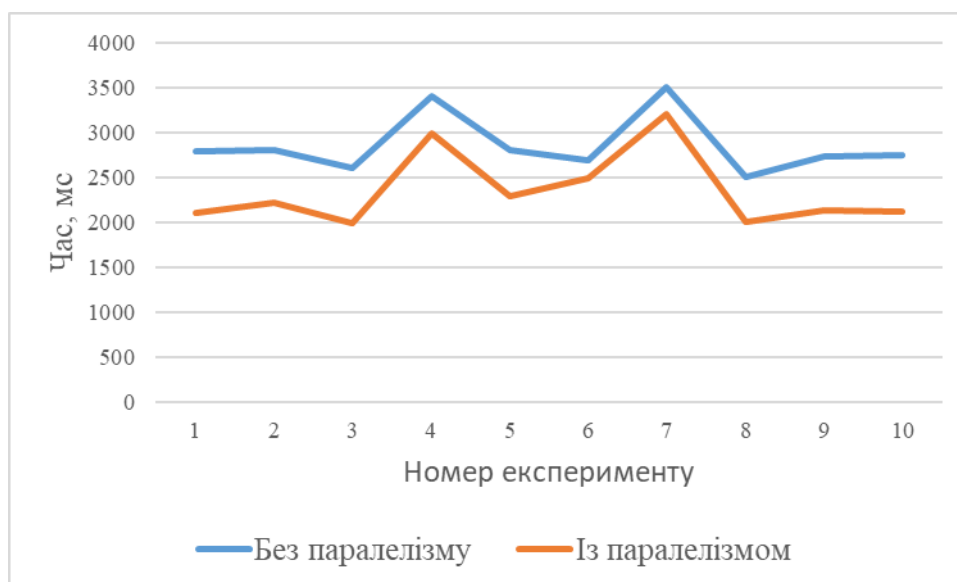


Рисунок 3.9 – Порівняльний аналіз проведених експериментів

Усереднений результат часу виконання експериментів із використанням паралелізму та без паралелізму наведено на рисунку 3.10

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

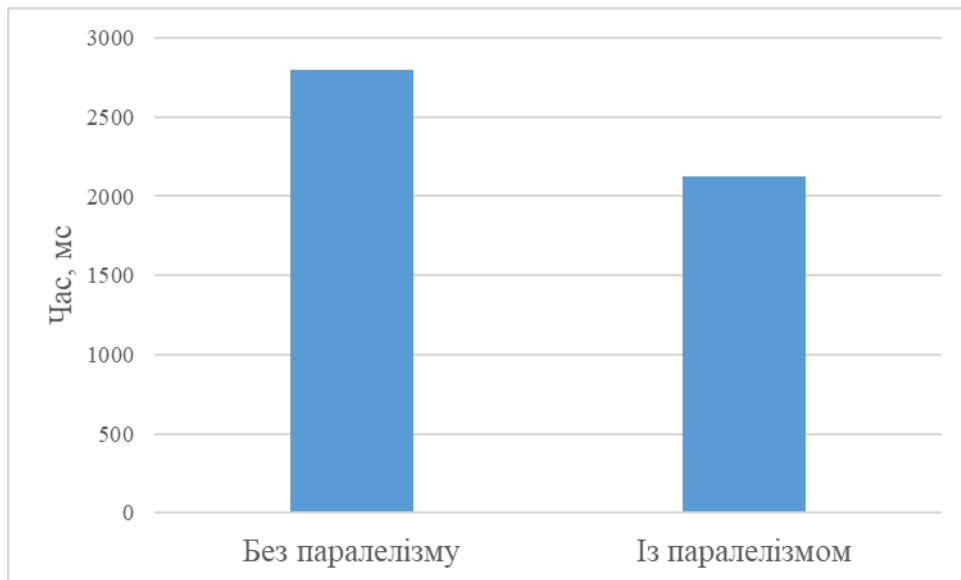


Рисунок 3.10 – Усереднений результат часу виконання експериментів із використанням паралелізму та без паралелізму

В 3 розділі було проаналізовано різні мови програмування, середовища розробки модуля багатопотокової обробки зображення. Було описано виконані алгоритми роботи проекту і детально описано кожен з них. В кінці було детально описано і графічно представлено роботу розробленого модуля, графічно було відображено кожен крок тестування модуля, а також по-кроково описано роботу з модулем.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

4.1 Розрахунок витрат на розробку програмного забезпечення

Техніко-економічне обґрунтування є невід'ємною частиною розробки будь-якого проекту, включаючи модуль керування засобами робочого столу жестами. Ця технологія набула все більшої популярності в останні роки, оскільки вона дозволяє користувачам ефективно та зручно управляти своїм комп'ютером за допомогою жестів рук.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників (*Воп*);
- витрати на відрахування у спеціальні державні фонди (*Вф*);
- витрати на закупні вироби (*Пв*);
- витрати на придбання спецобладнання для проведення експериментальних робіт (*Об*);
- накладні витрати (*Н*);
- інші витрати (*Ів*).

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування.

У розробці програмного забезпечення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу.

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

№ п/п	Посада виконавців	Час розробки, год	Погодинна оплата	Витрати на оплату
1	Керівник	16	93	1488 грн
2	Студент	400	93	37200

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{OH} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.1)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

Середньо годинна ставка працівника може бути розрахована за формулою:

$$C_{ij} = \frac{C_{ij}^0 (1+h)}{PЧ_i}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ - місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Воп= 38688 (грн)

Крім того, слід визначити відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

$$ВФ = 20.5 * 38688 = 7931 \text{ (грн.)}$$

Розрахунок матеріальних витрат

Матеріальні витрати — це вартість витрачених матеріалів, малоцінних та швидкозношуваних предметів на виробництво продукції, робіт або послуг, а також матеріалів, витрачених на адміністративні, збутові та інші потреби.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою:

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Звідси, витрати на матеріальні ресурси дорівнюватимуть:

$$B_M = 332 \text{ грн.}$$

Проведені розрахунки занесемо у таблицю 4.2.

Таблиця 4.2 - Зведені розрахунки покупних виробів

п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп	200	1	200	20,0	220,0
3	Зшивання диплому	ч.р	110	1	100,0	10,0	110,0
Разом							332

Обчислення накладних витрат

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати.

Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H_B = 1,5 \cdot B_{OP}, \quad (4.4)$$

де H_B – накладні витрати.

$$H_B = 1,5 \cdot 38688 = 58032 \text{ грн.}$$

Обчислення інших витрат

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 38688 \cdot 0.1 = 3868.8 \text{ грн}$$

Складання кошторису витрат та визначення собівартості

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{OP} + B_{\Phi} + B_{ПВ} + H + I \quad (4.5)$$

$$K_1 = 38688 + 7931 + 332 + 58032 + 3868.8 = 1147,9 \text{ (грн.)}$$

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.6)$$

де $S_{м.г.}$ - вартість однієї машино-години роботи ПК, грн./год.

$t_{від}$ - комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2,5 години. Вартість години роботи комп'ютера дорівнює 5,2 грн. Тому:

$$K_2 = 5,2 \cdot 75 = 390 \text{ грн.}$$

На основі отриманих даних складаємо кошторис витрат на розробку програмного забезпечення. Проведені розрахунки занесемо у таблицю 4.3.

Таблиця 4.3- Кошторис витрат на розробку програмного забезпечення

№	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	38688
2	Відрахування у спеціальні державні фонди	7931
3	Витрати на куповані вироби	332
4	Накладні витрати	58032
5	Інші витрати	3868.8
6	Витрати на відлагодження і дослідну експлуатацію програмного продукту	390,0
Разом		18353

В таблиці 4.3 було проведено загальні розрахунки всіх витрат на розробку програмного забезпечення.

4.2 Визначення прогнозованої (договірної) ціни КС

Величина можливої (договірної) ціни КС повинна визначатися з урахуванням ефективності, якості і термінів її виконання на рівні, що

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		53

відповідає економічним інтересам замовника (споживача) і виконавця (підрядника). Договірна ціна (Дц) для прикладних КС розраховується за формулою:

$$Дц = ВКС * (1 + p) = 18353 * 1,25 = 22941, \quad (4.7)$$

де ВКС - витрати на розробку КС (з таблиці 4.3), грн.;

p - середній рівень рентабельності КС, % (приймається в розмірі 20-30%).

Розраховану можливу ціну КС варто порівняти із цінами на аналогічні розробки, що існують на ринку ІТ продуктів.

Витрати, пов'язані з розробкою програми на ПК розраховуються:

$$В_{ПК} = T_{ПК} * t_c, \quad (4.8)$$

Де

$T_{ПК}$ - час використання ПК для розробки програми,

$C_{ПК}$ - собівартість машинного часу обчислювальної техніки (розраховує бухгалтерія підприємства).

$$В_{ПК} = T_{ПК} * 0,09 = 3295 * 0,09 = 3295$$

Собівартість однієї години роботи ПК дорівнює:

$$C_{ПК} = \frac{V_e}{\Phi_{ПК}} = \frac{1500}{1797} = 0,83 \quad (4.9)$$

де V_e - річні поточні витрати на експлуатацію ПК,

$\Phi_{ПК}$ - річний фонд часу корисної роботи ПК.

Розрахуємо річний фонд часу роботи ПК. Визначивши дійсний річний фонд часу ПК у годинах, отримаємо можливість оцінити собівартість годин машинного часу.

Дійсний річний фонд часу ПК дорівнює:

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

$$\Phi_{\text{д}} = N_{\text{р}} - (\Phi_{\text{м}} + \Phi_{\text{річ}}) = 1976 - (5 + 144) = 1827 \quad (4.10)$$

де $\Phi_{\text{м}}$ - місячний фонд часу на профілактику і ремонт ПК (час профілактики щомісячно – 5 годин),

$\Phi_{\text{річ}}$ - річний фонд часу на профілактику і ремонт ПК (час профілактики щорічно – 6 діб)

Річні поточні витрати на експлуатацію програмного забезпечення визначаються за формулою:

$$B_{\text{ПК}} = B_{\text{Е}} + B_{\text{А}_{\text{р}}} + B_{\text{РЕМ}_{\text{р}}} + B_{\text{ДК}} + B_{\text{І}} \quad (4.11)$$

де $B_{\text{А}_{\text{р}}}$ – річні відрахування на амортизацію,

$B_{\text{Е}}$ – річні витрати на електроенергію для ПК,

$B_{\text{РЕМ}_{\text{р}}}$ – річні витрати на ремонт ПК,

$B_{\text{ДК}}$ – річні витрати на додаткові комплектуючі ПК,

$B_{\text{І}}$ – інші витрати.

Суму річних амортизаційних відрахувань визначаємо за такою формулою:

$$B_{\text{А}_{\text{р}}} = \text{Ц}_{\text{ПК}} * N_{\text{А}} = 28000 * 0,15 = 4200 \quad (4.12)$$

де $\text{Ц}_{\text{ПК}}$ – балансова вартість ПК,

$N_{\text{А}}$ – норма амортизаційних відрахувань (дорівнює 15% у квартал).

Балансову вартість ПК розраховуємо за формулою:

$$\text{Ц}_{\text{ПК}} = \text{Ц}_{\text{р}} * (1 + K_{\text{УН}}) = 25000 * (1 + 0,12) = 28000 \quad (4.13)$$

де $\text{Ц}_{\text{р}}$ – ринкова вартість ПК,

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

$K_{УН}$ – коефіцієнт, що враховує витрати на установку й налагодження ПК(приймається рівним 12%).

Витрати на електроенергію, що споживає ПК, визначаємо за формулою:

$$V_E = P_{ПК} * \Phi_{ПК} * C_E * K_{ІВ} = \frac{0,06кВт}{год} * 1797 * 1,68 * 0,9 = 13391,25 \quad (4.14)$$

де $P_{ПК}$ – паспортна потужність ПК,

$\Phi_{ПК}$ – річний фонд корисного часу роботи ПК,

C_E – вартість 1 кВт/год електроенергії,

$K_{ІВ}$ – коефіцієнт інтенсивного використання ПК (0,7 - 1).

Таким чином, розрахункове значення витрат на електроенергію, що споживає ПК, складає:

- витрати на поточний і профілактичний ремонт (приймаються рівними 6% від вартості ПК):

$$V_{РЕМ,р} = C_{ПК} * 0,06 = 28000 * 0,06 = 1680 \quad (4.15)$$

- витрати на додаткові комплектуючі – витрати необхідні для забезпечення експлуатації ПК (приймаються рівними 2% від вартості ПК):

$$V_{ДК,р} = C_{ПК} * 0,02 = 28000 * 0,02 = 560 \quad (4.16)$$

- інші витрати, тобто непрямі витрати пов'язані з експлуатацією ПК (приймаються рівними 5-10% від вартості ПК):

$$V_{І,р} = C_{ПК} * 0,05 = 28000 * 0,05 = 1400 \quad (4.17)$$

У ході розробки програмного комплексу ПК використовується на таких етапах програмування:

– написання програми за готовою схемою алгоритму;

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

- налагодження програми на ПК;
- підготовки документації по задачі.

Таким чином, витрати машинного часу склали ($t_{\text{маш}}$):

$$t_{\text{маш}} = t_n + t_{\text{налаг}}^k + t_d = 303 + 2222,22 + 648,14 = 3173,36 \quad (4.18)$$

Витрати на оплату машинного часу розраховуємо за формулою:

$$V_{\text{маш}} = t_{\text{маш}} * C_{\text{ПК}} = 3173,36 * 0,83 = 2638 \quad (4.19)$$

Загальні витрати на розробку програмного комплексу складають:

$$V_{\text{заг}} = V_{\text{оп}} + V_{\text{маш}} = 75151 + 2638 = 77789 \quad (4.20)$$

4.3 Визначення економічної ефективності

За міжнародним стандартам для оцінки ефективності розробки ПЗ застосовують такі показники:

- внутрішня норма дохідності;
- чистий приведений дохід;
- рентабельність;
- термін окупності.

Показник внутрішньої дохідності характеризує величину чистого прибутку (чистого валового доходу), що припадає на одиницю інвестиційних

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		57

вкладень у кожному часовому інтервалі життєвого циклу проекту.

Розрахунок

цього показника виконується за такою формулою:

$$\sum_{i=0}^T \frac{D_i}{(1+q)^i} - \sum_{i=0}^T \frac{K_i}{(1+q)^i} = 0 \quad (4.21)$$

де D_i - дохід (прибуток) у i -му періоді;

K_i - інвестиційні вкладення в i -му періоді з урахуванням інфляційних процесів;

i - періоди виконання і впровадження проекту;

T - загальний період (тривалість) життєвого циклу проекту;

q - показник внутрішньої норми дохідності.

Показник інвестиційних вкладень з урахуванням інфляційних процесів обчислюємо за формулою:

$$K_i = \varphi_i + R_i. \quad (4.22)$$

де φ_i - коефіцієнт інфляції на поточний період;

R_i - інвестиційні платежі в i -му періоді (капітальні вкладення).

Дохід від розробки ПЗ у i -му періоді розраховуємо за формулою:

$$D_i = J_i(B_i - C_i). \quad (4.23)$$

де B_i - ціна продажу програмного продукту в i -му періоді;

C_i - собівартість програмного продукту (фактично дорівнює сумі витрат на розробку ПЗ);

J_i - кількість ПЗ.

Вартість продажу розробленого продукту розраховують за формулою:

$$B_i = B_{\text{заг}} \left(1 + \frac{p}{100}\right). \quad (4.24)$$

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		58

де p - середній рівень рентабельності на поточний період.

Показник рентабельності інвестицій. У практиці середнього бізнесу для визначення ефективності проектних рішень широко використовується.

$$p = \frac{\sum_{i=0}^T \frac{D_i}{(1+q)^i}}{\sum_{i=0}^T \frac{K_i}{(1+q)^i}} - 1 > 0. \quad (4.25)$$

У ринкових умовах при ціновій політиці, що змінюється, показник терміну окупності є одним з головних для підприємств. Він визначається на основі величини капітальних витрат по періодах розробки програмного продукту та величини фактичних чи прогнозних доходів:

$$\sum_{i=0}^T K_i = \sum_{i=0}^T D_i \quad (4.26)$$

де T - термін окупності,

D_i - дохід (прибуток) у поточному періоді,

K_i - капітальні витрати у поточному періоді.

Економічна ефективність полягає у відношенні результату від розробленого програмного продукту до затрачених ресурсів:

$$E = \frac{D_i}{B_{\text{зат}}} \quad (4.27)$$

$$E_p = 5056 / 33712,75 = 0,15$$

Тоді термін окупності можна розрахувати за такою формулою:

$$T = \frac{1}{E} \quad (4.28)$$

$$T = 1 / 0,15 = 6,5$$

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		59

Враховуючи основні економічні показники, що стосуються розробки програмного продукту, можна зробити висновок, щодо доцільності запропонованої розробки. Якщо отримано суттєвий економічний ефект від розробки програмного продукту, а термін окупності капітальних вкладень не більший 10 років, то така розробка є економічно вигідною та конкурентоздатною на ринку подібних ІТ продуктів.

Отже в цій частині дипломного проекту здійснюється розрахунок витрат на розробку програмного забезпечення. Проведено порівняння з існуючими аналогами, проаналізовано відібрані дані, які показали економічну доцільність розробки модуля способу управління робочим столом за допомогою жестів.

Виходячи з проведених економічних аргументів, програмне забезпечення є конкурентоспроможним.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		60

ВИСНОВОК

В результаті виконання бакалаврської роботи було розроблено модуль багатопотокового розпаралелення процесу опрацювання зображень. Програмний модуль був розроблений за допомогою мови програмування Python і також за допомогою різноманітних бібліотек. Для розробки використовувалось середовище програмування VS Code.

1. Було проаналізовано різні мови програмування для реалізації модуля. Також розглянули готові рішення і різні види програм які використовують даний модуль. Було розглянуто різні види інтерфейсів для розробки рішення а разом із цим було вирішено додати додатковий функціонал до власного модуля.

2. Розроблений програмний продукт в майбутньому можна розширювати великою кількістю нових функцій адже було обрано мову програмування яка є легкою в розумінні і в розробці.

3. В майбутньому даний вид програмного забезпечення буде застосовуватись в медицині, комп'ютерній графіці і виробництві. Ці сфери в особливості медицина є важливим для буденного життя людей.

4. Загалом бакалаврська робота виконує всі поставлені завдання, тобто описує основні вимоги до розробки модуль багатопотокового розпаралелення процесу опрацювання зображень, які згодом використовуються для реалізації самого проекту. Також проведено порівняльний аналіз існуючих програм, що стосуються теми, а також описані їх переваги та недоліки, які в подальшому використовуються для вдосконалення проекту та його функціональності. Проаналізовано алгоритм реалізації та функціонування існуючого модуль багатопотокового розпаралелення процесу опрацювання зображень, що також має велике значення для подальшої реалізації проекту.

Нарешті завершено повний дизайн модуля багатопотокового розпаралелення процесу опрацювання зображень, виконано всі етапи тестування проекту та описано всі функції, які використовуються в проекті.

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Водзінська Віталіна Миколаївна Модуль багатопотокового розпаралелення процесу опрацювання зображень . VII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 23 травня 2023 р. Тернопіль. Україна - с. 8.

2. Yeeleng S. Vang, Zhen Chen, Xiaohui Xie (2018) Deep Learning Framework for Multi-class Breast Cancer Histology Image Classification. Computer Vision and Pattern Recognition - CIAR2018 workshop <https://doi.org/10.48550/arXiv.1802.00931>

3. Ramirez-Quintana, J., Acosta-Lara, I., Ramirez-Alonso, G., Chacon-Murguia, M., Corral-Saenz, A. (2022). A Lightweight Convolutional Neural Network for Breast Cancer Diagnosis with Histology Images. Pattern Recognition. MCP R 2022. Lecture Notes in Computer Science, vol 13264. Springer, Cham. https://doi.org/10.1007/978-3-031-07750-0_30

4. Y. Li, J. Wu and Q. Wu, (2019) Classification of Breast Cancer Histology Images Using Multi-Size and Discriminative Patches Based on Deep Learning, in IEEE Access, vol. 7, pp. 21400-21408 <https://ieeexplore.ieee.org/document/8636921>

5. Peizhen Xie , Ke Zuo, Jie Liu , Mingliang Chen, Shuang Zhao (2021) Interpretable Diagnosis for Whole-Slide Melanoma Histology Images Using Convolutional Neural Network. Journal of Healthcare Engineering - vol. 2021, Article ID 8396438, 7 pages. <https://doi.org/10.1155/2021/8396438>

6. Araújo T, Aresta G, Castro E, Rouco J, Aguiar P, Eloy C, et al. (2017) Classification of breast cancer histology images using Convolutional Neural Networks. PLoS ONE 12(6): e0177544. <https://doi.org/10.1371/journal.pone.0177544>

7. O. Berezsky, O. Pitsun, N. Batryn, K. Berezska, N. Savka and T. Dolynyuk (2018) Image Segmentation Metric-Based Adaptive Method, IEEE Second

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

International Conference on Data Stream Mining & Processing (DSMP), pp. 554-557. <https://ieeexplore.ieee.org/document/8478579>

8. Talha Qaiser, Yee-Wah Tsang, Daiki Taniyama, Naoya Sakamoto, Kazuaki Nakane, David Epstein, Nasir Rajpoot (2019) Fast and accurate tumor segmentation of histology images using persistent homology and deep convolutional features, Medical Image Analysis, vol. 55. pp. 1-14 <https://doi.org/10.1016/j.media.2019.03.014>

9. Hongming Xu, Lina Liu, Xiujuan Lei, Mrinal Mandal, Cheng Lu. (2021) An unsupervised method for histological image segmentation based on tissue cluster level graph cut. Computerized Medical Imaging and Graphics, vol. 93. <https://doi.org/10.1016/j.compmedimag.2021.101974>

10. Mohamed Amgad, Habiba Elfandy, Hagar Hussein (2019) Structured crowdsourcing enables convolutional segmentation of histology images. Bioinformatics, Volume 35, Issue 18, 15 September 2019, Pages 3461–3467, <https://doi.org/10.1093/bioinformatics/btz083>

11. N. Fujima, K. Hirata, T. Shiga, R. Li. (2018) Integrating quantitative morphological and intratumoural textural characteristics in FDG-PET for the prediction of prognosis in pharynx squamous cell carcinoma patients. Clinical Radiology, Volume 73, Issue 12. <https://doi.org/10.1016/j.crad.2018.08.011>

12. O. Berezsky, O. Pitsun, L. Dubchak, K. Berezka, T. Dolynyuk and B. Derish, (2020) "Cytological Images Clustering of Breast Pathologies," 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), pp. 62-65, <https://ieeexplore.ieee.org/document/9321867>

13. O. Berezsky, O. Pitsun, (2021) Application of linear regression method for analysis of cytological images quantitative characteristics", Ukrainian Journal of Information Technology, vol.3, no.1, pp.73. <https://doi.org/10.23939/ujit2021.03.073>

14. Shadi AlZu'bi, Mohammed Shehab, Mahmoud Al-Ayyoub, Yaser Jararweh, Brij Gupta. (2020) Parallel implementation for 3D medical volume fuzzy segmentation, Pattern Recognition Letters, vol 130, pp. 312-318 <https://doi.org/10.1016/j.patrec.2018.07.026>

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

15. Jianfang Cao, Lichao Chen, Min Wang, Yun Tian, (2018) "Implementing a Parallel Image Edge Detection Algorithm Based on the Otsu-Canny Operator on the Hadoop Platform", Computational Intelligence and Neuroscience, vol. 2018, Article ID 3598284, 12 pages. <https://doi.org/10.1155/2018/3598284>

16. Gu, Haiyan, Yanshun Han, Yi Yang, Haitao Li, Zhengjun Liu, Uwe Soergel, Thomas Blaschke, and Shiyong Cui. 2018. "An Efficient Parallel Multi-Scale Segmentation Method for Remote Sensing Imagery" Remote Sensing 10, no. 4: 590. <https://doi.org/10.3390/rs10040590>

17. Huang, B., Xue, J., Lu, K., Tan, Y., Zhao, Y. (2021). MPNet: Multi-scale Parallel Codec Net for Medical Image Segmentation. In: Fang, L., Chen, Y., Zhai, G., Wang, J., Wang, R., Dong, W. (eds) Artificial Intelligence. CICA 2021. Lecture Notes in Computer Science(), vol 13069. Springer, Cham. https://doi.org/10.1007/978-3-030-93046-2_42

18. Roels, J., De Vylder, J., Saeys, Y., Goossens, B., Philips, W. (2016). Decreasing Time Consumption of Microscopy Image Segmentation Through Parallel Processing on the GPU. In: Blanc-Talon, J., Distant, C., Philips, W., Popescu, D., Scheunders, P. (eds) Advanced Concepts for Intelligent Vision Systems. ACIVS 2016. Lecture Notes in Computer Science(), vol 10016. Springer, Cham. https://doi.org/10.1007/978-3-319-48680-2_14

19. Hangün, B. , Eyecioğlu, Ö. "Performance Comparison Between OpenCV Built in CPU and GPU Functions on Image Processing Operations". International Journal of Engineering Science and Application 1 (2017): 34-41

20. D. Demirović, E. Skejić and A. Šerifović–Trbalić, (2018) Performance of Some Image Processing Algorithms in Tensorflow, 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 2018, pp. 1-4, <https://ieeexplore.ieee.org/document/8439714/>

21. Willian Barreiros, Alba C.M.A. Melo, Jun Kong, Renato Ferreira, Tahsin M. Kurc, Joel H. Saltz, George Teodoro (2022) Efficient microscopy image analysis on CPU-GPU systems with cost-aware irregular data partitioning, Journal of Parallel and Distributed Computing, Volume 164, pp 40-54, <https://doi.org/10.1016/j.jpdc.2022.02.004>

					КР.КІ. 8351493.00.00.000.ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

22. V. Gancheva, "Parallel Multithreaded Medical Images Filtering," 2021 International Conference on Computational Science and Computational Intelligence (CSCI), 2021, pp. 1788-1793, <https://ieeexplore.ieee.org/document/9799119>

23. Qihan Wang;Jianming Pang;Feng Yue;Shudan Yang- SWPy: Python Numerical Computing Library Optimization for Domestic Many-core Processors 2022 P100

24. Timothy G. Mattson;Todd A. Anderson;Giorgis Georgakoudis - PyOMP: Multithreaded Parallel Programming in Python 2022 P122

25. Feng He;Xuran Hu;Shuren Liu;Tao Li;Kaifeng Zhu;Xin Bao;Chao Jiang - Algorithm for Improving Processor Utilization in Multi-core Processor Environment by Python Language 2021 P23

26. Myungsun Kim - Guaranteeing That Multilevel Prioritized DNN Models on an Embedded GPU Have Inference Performance Proportional to Respective Priorities 2021 P 10-43

27. Nikolai N. Smolyaninov;Denis S. Kharenko - Bottlenecks in Implementation of the Mode Decomposition Algorithm Based on Phase-Only Spatial Light Modulator 2022 P 200-210

28. Joseph Kready;Shishila Awung Shimray;Muhammad Nihal Hussain;Nitin Agarwal - YouTube Data Collection Using Parallel Processing 2020

29. Mansi Agarwal;Chandan Kumar;Pramod Kumar;Swadesh Verma - AI layer reference generator engine for early enablement of workloads over multiple frameworks 2019 P 26-31

30. Saud Wasly - ERAsim: A Flexible Python-based Architectural Modeling and Simulation Framework 2021 P 45-46

31. Haolei Ye;Shu Liu;Wei Xiao;Eric C. McCreath;Xin Zhuang Zhang - A Multi-Threaded Accelerated Drug Combination Pathway Network Implementation 2020 P 49-53

32. Qihan Wang;Jianming Pang;Feng Yue;Shudan Yang- SWPy: Python Numerical Computing Library Optimization for Domestic Many-core Processors 2022 P100

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		65

33. Timothy G. Mattson;Todd A. Anderson;Giorgis Georgakoudis - PyOMP: Multithreaded Parallel Programming in Python 2022 P122
34. Feng He;Xuran Hu;Shuren Liu;Tao Li;Kaifeng Zhu;Xin Bao;Chao Jiang - Algorithm for Improving Processor Utilization in Multi-core Processor Environment by Python Language 2021 P23
35. Roels, J., De Vylder, J., Saeys, Y., Goossens, B., Philips, W. (2016). Decreasing Time Consumption of Microscopy Image Segmentation Through Parallel Processing on the GPU. In: Blanc-Talon, J., Distant, C., Philips, W., Popescu, D., Scheunders, P. (eds) Advanced Concepts for Intelligent Vision Systems. ACIVS 2016. Lecture Notes in Computer Science(), vol 10016. Springer, Cham. https://doi.org/10.1007/978-3-319-48680-2_14
36. Hangün, B. , Eyecioğlu, Ö. "Performance Comparison Between OpenCV Built in CPU and GPU Functions on Image Processing Operations". International Journal of Engineering Science and Application 1 (2017): 34-41
37. D. Demirović, E. Skejić and A. Šerifović–Trbalić, (2018) Performance of Some Image Processing Algorithms in Tensorflow, 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 2018, pp. 1-4, <https://ieeexplore.ieee.org/document/8439714/>
38. Sandya Subramanian;Bryan Tseng;Riccardo Barbieri;Emery N Brown - Unsupervised Machine Learning Methods for Artifact Removal in Electrodermal Activity 2021 P23-24
39. Yuhan Liu;Qian Sun;Yong Tang;Ying Li;Weiqi Jiang;Jianing Wu - Virtual reality system for industrial training 2020 P 90-95
40. Ruoxi Guo;Jiahao Cui;Wanru Zhao;Shuai Li;Aimin Hao - Hand-by-Hand Mentor: An AR based Training System for Piano Performance 2021 P67-76
41. Ramazan Özgür Doğan;Hülya Doğan;Cemal Köse Virtual mouse control with hand gesture information extraction and tracking 2018 P38-41
42. Anton Franzluebbers;Changying Li;Andrew Paterson;Kyle Johnsen - Virtual Reality Point Cloud Annotation 2020 P 90-96

					КР.КІ. 8351493.00.00.000.ІІЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

43. Xingyu Chen;Yufeng Liu;Yajiao Dong;Xiong Zhang;Chongyang Ma;Yanmin Xiong;Yuan Zhang;Xiaoyan Guo - MobRecon: Mobile-Friendly Hand Mesh Reconstruction from Monocular Image 2022 P 190-193

44. Mathis Petrovich;Michael J. Black;Gül Varol Action-Conditioned 3D Human Motion Synthesis with Transformer VAE 2020 P 34

45. Alim Misbullah;Laina Farsiah;Nazaruddin;Furqan Hermawan Voice-Zikr: A Speech Recognition System Implementation for Hands-Free Zikr Based on Deep Learning 2022 P 49-69

46. Xingle Feng;Wenshan Hu;Hong Zhou;Guo-Ping Liu Design of Virtual Thermal Process Control Framework for NCSLab 2021 P 140-143

47. Zhimin Gao;Yongji Cao;Hengxu Zhang;Hao Qin;Xiaoning Zhang;Changgang Li - Cooperative Control Strategy for Virtual Inertia Based on Fuzzy Control 2022 P 67

48. Jingjie Liao;Hai Xiao Gong;Hongyan Lu;Yuling Yang;Jiliang Lin A Control Strategy for Passing and Receiving in Virtual Human Soccer 2022 P 98

49. Linxiu Sha;Changfeng Cheng Design and Development of Virtual Experiment Network Platform for Rig Control 2020 P 104

50. Arun Kumar;Supriya.P. Panda - A Survey: How Python Pitches in IT-World 2020 P130-132

51. Claudio David López;Miloš Cvetković;Peter Palensky - Enhancing PowerFactory Dynamic Models with Python for Rapid Prototyping 2018 P 48-96

					КР.КІ. 8351493.00.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67