

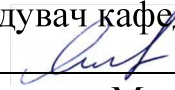
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**Національний університет кораблебудування імені адмірала Макарова**

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ


к.т.н, доц. Михелєв І.Л.

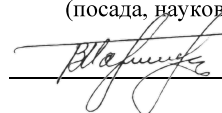
"___" "___" 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА**на здобуття ступеня вищої освіти "магістр"****на тему: Дослідження ефективності алгоритмів OCR для обробки тексту з
зображень у iOS додатках**

Виконав: студент групи 6141 м

 **Дробот В.В.**
(підпис)

Керівник роботи:

доцент кафедри ІУСТ, к.т.н.,
(посада, науковий ступінь вчене звання) **Партас В.К.**
(підпис)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

В. Ажищев

к.т.н, доц. Ажищев В.Ф.

"__" "_____" 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "магістр"

Студенту Дробот Володимир Вадимович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Дослідження ефективності алгоритмів OCR для обробки тексту з зображень у iOS додатках»

Керівник роботи: доцент кафедри ІУСТ, к.т.н Партаєв Віктор Кирилович

Затверджені наказом ректора № 1055-уч. від "08" жовтня 2024 року.

2. Термін подання роботи: 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

1. Аналіз предметної області 2. Дослідження методів та засобів оптичного розпізнавання 3. Практична реалізація та оптимізація осг моделі на прикладі додатку для перекладу 4. Економічна ефективність розробленого програмного забезпечення 5. Охорона праці 6. Охорона навколишнього середовища

5. Перелік презентаційних матеріалів

1. Актуальність та мета дослідження 2. Основні завдання дослідження 3. Сучасні технології OCR 4. Методи та алгоритми OCR 5. Діаграма прецедентів 6. Діаграма послідовностей 7. Технології оптимізації для iOS 8. Результати тестування. 9. Висновки .

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Партас Віктор Кирилович	05.09.2023	22.12.2023
2	Партас Віктор Кирилович	23.12.2023	15.06.2024
3	Партас Віктор Кирилович	15.06.2024	25.09.2024
4	Партас Віктор Кирилович	25.09.2024	14.11.2024
5	Партас Віктор Кирилович	25.09.2024	14.11.2024
6	Партас Віктор Кирилович	25.09.2024	14.11.2024

7. Дата видачі завдання 05 вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

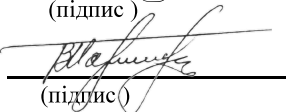
№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2023	
2	Розробка концепції	15.03.2024	
3	Розділ з розробки проектних рішень по системі (та її частинах)	15.06.2024	
4	Реалізація проекту та розробка робочої документації	25.09.2024	
5	Обчислення економічної ефективності	14.11.2024	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2024	
7	Оформлення пояснювальної записки	14.11.2024	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2024	
9	Подання магістерської роботи рецензенту	20.11.2024	
10	Подання магістерської роботи на державний захист	15.12.2024	

Студент


 (підпис)
Дробот В.В.

(ПІБ)

Керівник роботи


 (підпис)
Партас В.К.

(ПІБ)

АНОТАЦІЯ

Магістерська робота «Дослідження ефективності алгоритмів OCR для обробки тексту з зображень у iOS-додатках» за спеціальністю 122 – Комп'ютерні науки.

Робота присвячена **теоретичному та практичному аналізу алгоритмів оптичного розпізнавання символів (OCR)** для автоматизації процесів обробки текстової інформації з графічних матеріалів у мобільних додатках під керуванням iOS.

Об'єкт дослідження – процеси обробки тексту, отриманого з графічних зображень за допомогою OCR-технологій у мобільних додатках.

Предмет дослідження – методи та алгоритми OCR, що використовуються для аналізу та розпізнавання тексту з зображень у iOS-середовищі.

Метою роботи є підвищення ефективності використання OCR-алгоритмів у iOS-додатках за рахунок вдосконалення методів розпізнавання тексту та оптимізації їх продуктивності.

Завдання дослідження:

1. Проаналізувати сучасні технології та алгоритми OCR для розпізнавання тексту з зображень;
2. Визначити критерії ефективності та якісні показники роботи OCR-алгоритмів у iOS-додатках;
3. Розробити методи оптимізації обраних алгоритмів OCR для підвищення точності та швидкості обробки;
4. Реалізувати та протестувати програмний прототип для підтвердження отриманих теоретичних результатів;
5. Провести техніко-економічний аналіз та оцінити вплив впровадження вдосконалених OCR-технологій на практичну діяльність.

Методи дослідження: методи математичного моделювання, системного аналізу, порівняльного тестування алгоритмів OCR, а також методи програмної реалізації та оптимізації коду.

Наукова новизна полягає у вдосконаленні підходів до оптимізації роботи OCR-алгоритмів у середовищі iOS, що дозволить отримувати більш точні та швидкі результати обробки тексту зі зображень.

Наукове значення роботи полягає у формуванні методологічних основ ефективного застосування OCR-технологій у мобільних додатках, що сприятиме розширенню можливостей автоматизації роботи з текстовою інформацією.

Практичне значення роботи полягає у розробці програмного прототипу та практичних рекомендацій щодо використання вдосконалених OCR-алгоритмів в iOS-додатках, що дозволить підвищити якість та продуктивність систем, орієнтованих на обробку текстових даних.

Ключові слова: OCR (оптичне розпізнавання символів), iOS, Swift, Vision, машинне навчання, нейронні мережі, CoreML, глибоке навчання, оптимізація, рукописний текст, обробка зображень, методи.

ANNOTATION

Master's thesis "Research into the effectiveness of OCR algorithms for processing text from images in iOS applications" in specialty 122 - Computer Science.

The work is dedicated to the theoretical and practical analysis of optical character recognition (OCR) algorithms aimed at automating the processing of textual information extracted from graphical materials in mobile applications running on iOS.

Object of the research: The processes of text processing obtained from graphical images using OCR technologies in mobile applications.

Subject of the research: OCR methods and algorithms employed for analyzing and recognizing text from images in the iOS environment.

Purpose of the study: To increase the efficiency of using OCR algorithms in iOS applications by improving text recognition methods and optimizing their performance.

Research objectives:

1. Analyze state-of-the-art OCR technologies and algorithms for recognizing text from images.
2. Determine the efficiency criteria and quality indicators for the performance of OCR algorithms in iOS applications.
3. Develop optimization methods for selected OCR algorithms to improve accuracy and processing speed.
4. Implement and test a software prototype to validate the theoretical findings.
5. Conduct a technical and economic analysis and assess the impact of implementing enhanced OCR technologies on practical activities.

Research methods: Mathematical modeling, systems analysis, comparative testing of OCR algorithms, as well as methods of software implementation and code optimization.

Scientific novelty: The work improves approaches to optimizing the performance of OCR algorithms in the iOS environment, enabling more accurate and faster text processing results from images.

Scientific significance: The study establishes methodological foundations for the effective application of OCR technologies in mobile applications, contributing to expanded capabilities in automating the handling of textual information.

Practical significance: The development of a software prototype and practical recommendations for using enhanced OCR algorithms in iOS applications will increase the quality and performance of systems oriented toward processing textual data.

Keywords: OCR (Optical Character Recognition), iOS, Swift, Vision, Machine Learning, Neural Networks, CoreML, Deep Learning, Optimization, Handwritten Text, Image Processing, Methods.

ЗМІСТ

Вступ.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1Опис предметної галузі.....	11
1.2 Вимоги до засобів для перекладу тексту з використанням OCR та їх недоліки	16
1.3 Актуальність, ціль та задачі дослідження	19
1.4 Висновок	21
РОЗДІЛ 2. Дослідження існуючих засобів та методів оптичного розпізнавання текстів.	22
2.1 Математичний аналіз існуючих засобів оптичного розпізнавання.....	22
2.2 Порівняний аналіз існуючих методів	25
2.3 Висновок	28
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОПТИМІЗАЦІЯ OCR МОДЕЛІ НА ПРИКЛАДІ ДОДАТКУ ДЛЯ ПЕРЕКЛАДУ	30
3.1 Використані технології для оптимізації OCR Vision на Swift	30
3.2 Побудова діаграм варіантів використання.....	34
3.3 Проведення дій з прискорення завантаження камери для OCR на Swift	36
3.4 Проведення дій з прискорення роботи Vision в мобільному додатку	40
3.5 Результат роботи над оптимізацією та заміри.....	48
3.6 Висновок	56
РОЗДІЛ 4. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	58
4.1 Вступна частина	58
4.2 Розрахунок витрат на створення й експлуатацію програми	58
4.3 Розрахунок економічної ефективності	61
4.4 Розрахунок часу окупності програми	61
4.5 Висновок	61
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	62
5.1 Вступ.....	62
5.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу	62
5.2.1 Організація робочого місця.....	62

5.2.2 Мікроклімат робочої зони програміста	63
5.2.3 Освітлення робочого місця	64
5.2.4 Вплив шуму на програміста.....	66
5.2.5 Виробничі випромінювання.....	66
5.2.6 Електробезпека. Статична електрика.....	68
5.3 Ергономіка та організація робочого місця.....	69
5.4 Висновки	70
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	71
6.1 Забруднення навколишнього середовища надлишками праці.....	71
ВИСНОВКИ	74

Вступ

Текст є невід’ємною частиною нашого життя, оскільки є одним із головних інструментів передачі інформації між людьми. Це твердження стосується також фотографій — з розвитком технологій майже кожен має доступ до можливості зробити фото. Але що робити, коли ці два типи інформації змішуються — фото з текстом? Раніше, окрім варіанту переписати текст вручну, не було інших варіантів. Для вирішення цієї проблеми існує технологія OCR. Optical Character Recognition (OCR) — це перетворення тексту з фото у коди символів для подальшого використання. Хоча розробка даної технології розпочалася досить давно, протягом усього цього часу вона розвивалась, і з’явилася можливість її використання безпосередньо на мобільних пристроях.

З розвитком мобільних технологій та зростанням попиту на швидкий доступ до інформації, можливість автоматичного розпізнавання тексту з фотографій стає особливо актуальною. Це дозволяє користувачам ефективно обробляти документи, зберігати інформацію та ділитися нею без зайвих зусиль. Окрім цього, технологія OCR може бути корисною в різних сферах, таких як освіта, бізнес, медицина та інші, де необхідно швидко та точно перетворювати зображення тексту у цифровий формат. Впровадження OCR у мобільні додатки для платформи iOS сприяє підвищенню продуктивності та зручності використання цих додатків, що, у свою чергу, покращує загальний користувацький досвід.

Поставлені задачі :

1. Провести аналіз сучасного стану засобів (програмного забезпечення) для перекладу документів та обґрунтувати необхідність їх удосконалення.
2. Дослідити існуючі засоби та методи оптичного розпізнавання тексту.
3. Розробити технічне удосконалення технології OCR на мові Swift.
4. Провести тестування та оцінку ефективності розробленого додатку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної галузі

З розвитком цифрових технологій та збільшенням обсягів інформації, що потребує перекладу та аналізу фото з текстом, значно зросла потреба у використанні програмного забезпечення для автоматизації цього процесу. Сьогодні існує велика кількість систем, які забезпечують автоматичний переклад тексту з різних мов. Проте, для отримання якісного результату часто потрібні додаткові засоби для попередньої обробки тексту, зокрема засоби оптичного розпізнавання тексту (Optical Character Recognition, OCR). OCR-технології використовуються для перетворення тексту зображень у цифровий формат, що дозволяє редагувати, аналізувати та перекладати його. Це особливо важливо для обробки документів, які зберігаються у вигляді сканів, фотографій або PDF-файлів, де текст представлений у вигляді графічних об'єктів.

Дана технологія використовується у таких сегментах бізнесу як мобільні додатки для сканування документів: CamScanner та Adobe Scan

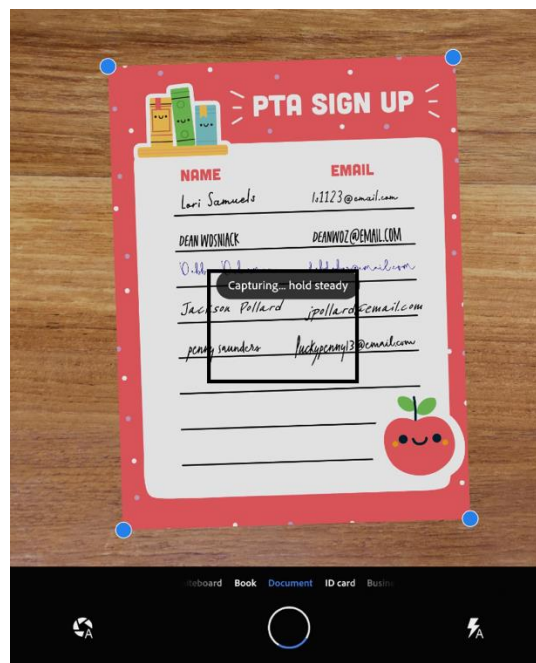


Рисунок 1.1 - Adobe Scan

Також OCR технології використовуються у сервісах фінансування та банкінгу.

Ці популярні додатки дозволяють користувачам сканувати документи за допомогою камери смартфона, автоматично розпізнавати текст та зберігати його у PDF-форматі.

Наразі OCR технології використовують у таких сферах життя для автоматичного зчитування інформації з чеків, квитанцій та платіжних документів, що полегшує користувачам введення даних це можна побачити в таких сервісах як Revolut або Monobank. Також банки можуть набагато швидше та якісніше обробляти великі обсяги документів без участі людей.

Велика частиною медичного світу стала використовувати дану технологію у своїй галузі - як Practo та Epocrates вони оцифровують текст медичних записів, результатів аналізів та рецептів, що дозволяє лікарям швидше знаходити необхідну інформацію та покращує якість медичних послуг. Одною із тенденцій світової економіки стали магазини без продавців і цей функціонал потрібен для автоматичного сканування штрих-кодів, етикеток та товарних накладних, що дозволяє швидко відстежувати запаси та обробляти замовлення. В Amazon Go та в подібних безкасових магазинах використовують OCR та штучний інтелект для автоматизації процесів оплати та контролю товарів.



Рисунок 1.2 - магазин без продавців Amazon go

Оптичне розпізнавання символів (OCR) є сучасною технологією, яка дозволяє автоматизувати процес оцифровування тексту з паперових документів, зображень або сканів. Її ключова перевага полягає у здатності швидко і точно розпізнавати текст, конвертуючи його у цифровий формат. Це робить OCR незамінним інструментом для багатьох бізнесів, державних установ та освітніх організацій, що прагнуть оптимізувати свої процеси управління інформацією. Завдяки своїй універсальності, технологія знаходить застосування у різноманітних сферах – від обробки фінансових документів до оцифровування архівів.

Однією з головних переваг OCR є значне підвищення продуктивності та економія часу. Автоматизація введення даних скорочує години, які раніше витрачалися на ручне введення тексту, дозволяючи компаніям обробляти великі обсяги документів набагато швидше. Це особливо актуально для бізнесів, що щоденно працюють з рахунками, контрактами або звітами. У масштабах великих організацій така ефективність не лише оптимізує процеси, але й економить значні фінансові ресурси.

Ще одна важлива перевага – мінімізація людських помилок. Ручна робота, особливо з великими обсягами інформації, нерідко призводить до помилок у введенні даних. OCR забезпечує високу точність розпізнавання тексту, що суттєво знижує ризик помилок. Це критично важливо для фінансових установ, медичних закладів або юридичних компаній, де навіть незначна похибка може мати серйозні наслідки.

Технологія також сприяє цифровій трансформації бізнес-процесів. Використання OCR дозволяє перейти на безпаперовий формат роботи, що підвищує ефективність управління даними, спрощує доступ до них та економить ресурси. Оцифровані документи легко шукати, аналізувати та використовувати у різних додатках. У бібліотеках, архівах та наукових організаціях це сприяє збереженню інформації та полегшує доступ до неї, роблячи знання доступними для ширшого кола користувачів.

Сучасні технології OCR (оптичного розпізнавання символів) представлені різними інструментами, що відповідають різним потребам та сценаріям використання. Одним із найпопулярніших безкоштовних рішень є Tesseract OCR, розроблений Google. Цей інструмент відкритий і широко використовується завдяки підтримці багатьох мов і можливості налаштування для конкретних задач. Він використовує алгоритми глибокого навчання для підвищення точності розпізнавання тексту, що робить його ефективним вибором для проєктів з обмеженим бюджетом або ентузіастів, які створюють власні рішення.

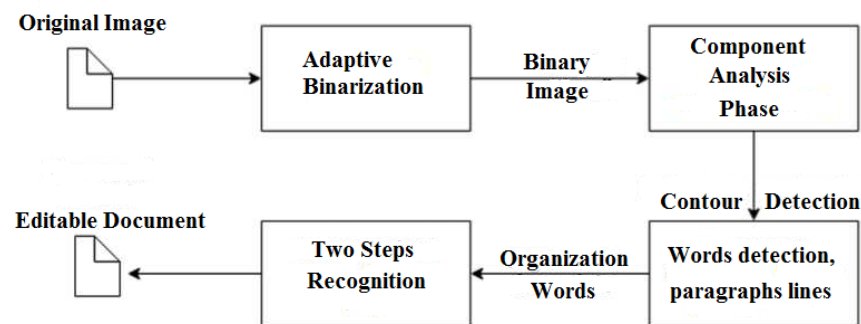


Рисунок 1.3 - Архітектура Tesseract OCR

Для тих, хто потребує високої точності та розширених функцій, відмінним вибором є ABBYY FineReader. Це комерційне програмне забезпечення, яке відоме своїми точними алгоритмами розпізнавання та підтримкою величезної кількості мов, включаючи складні шрифти та рукописні тексти. ABBYY FineReader також пропонує інтегровані інструменти для редагування та обробки документів, що робить його ідеальним для великих організацій, які працюють із юридичними, фінансовими чи іншими критично важливими документами.

Хмарні сервіси, такі як Google Cloud Vision та Microsoft Azure Cognitive Services, надають можливість використовувати штучний інтелект для аналізу зображень і розпізнавання тексту. Ці сервіси не лише обробляють текст, але й можуть ідентифікувати об'єкти, визначати контекст зображення та навіть перекладати текст. Завдяки своїй хмарній архітектурі, вони забезпечують масштабованість і дозволяють інтегрувати OCR-функціонал у веб-додатки або мобільні застосунки без необхідності локальної обробки даних.

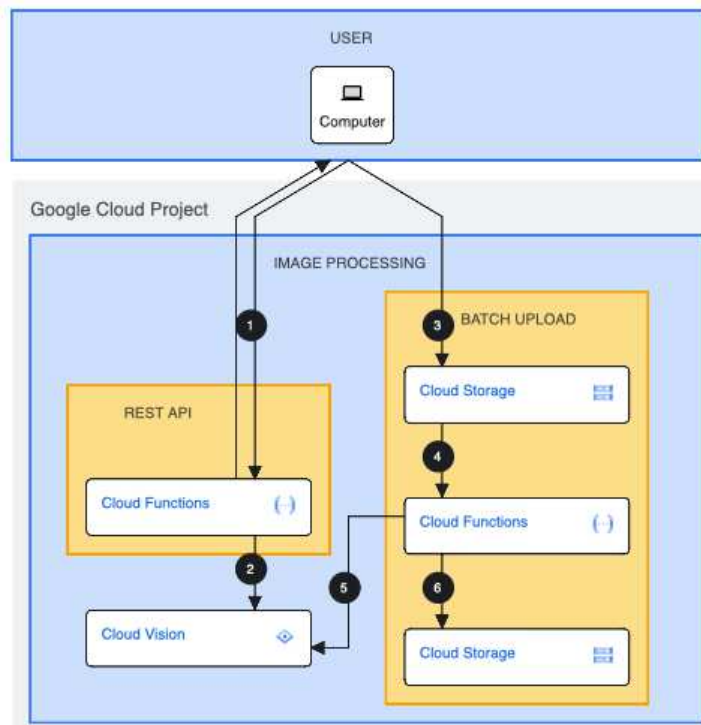


Рисунок 1.4 - Архітектура Google Cloud Vision

Apple Core ML, частина екосистеми Apple, пропонує розробникам засоби для інтеграції OCR у додатки на iOS і macOS. Завдяки машинному навчанню та оптимізації для пристроїв Apple, цей інструмент дозволяє створювати швидкі та ефективні рішення для розпізнавання тексту. Core ML чудово інтегрується з іншими інструментами Apple, такими як Vision Framework, забезпечуючи зручність у роботі з фото- та відеоконтентом. Це робить його ідеальним для створення користувацьких додатків, орієнтованих на iPhone та iPad.

Таким чином, використання OCR у мобільних додатках та інфраструктурі дозволяє автоматизувати багато рутинних процесів, підвищуючи ефективність та точність обробки даних. Інтеграція цих технологій у iOS-додатки на мові Swift відкриває нові можливості для бізнесу та користувачів, дозволяючи їм обробляти інформацію прямо на мобільних пристроях без необхідності використання додаткового обладнання або підключення до Інтернету.

Для розробки ефективного програмного забезпечення, що використовує OCR для перекладу тексту, необхідно враховувати специфічні вимоги до алгоритмів та

інструментів розпізнавання тексту. Це забезпечить високу точність та продуктивність роботи навіть на мобільних платформах з обмеженими ресурсами. Давайте детальніше розглянемо ці вимоги та проаналізуємо недоліки існуючих рішень.

1.2 Вимоги до засобів для перекладу тексту з використанням OCR та їх недоліки

Технологія OCR (оптичного розпізнавання символів) стала важливим інструментом у багатьох сферах життя та бізнесу. Вона дозволяє автоматизувати процеси розпізнавання тексту з різних джерел, таких як документи, зображення чи відео. Проте ефективність OCR залежить від багатьох чинників, зокрема точності розпізнавання, швидкості роботи, підтримки різних мов, автономності та здатності інтегруватися в мобільні платформи. Розглянемо детальніше кожен з цих аспектів.

Одна з найважливіших характеристик OCR – це точність розпізнавання тексту. Технологія повинна справлятися з різними викликами, такими як низька якість зображень, нерівне освітлення, викривлення чи наявність відблисків. Особливо важливо, щоб система могла розпізнавати текст, написаний складними шрифтами, курсивом або прописними літерами. Водночас проблеми часто виникають під час роботи з текстом на фонах, які містять графічні елементи, водяні знаки чи кольорові елементи. Забезпечення точності в таких випадках є важливим завданням для OCR-інструментів.

Швидкість розпізнавання тексту є критично важливою, особливо для мобільних додатків, де користувачі очікують миттєвих результатів. Система повинна працювати в режимі реального часу або з мінімальними затримками, особливо якщо текст розпізнається “на льоту”, наприклад, під час сканування документів чи перекладу тексту з фото. Обмежені обчислювальні ресурси на мобільних пристроях вимагають оптимізації алгоритмів, щоб забезпечити швидку обробку навіть на менш потужних процесорах.

Глобальне використання OCR неможливе без підтримки різних мов, зокрема тих, які використовують нелатинські алфавіти, такі як арабська, китайська чи японська. Окрім того, система має успішно розпізнавати кирилицю, ієрогліфи та інші складні системи письма. Ця функція дозволяє розширити можливості програмного

забезпечення, зробивши його корисним для міжнародних компаній, багатомовних платформ та локалізованих мобільних додатків.

Можливість працювати автономно без підключення до Інтернету є важливою перевагою для OCR-інструментів. Це необхідно для захисту конфіденційності даних або у випадках, коли доступ до мережі обмежений. Програмне забезпечення має підтримувати локальну обробку зображень, залишаючись компактним і не надмірно споживаючи ресурси пристрою, такі як пам'ять чи батарея.

Для забезпечення оптимальної роботи на мобільних пристроях OCR-додатки повинні бути адаптовані до обмежень операційних систем iOS та Android. Це включає мінімізацію енергоспоживання та використання платформи специфічних можливостей, таких як Apple Vision Framework або ML Kit від Google. Інтеграція нативних API та бібліотек дозволяє досягти максимальної ефективності та швидкості роботи. Зважаючи на сучасні виклики, такі як низька якість зображень, потреба у швидкій обробці та глобальне використання різних мов, OCR вже демонструє значні досягнення, які покращують бізнес-процеси та користувацький досвід(Рис.1.5)

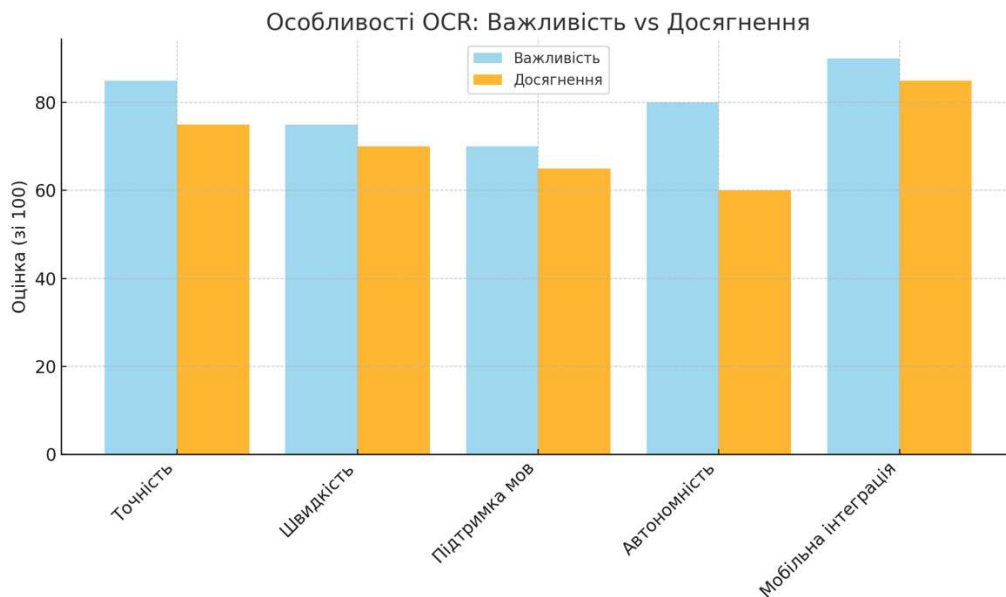


Рисунок 1.5 – графік залежності важливості і досягнення

Але окрім переваг використання, важливо також ознайомитись і з недоліками з якими можна зустрітись під час використання. Багато сучасних рішень OCR стикаються з труднощами, коли текст знаходиться на зображеннях низької якості або

розміщений на складному фоні. Наприклад, фотографії з поганим освітленням, низькою роздільною здатністю чи викривленнями часто призводять до некоректного розпізнавання тексту. Рукописний текст або шрифти з нестандартним стилем також залишаються серйозними викликами для більшості OCR-алгоритмів, що обмежує їх універсальність у реальних сценаріях використання.

Комерційні OCR-продукти, такі як ABBYY FineReader чи Microsoft Azure OCR, забезпечують високу точність та широкий спектр функцій, але їх вартість може бути непосильною для малих компаній чи індивідуальних розробників. Ліцензії на такі рішення часто коштують дорого і можуть бути недоступними для стартапів або дослідницьких проєктів, які працюють із обмеженим бюджетом. Це робить використання цих технологій викликом для організацій з малими ресурсами.(таблиця 1)

Продукт	Тип ліцензії	Вартість (USD)	Примітки
ABBYY FineReader PDF Standard	Постійна ліцензія	\$199	Одноразова оплата за ліцензію на одного користувача.
ABBYY FineReader PDF Corporate	Постійна ліцензія	\$299	Розширені функції для бізнесу, одноразова оплата.
Microsoft Azure Computer Vision	Підписка (щомісячна)	Залежить від використання	Ціна залежить від кількості оброблених зображень; починається від \$1 за 1,000 транзакцій.
Google Cloud Vision	Підписка (щомісячна)	Залежить від використання	Ціна залежить від кількості оброблених зображень; починається від \$1.50 за 1,000 зображень.

Таблиця 1 – Ціни на ліцензійне рішення

Багато високоточних рішень OCR, зокрема Google Cloud Vision та Microsoft Azure Cognitive Services, потребують постійного підключення до Інтернету. Це ускладнює використання технології в автономному режимі, особливо в умовах відсутності або слабого з'єднання, наприклад, у віддалених регіонах. Крім того, для захисту конфіденційності даних користувачі часто обирають локальні рішення, які не надсилають зображення на сервери для обробки, проте такі локальні моделі зазвичай поступаються за точністю.

Хоча існують інструменти, такі як Apple Vision та ML Kit, багато алгоритмів OCR все ще недостатньо оптимізовані для роботи на мобільних платформах. Це призводить до підвищеного споживання енергії, що може швидко розряджати батарею пристрою. Оскільки мобільні пристрої мають обмежені ресурси, включаючи процесор і пам'ять, складні алгоритми машинного навчання часто працюють із затримками або значними обмеженнями в реальному часі. (Рис. 1.6)

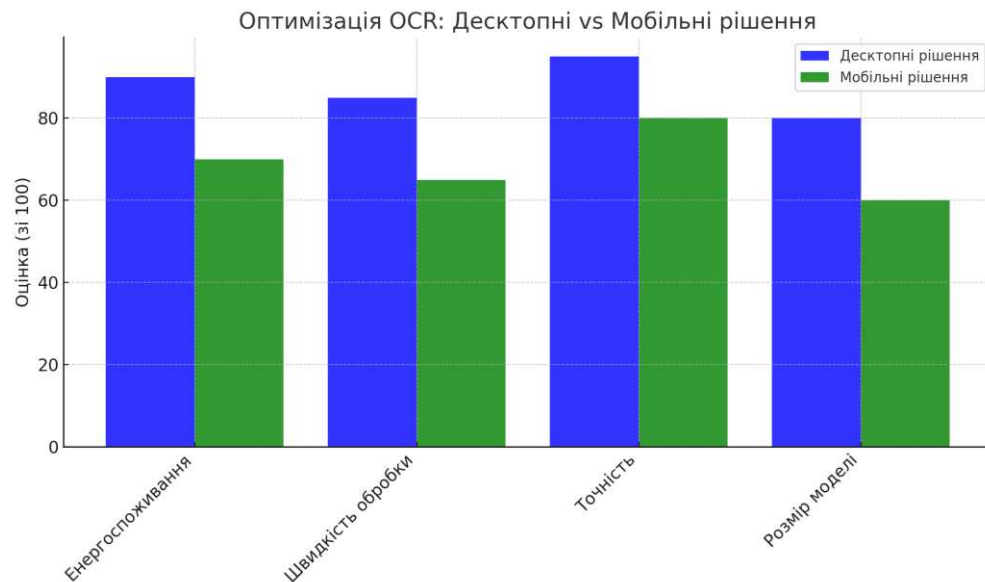


Рисунок 1.6 – графік залежності між десктопним і мобільними рішеннями

OCR-системи часто не можуть точно розпізнати спеціальні символи, такі як математичні формули, хімічні позначення або складні таблиці та графіки. Це значно обмежує їх використання у сферах, які потребують обробки наукових статей, технічних документів чи інших текстів зі складним форматуванням. Для покращення точності в таких випадках необхідно використовувати додаткові алгоритми післяобробки, що ускладнює інтеграцію OCR у мобільні додатки та збільшує витрати на розробку. (Рисунок 1.7)

$$E = mc^2 + \int x^2 dx$$

Рисунок 1.7 – Приклад фото з математичною формулою

1.3 Актуальність, ціль та задачі дослідження

У сучасному світі інформація стає ключовим ресурсом, який впливає на розвиток різних сфер життя — від бізнесу та медицини до освіти та державного управління. Технології оптичного розпізнавання тексту (OCR) активно використовуються для

автоматизації роботи з документами, що робить їх важливим інструментом у багатьох галузях.

Сучасні компанії стикаються з великим обсягом паперових документів, таких як рахунки, контракти та податкові звіти. Технології OCR дозволяють автоматизувати їх обробку, значно скорочуючи час і витрати, які раніше витрачалися на ручне введення даних. Завдяки цьому документи інтегруються у цифрові системи управління, що полегшує їх пошук, аналіз і зберігання. Також OCR знижує ризик людських помилок, що є критично важливим для точного ведення бізнес-операцій.

У медичних установах накопичується велика кількість паперових записів: історії хвороб, результати аналізів, рецепти. OCR допомагає автоматизувати введення цих даних у електронні медичні системи, підвищуючи ефективність роботи медичного персоналу та якість обслуговування пацієнтів. Крім того, технології розпізнавання тексту дозволяють швидко оцифровувати й аналізувати записи, що є важливим для досліджень та прийняття рішень у процесі лікування.

Автоматизація обробки навчальних матеріалів, таких як книги та наукові статті, забезпечує створення електронних бібліотек та баз даних, що спрощує доступ до знань. OCR дозволяє студентам і науковцям ефективніше працювати з текстовими матеріалами, зберігати їх у цифровому вигляді й швидко аналізувати. Завдяки цьому наука та освіта отримують потужний інструмент для розвитку та збереження інформації.

Державні установи працюють із великим обсягом паперових документів, таких як довідки, свідоцтва, звіти. Використання OCR спрощує обробку цих документів, підвищуючи ефективність роботи та зменшуючи витрати часу. У юридичних фірмах OCR використовується для швидкого аналізу контрактів, судових рішень та інших документів, що оптимізує робочі процеси та скорочує час на підготовку справ.

У сфері логістики OCR дозволяє швидко оцифровувати дані з вантажних документів, накладних і рахунків. Це сприяє автоматизації процесів перевезення, відстеження товарів і митного оформлення. Розпізнавання тексту з транспортних етикеток та QR-кодів також спрощує контроль і оптимізацію логістичних ланцюгів, що покращує обслуговування клієнтів і знижує витрати.

Мобільні додатки з функцією OCR дозволяють користувачам сканувати документи на ходу, розпізнавати текст з фотографій квитанцій, візитних карток чи інших матеріалів. Це робить технології OCR зручними для повсякденного використання. Наприклад, мобільні додатки для перекладу тексту можуть автоматично розпізнавати текст із зображень і швидко перекладати його, що особливо корисно для туристів.

В електронній комерції OCR використовується для автоматизації введення даних про товари, обробки відгуків клієнтів і аналізу текстового контенту. Застосування OCR для розпізнавання тексту з етикеток і пакувань товарів значно спрощує управління товарними запасами. Це дозволяє компаніям швидше реагувати на запити клієнтів і покращувати якість обслуговування.

Зростаюча потреба у точності, швидкості та автономності OCR-технологій диктує необхідність їх удосконалення. Це особливо важливо для сфер, де обробка документів має виконуватися в реальному часі. Інтеграція OCR у мобільні додатки, зокрема на платформі iOS, дозволить користувачам отримувати швидкі результати навіть без доступу до Інтернету, що значно розширює можливості автоматизації в різних галузях.

1.4 Висновок

1. Проаналізовано OCR технології в сучасному світі. Які галузі використовують дану технології.
2. Сучасним недоліком залишається оптимізація на мобільному пристрою і визначення тексту в великому об'ємі.

Поставлено наступні задачі:

1. Провести математичний аналіз OCR моделей та їх методів
2. Порівняти методи та обрати модель для оптимізації

РОЗДІЛ 2. Дослідження існуючих засобів та методів оптичного розпізнавання текстів.

2.1 Математичний аналіз існуючих засобів оптичного розпізнавання

Традиційні методи оптичного розпізнавання тексту (OCR) з'явилися задовго до сучасних підходів, що базуються на машинному навчанні. Вони використовуються для розпізнавання тексту в друкованих документах і засновані на аналізі шаблонів, виділенні ознак та статистичних методах. Розглянемо детально ці методи та формули, які використовуються для реалізації OCR.

Розглянемо метод шаблонуого розпізнавання (Template Matching). Шаблонне розпізнавання працює шляхом порівняння вхідного зображення символу з набором збережених шаблонів. Для кожного символу створюється шаблон, і при обробці тексту система шукає найбільш схожий шаблон серед наявних. Цей метод є ефективним для обробки друкованих текстів з фіксованими шрифтами та розміром символів. (Рис. 2.1)



Рисунок 2.1 – Приклад опрацювання фото через Template matching

Шаблонне розпізнавання працює шляхом порівняння вхідного зображення символу з набором збережених шаблонів. Для кожного символу створюється шаблон, і при обробці тексту система шукає найбільш схожий шаблон серед наявних. Цей метод є ефективним для обробки друкованих текстів з фіксованими шрифтами та розміром символів. Модель побудована на метриці подібності, де в основі кореляція

$$C(x, y) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} T(i, j) \cdot I(x + i, y + j)$$

де $C(x, y)$ — результат кореляції, $T(i, j)$ — шаблон, $I(x, y)$ — вхідне зображення.

Якщо значення $C(x, y)$ перевищує певний поріг, символ вважається розпізнаним.

Серед недоліків можна виділити погано працює з текстами, які мають змінні шрифти, нахили або шуми та чутливий до змін розміру та повороту символів.

Наступний метод можна розглянути – Метод аналізу особливостей (Feature Extraction) (Рисунок 2.2). Цей підхід базується на виділенні особливих ознак символів, таких як контури, кути, вершини, горизонтальні та вертикальні лінії. Використовуються методи аналізу контурів та геометричні ознаки для класифікації символів.



Рисунок 2.2 – блок-схема роботи методу Feature Extraction

Для виділення контурів застосовуються градієнтні оператори, наприклад, оператор Собеля:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

де G_x та G_y — градієнти по горизонталі та вертикалі відповідно. Для обчислення сили градієнта:

$$G = \sqrt{G_x^2 + G_y^2}$$

На основі отриманих ознак застосовуються класифікатори, такі як k-ближчі сусіди (k-NN) або методи байєсівської класифікації.

Але для результату потрібно попередньо обробити фото від шумів та на жаль не можливо використати даний метод з рукописним текстом, так як із-зі великої варіації символів – метод унеможлиблює його обробку

Також одним із методів OCR виділяють – метод оптичної фільтрації (Optical Filtering). Застосовується для розпізнавання символів за допомогою частотного аналізу. Використовуються перетворення Фур'є для виділення характерних ознак символів у частотному просторі. Цей метод часто використовується для обробки текстів на складному фоні або в умовах низької якості зображення.

Для математичного обчислення використовується перетворення Фур'є

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) e^{-2\pi i \left(\frac{x}{M} u + \frac{y}{N} v \right)}$$

де $I(x, y)$ — інтенсивність пікселя у вихідному зображенні, $F(u, v)$ — частотне представлення.

Після перетворення Фур'є зображення фільтрується для видалення шумів і виділення основних компонент тексту. Недоліки цього методу є висока обчислювальна складність, що робить метод менш ефективним для використання на мобільних пристроях та дуже чутливий до змін масштабу та обертання символів під різні градуси (Рисунок 2.3)



Рисунок 2.3 – приклад обертання символів

Один з останніх методів можна виділити Байєсівські класифікатори (Bayesian Classifiers). Для класифікації символів використовується теорема Байєса, яка враховує ймовірності різних гіпотез на основі ознак зображення. Байєсівські класифікатори оцінюють, до якого класу символів належить вхідне зображення на основі ймовірнісних моделей (Рисунок 2.4)

Використовується для класифікації символів на основі набору виділених ознак.

$$P(C | X) = \frac{P(X | C)P(C)}{P(X)}$$

де $P(C|X)$ — ймовірність того, що ознака X належить до класу C , $P(X|C)$ — ймовірність ознаки X для класу C , $P(C)$ — апіорна ймовірність класу, $P(X)$ — ймовірність ознаки.



Рисунок 2.4 – приклад роботи метода Bayesian Classifiers

Під час тестування було помічені проблеми з великим обсягом тренувальних даних для побудови надійних ймовірнісних моделей та чутливий до варіацій у шрифтах та стилях написання.

2.2 Порівняний аналіз існуючих методів

У цьому розділі ми порівняємо основні методи оптичного розпізнавання тексту (OCR), враховуючи їх точність, швидкість, вимоги до обчислювальних ресурсів, гнучкість та придатність для використання на різних платформах.

Template Matching є одним із найпростіших методів оптичного розпізнавання тексту. У цьому методі за допомогою шаблонів символів здійснюється пошук відповідностей у зображенні. Цей метод добре працює для текстів зі стандартними шрифтами і фіксованими розмірами символів. Метод є швидким, ефективним для простих завдань. Підходить для обробки стандартних текстів без складних варіацій. Але погано працює з текстами, що мають змінювані шрифти або деформації. Втрачає точність при обробці спотворених зображень. Шаблонне співставлення є ідеальним для простих задач OCR, але не підходить для складних текстів з варіативними шрифтами.

Метод виділення ознак базується на пошуку характерних елементів символів, таких як контури або кути, що дозволяє досягти більш високої точності, ніж шаблонне співставлення. Він особливо корисний для друкованих текстів, де шрифти можуть змінюватися. Підвищує точність розпізнавання при роботі з різними

шрифтами. Можна використовувати для текстів з різними шрифтами і стилями. При роботі віділяють погану якість зображення або шум може знизити ефективність та для досягнення точності необхідна ретельна попередня обробка зображення. Це потужний метод для різних шрифтів, але він вимагає високої якості зображень для досягнення найкращих результатів.

Оптичне фільтрування використовує частотний аналіз для виділення важливих особливостей тексту, що дозволяє зберігати інформацію навіть на складних фонах або при поганій якості зображення. Дає можливість розпізнавати текст на заплутаному фоні. Зазвичай ефективно працює при незначних спотвореннях тексту. Під час тестування зустрівся з такими проблемами як, необхідні великі обчислювальні потужності для обробки зображень. Погіршується ефективність при зміні масштабу чи орієнтації тексту. Підходить для специфічних умов, де текст знаходиться на складному фоні, але вимоги до ресурсів є значними.

CNN є однією з найбільш потужних архітектур для OCR, оскільки вони здатні виділяти складні ознаки тексту і мають високу точність при розпізнаванні друкованих і рукописних текстів. CNN забезпечують відмінні результати при розпізнаванні тексту, навіть у складних умовах. Підходять для роботи з різними типами текстів. Але в роботі зустрічається також і недоліки - потрібно значне обчислювальне навантаження для тренування і використання мереж. Для високої точності необхідно мати великий обсяг даних для навчання. CNN є найбільш сучасним і точним методом OCR, але вимоги до ресурсів і даних обмежують їх використання в деяких випадках.

RNN і LSTM є ефективними для обробки послідовностей символів, оскільки вони враховують контекст і зв'язки між символами, що важливо при розпізнаванні рукописного тексту. Моделі враховують попередні символи, що дозволяє точніше розпізнавати послідовність символів. Особливо ефективні при обробці рукописних текстів, де контекст важливий для точного розпізнавання.

Під час тестування даного методу, було помічені такі недоліки, як моделі потребують багато пам'яті для зберігання та обробки даних. Може бути повільним на мобільних пристроях через високу потребу в обчислювальних потужностях. RNN і LSTM є корисними для задач, де важливо враховувати контекст і послідовність, але

вони можуть бути повільними і потребують значних ресурсів. Результати порівнянь представлені в таблиці 2.1

Метод	Переваги	Недоліки	Точність	Швидкість	Вимоги до ресурсів
Template Matching	Швидкість, ефективність для простих завдань з фіксованими шрифтами.	Низька гнучкість, чутливість до викривлень.	Середня	Висока	Низькі
Feature Extraction	Вища точність, гнучкість для різних шрифтів.	Чутливість до якості зображення, потребує попередньої обробки.	Висока	Середня	Середні
Optical Filtering	Виділення тексту на складному фоні, добре працює з простими текстами.	Високі вимоги до ресурсів, чутливість до масштабу та обертання.	Середня	Низька	Високі
Bayesian Classifiers	Ефективність для великих даних, адаптація до різних шрифтів.	Потребують великих тренувальних даних, менша ефективність для складних задач.	Середня	Середня	Середні
CNN (Convolutional Neural Networks)	Висока точність, гнучкість для різних типів текстів.	Високі вимоги до ресурсів, потребують великого набору	Висока	Низька	Високі

		тренувальних даних.			
RNN/LSTM (Long Short-Term Memory)	Контекстне розпізнавання, корисні для рукописних текстів.	Великі вимоги до пам'яті, повільна обробка на мобільних пристроях.	Висока	Низька	Високі

Таблиця 2.1 – порівняння методів OCR

2. 3 Висновок

Проведений аналіз методів оптичного розпізнавання тексту (OCR) показав, що сучасні технології значно еволюціонували — від традиційних методів, таких як шаблонне розпізнавання та аналіз ознак, до складних моделей на основі нейронних мереж і трансформерів.

Традиційні методи, такі як Template Matching і Feature Extraction, залишаються актуальними для простих завдань, де потрібно швидко і з мінімальними обчислювальними ресурсами розпізнавати текст із високоякісних зображень. Вони добре підходять для обробки друкованих текстів з фіксованими шрифтами, але мають суттєві обмеження при роботі зі змінними шрифтами, рукописними текстами та зображеннями низької якості.

Сучасні методи на основі нейронних мереж (CNN, RNN, Transformers) забезпечують значно вищу точність розпізнавання, особливо в умовах складного фону, низької якості зображень і для рукописних текстів. Вони дозволяють досягати високих результатів завдяки здатності враховувати контекст і адаптуватися до різноманітних сценаріїв. Однак ці методи потребують великих обчислювальних потужностей і пам'яті, що обмежує їх ефективність на мобільних платформах.

Отже, вибір оптимального методу OCR залежить від конкретних умов застосування. Для завдань, де важлива висока швидкість та мінімальні ресурси, доцільно використовувати традиційні методи. Для складніших завдань, таких як обробка рукописних текстів або багатомовних документів, краще підходять моделі на основі нейронних мереж.

У перспективі, подальший розвиток OCR-технологій повинен бути спрямований на оптимізацію алгоритмів для роботи на мобільних пристроях, що дозволить забезпечити високу точність та швидкість розпізнавання навіть в умовах обмежених ресурсів. Це має особливе значення для мобільних додатків, де важлива автономність і здатність працювати без підключення до Інтернету, що відкриває нові можливості для використання OCR у різних сферах — від бізнесу та медицини до освіти та державного управління.

1. Проаналізовано методи OCR та проведено їх математичний аналіз.
2. Зроблено порівняння кожних із методів за критерієм та вибраний метод для оптимізації

Наступні кроки:

1. Використовуючи мову Swift оптимізувати CNN метод на мобільній платформі
2. Розібрати методи оптимізації та порівняти з вихідними даними

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОПТИМІЗАЦІЯ OCR МОДЕЛІ НА ПРИКЛАДІ ДОДАТКУ ДЛЯ ПЕРЕКЛАДУ

3.1 Використані технології для оптимізації OCR Vision на Swift

Після проведення математичного аналізу було обрано алгоритм LSTM так як на основі параметрів аналізу – точність, швидкість, вимоги до ресурсів. Висока точність і робота з різними шрифтами буде нам у нагоді, а полем для оптимізації є швидкість роботи та використання ресурсів. Модель LSTM використовується у Vision framework розроблений компанією Apple з нативними функціями для написання оптимізації на мові Swift.

Для реалізації ефективної оптимізації OCR Vision на Swift були обрані технології та інструменти, які дозволяють забезпечити високу точність розпізнавання тексту, швидку обробку зображень та оптимізацію використання ресурсів на мобільних пристроях. Ці технології включають потужні фреймворки та інструменти, що дозволяють працювати як з простими, так і з складними текстами, зокрема рукописними або деформованими. Для виконання оптимізації був обраний редактор коду - Xcode – основний інструмент для розробки програмного забезпечення для iOS. Xcode є комплексною середовищем для написання коду на Swift, тестування та налагодження додатків, а також інтеграції з іншими технологіями Apple. Вона надає все необхідне для роботи з інтеграцією OCR, включаючи можливість працювати з відлагоджувачем, використовувати вбудовані інструменти для тестування ефективності розпізнавання тексту та інтерфейсами користувача.

Далі для роботи і оптимізації був обраний Vision Framework - це основний фреймворк для обробки зображень на платформі iOS, який використовується для виконання завдань комп'ютерного зору. Він включає API для виконання розпізнавання тексту, виявлення об'єктів та осіб, а також для визначення орієнтації зображень. Для OCR Vision цей фреймворк є основним інструментом, оскільки він має високу ефективність в обробці зображень та підтримує реальне розпізнавання тексту з високою точністю. Завдяки інтеграції з Metal (графічним API Apple) фреймворк оптимізує використання GPU, що дозволяє прискорити обробку зображень. У рамках Vision Framework для iOS реалізовані потужні інструменти для

виконання OCR, які використовують методи обробки зображень, штучного інтелекту і математичні моделі для ефективного розпізнавання тексту.

OCR-процес у Vision починається з виявлення текстових блоків у зображенні. Це може включати як друковані, так і рукописні тексти. Для цього Vision Framework використовує методи детекції контурів та аналітичні методи виявлення текстових елементів. Наприклад, коли користувач намагається захопити зображення документа за допомогою камери, Vision автоматично шукає блоки, що схожі на текстові лінії, щоб потім виконати точніше розпізнавання.

Ключовим етапом OCR є власне розпізнавання тексту після того, як було виявлено текстові блоки. Це досягається за допомогою VNRecognizeTextRequest, який намагається ідентифікувати окремі символи в кожному з виявлених блоків.

Vision також дозволяє обробляти текст, що має незвичайні шрифти або викривлення. Для цього використовуються попередньо натреновані моделі на основі глибинного навчання (deep learning), що дозволяє отримати точні результати навіть при обробці складних шрифтів.

Для розпізнавання тексту Vision Framework може використовувати моделі машинного навчання, які можна інтегрувати через CoreML. Ці моделі навчаються на великих наборах даних і можуть розпізнавати текст у будь-якому шрифті або стилі.

OCR на базі Vision Framework також включає методи для розпізнавання рукописного тексту, де використовуються рекурентні нейронні мережі (RNN) або LSTM (Long Short-Term Memory) моделі для обробки послідовностей символів.

Математичні моделі OCR використовують кілька ключових підходів, щоб перетворити вхідні зображення в текст. Використовувані алгоритми та моделі базуються на численних математичних концепціях, таких як лінійна алгебра, статистика, теорія ймовірностей і чисельні методи.

Перед тим як зображення подається на вход моделі OCR, необхідно провести його попередню обробку. Це включає наступні етапи:

Нормалізація яскравості та контрасту: Для цього використовуються лінійні трансформації (наприклад, зміна інтенсивності пікселів в зображенні).

Бінаризація зображень: Конвертується в чорно-білий формат для полегшення пошуку контурів. Для цього використовується поріг Оцу (Otsu's Thresholding), що дозволяє знаходити оптимальний поріг для бінаризації.

Геометричні трансформації: Коригування перспективи та обертання тексту здійснюється за допомогою перетворень матриць (affine transformations) для коректної орієнтації.

Алгоритми виявлення тексту у Vision Framework працюють за допомогою методів градієнтного спуску для пошуку найкращих параметрів прямої, яка описує контур тексту в зображенні. Часто використовуються методи:

Метод Канни для пошуку контурів: Використовуються фільтри Гаусса для згладжування зображення і для пошуку країв тексту.

Лінійна регресія для визначення напрямку тексту та його границь.

OCR в Vision Framework також базується на моделях машинного навчання, де використовуються нейронні мережі для класифікації символів. Математичні моделі нейронних мереж, зокрема згорткові нейронні мережі (CNN) і рекурентні нейронні мережі (RNN), застосовуються для аналізу зображень і текстів, а саме:

CNN використовуються для виявлення об'єктів у зображеннях та створення фіч, на основі яких визначаються символи.

RNN або LSTM застосовуються для роботи з послідовностями, таких як слова або рядки, що мають контекст, що дозволяє точніше інтерпретувати розпізнаний текст.

В OCR для уточнення результатів і зменшення помилок використовуються статистичні методи та теорія ймовірностей. Один із популярних підходів — це Байєсівський класифікатор, який дозволяє оцінювати ймовірність того, що символ належить певному класу, на основі навчальних даних. Також використовуються методи максимального правдоподібності для оцінки параметрів моделі.

Після того як текст був розпізнаний, його ще потрібно очистити від помилок, таких як неправильне розпізнавання символів або відсутність пробілів між словами. Для цього застосовуються статистичні методи на основі моделі мови. Наприклад, n-gram моделі або мовні моделі допомагають відновити пропущені слова і коректно обробляти розпізнаний текст. В таблиці 3.1 зображено відношення процесу до часу виконання процесу OCR.

Етап процесу OCR	Час виконання (секунди)
Підготовка зображення	0.1-0.5
Виявлення текстових блоків	0.2-0.8
Розпізнавання тексту (нейронні мережі)	0.5-1.5
Постобробка результатів	0.1-0.3

Таблиця 3.1 – відношення роботи OCR у Vision Framework до часу

Також розглянемо допоміжні фреймворки CoreML – технологія машинного навчання, яка дозволяє інтегрувати попередньо натреновані моделі в iOS-додатки. Для OCR CoreML забезпечує підтримку нейронних мереж, які спеціалізуються на розпізнаванні тексту, зокрема для рукописних та складних шрифтів. Моделі, що використовуються для OCR, можуть бути налаштовані для різних мов і шрифтів, що робить їх універсальними для різних завдань. CoreML також дозволяє виконувати ці обчислення на мобільних пристроях без необхідності в постійному підключенні до серверів, що покращує швидкість і зменшує споживання енергії.

Для графічного відображення будемо використовувати фреймворк SwiftUI – фреймворк для побудови інтерфейсів користувача, що дозволяє швидко створювати адаптивні інтерфейси для різних пристроїв iOS, включаючи iPhone та iPad. Завдяки SwiftUI можна створювати інтуїтивно зрозумілі елементи управління для OCR-додатків, такі як кнопки, текстові поля та індикатори прогресу, які реагують на дії користувача, наприклад, при завантаженні та обробці зображень. SwiftUI дозволяє легко обробляти асинхронні запити на обробку зображень і інтегрувати їх з реальним часом.

Для реалізації та обробки асинхронних подій Combine Framework – для обробки асинхронних подій та реалізації реактивного програмування. Використання Combine дозволяє реалізувати елегантні рішення для управління потоком даних, наприклад, при обробці зображень OCR. Завдяки цьому фреймворку можна автоматично оновлювати інтерфейс користувача під час кожного етапу обробки зображення, забезпечуючи більш динамічний і зручний досвід для кінцевого користувача.

Також головним фактором оптимізації є робота з пристроєм вводу – в мобільних пристроях цим пристроєм є камера і для роботи з нею використовується AVFoundation — це потужний фреймворк від Apple для роботи з мультимедійними файлами, такими як відео, аудіо, а також з камерами та мікрофонами. Для застосунків, що включають OCR, важливо мати можливість захоплення зображень або відео з камери для подальшої обробки і розпізнавання тексту. AVFoundation надає повний контроль над камерою і дозволяє працювати з відео потоками в реальному часі.

3.2 Побудова діаграм варіантів використання

Діаграма варіантів використання (Use Case Diagrams) розроблюється з метою уточнення функцій програми, визначення схеми функціональної структури, визначення модулів програми і вивчення динаміки її роботи. На діаграмі зображуються системні прецеденти (use cases), системне оточення (діючих осіб або акторів - actors) та зв'язки між ними.

Розглянемо більш детально варіант використання, надаючи опис потоків подій.

Специфікація варіанту використання “Аналіз фото”

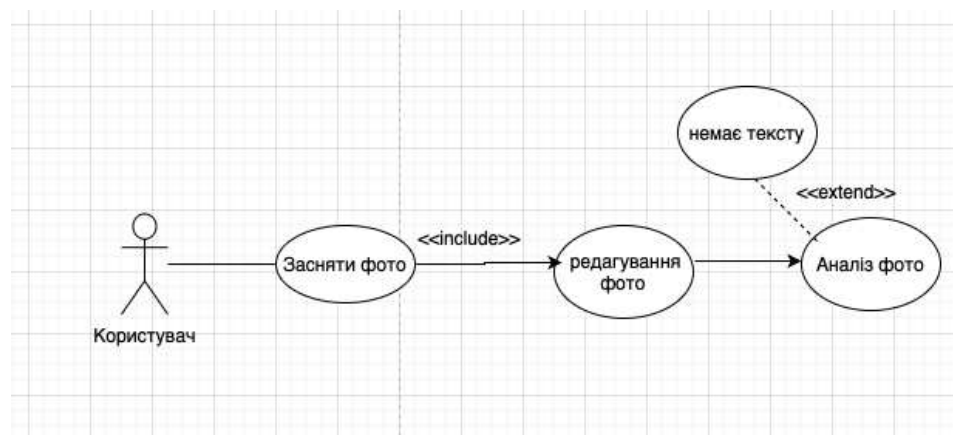


Рисунок 3.1 – Діаграма прецедентів

1. Призначення: призначений для отримання текст з фотографії зробленої на камеру пристрою.
2. Учасники: активний суб'єкт «Користувач».
3. Передумови: зробити фото з текстом
4. Стартова точка: ініціюється активним суб'єктом «Користувач».

5. Процедура

Користувач наводить камеру на об'єкт з текстом. Згодом програма робить фото і користувач може редагувати фото. Якщо тексту на фото немає – користувач отримує помилку. Якщо текст є – програма повертає текст з зображення.

6. Аварійні умови (при яких умовах настає аварія): відсутні

7. Результат: Після завершення програма показує текст з фото.

8. Умови після закінчення використання: немає.

9. Особливості: Якщо тексту на фото немає – то програма повідомляє користувачу.

10. Відповідні вимоги: відсутні.

11. Опис: Варіант використання ініціюється користувачем програми і надає можливість отримати аналіз фото.

Діаграма варіантів використання програми дозволяє визначити діаграму послідовності, яка дає розуміння поведінку системи, тобто особливостей взаємодії об'єктів.

Діаграма послідовності в UML відображає взаємодію між об'єктами або компонентами системи протягом часу, тобто показує, як об'єкти обмінюються повідомленнями для виконання певної функціональності. Діаграма представлена на рисунку 3.2

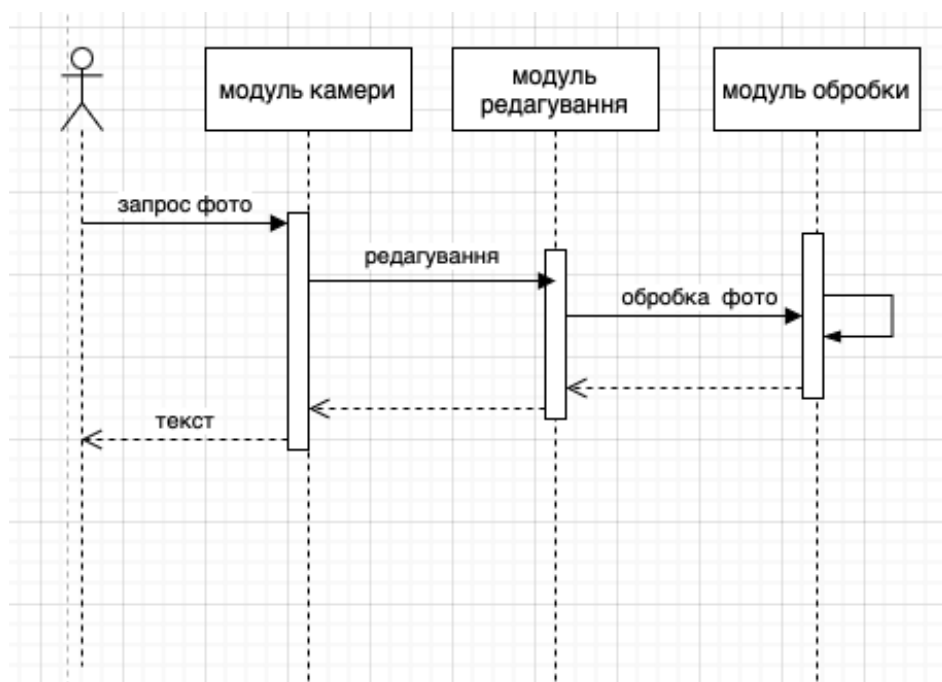


Рисунок 3.2 – Діаграма послідовності.

3.3 Проведення дій з прискорення завантаження камери для OCR на Swift

Прискорення завантаження камери є важливим аспектом для забезпечення швидкості і зручності в OCR-додатках на мобільних пристроях. Від швидкості відкриття камери залежить не тільки досвід користувача, але й загальна продуктивність додатку, особливо якщо це стосується розпізнавання тексту в реальному часі. Для досягнення оптимальних результатів необхідно врахувати кілька факторів, зокрема ефективне використання ресурсів та мінімізацію затримок при підключенні до камери.

Почнемо роботу з фреймворком AVFoundation – на початку потрібно вірно налаштувати сесію роботи з камерою. Головним нюансом роботи є запуск і налаштування камери у асинхронному середовищі для того, щоб користувач не мав проблем з доступом до додатку під час налаштувань пристрою запису. Це можливість завдяки якій на головній черзі додатку будуть оброблюватись інші дії користувача такі як натискання, перехід на інші екрани і можливість взаємодії з іншими модулями додатку.(рис 3.3 – рис 3.4)

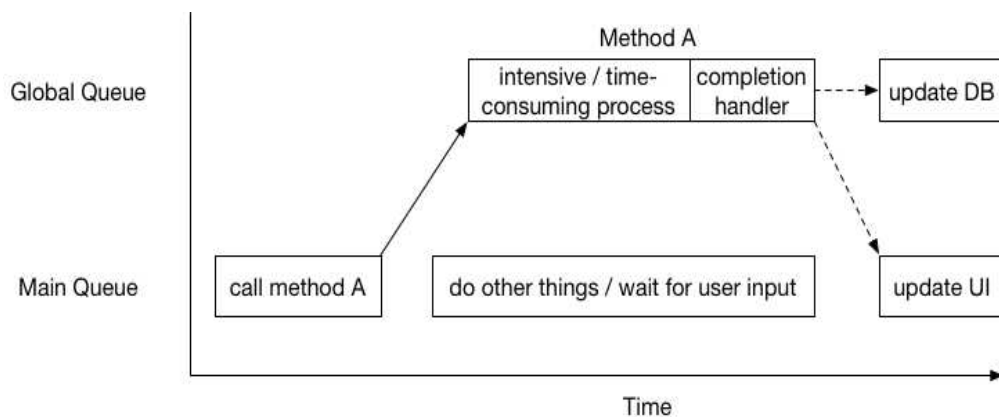


Рисунок 3.3 – Зображення роботи головної та глобальної черги

```
func startApp() {
    DispatchQueue.global(qos: .userInitiated).async {
        self.setupCameraSession()

        DispatchQueue.main.async {
            // Оновлюємо UI або продовжуємо роботу з камерою
        }
    }
}

func setupCameraSession() {
```

Рисунок 3.4 – Запуск налаштування камери

Хоча це і надає вже на початку нам велику перевагу в оптимізації, надалі нам також потрібно використати можливості налаштування AVFoundation. Так як сучасні пристрою можуть не тільки робити знімки, а ї записувати відео нам потрібно додати уточнення, що ми будемо працювати тільки з режимом відео але без навантаження на запис звуку – так як наша головна ціль отримати зображення.(Рис.3.3)

```
func setupCameraSession() {
    let videoDevice = AVCaptureDevice.default(for: .video)
    let videoInput = try AVCaptureDeviceInput(device: videoDevice!)
}
```

Рисунок 3.5 – встановлення камери без звуку

Також потрібно звернути увагу на оптимізацію відео запису та як вона відбувається. Коли додаток не вимагає високої роздільної здатності або частоти кадрів для задач, таких як OCR, можна використовувати більш низькі пресети для відео, що дозволить знизити навантаження на процесор і оперативну пам'ять, що значно прискорить завантаження камери та поліпшить продуктивність додатка.

AVCaptureSessionPreset — це параметр сесії, який визначає роздільну здатність відео. AVCaptureSessionPresetLow: Це низька роздільна здатність, яка ідеально підходить для швидкого захоплення відео з мінімальним навантаженням на систему. Використовується, якщо OCR не вимагає високої деталізації тексту.(рис 3.6)

```
do {
    let videoInput = try AVCaptureDeviceInput(device: videoDevice)
    if captureSession.canAddInput(videoInput) {
        captureSession.addInput(videoInput)
    }

    let videoOutput = AVCaptureVideoDataOutput()
    if captureSession.canAddOutput(videoOutput) {
        captureSession.addOutput(videoOutput)
    }
    captureSession.sessionPreset = .low
    captureSession.startRunning()
}
```

Рисунок 3.6 – Встановлення низької роздільної здатності для сесії

Іншим важливим аспектом є частота кадрів. Висока частота кадрів, наприклад, 60 кадрів на секунду, є ідеальною для відео, але може бути зайвою для задач OCR, де кожен кадр потрібно лише захопити та обробити без необхідності в реальному часі демонструвати його з високою плавністю а ось зменшення до 30 кадрів на секунду дозволяє значно знизити навантаження на систему. Це ідеальний вибір для OCR, де не потрібен високий рівень деталізації в реальному часі. ншення роздільної

здатності та частоти кадрів має кілька важливих переваг для продуктивності додатка. Зниження використання процесора і пам'яті: Менша роздільна здатність та знижена частота кадрів означають менше даних для обробки, що знижує навантаження на центральний процесор (CPU) та графічний процесор (GPU). Швидший старт камери: Пресети з низькою роздільною здатністю зменшують час, необхідний для ініціалізації камери, що важливо для забезпечення швидкого доступу до відео потоку, особливо в додатках OCR. Покращення енергетичної ефективності: Менше даних та знижене навантаження на процесор також допомагають знизити споживання енергії, що є критичним для мобільних додатків, особливо коли камери використовуються протягом тривалого часу.

Вибір правильних пресетів є критичним для досягнення балансу між якістю зображення та продуктивністю, що дозволяє досягти максимальної ефективності для OCR-додатків на мобільних пристроях.

Далі продовжимо тему з оптимізацією та обробкою зображень на стороні GPU Metal — це низькорівневий API для роботи з графічними і обчислювальними операціями на пристроях Apple, який дозволяє виконувати високопродуктивні операції безпосередньо на графічному процесорі (GPU). У випадку додатків, що працюють з OCR (оптичне розпізнавання тексту), використання Metal для обробки зображень має безпосередній вплив на продуктивність та ефективність. Це особливо важливо для обробки відео потоку в реальному часі, коли кожен кадр повинен бути оброблений швидко і з високою точністю.

Обробка зображень за допомогою GPU значно прискорює виконання обчислювальних операцій порівняно з CPU. Це стає критичним при роботі з великими обсягами даних, такими як відео потоки або великі зображення. У стандартних умовах, коли обробка зображень здійснюється за допомогою CPU, кожна операція, пов'язана з фільтрацією, згорткою або іншими графічними операціями, може значно сповільнювати процес, що веде до зниження швидкості розпізнавання тексту в реальному часі.

Перенесення цих обчислень на GPU зменшує час обробки, оскільки графічний процесор спеціалізується на паралельних обчисленнях, що дозволяє одночасно виконувати безліч обчислень для кожного пікселя в зображенні. Це особливо важливо для OCR, оскільки кожен кадр відео або кожне зображення потребує

численних математичних операцій (наприклад, перетворення зображення, фільтрація шумів, виявлення контурів тексту тощо).

OCR включає багато математичних операцій, таких як перетворення зображень, фільтрація контурів та згортка. Ці операції, особливо в реальному часі, можуть значно уповільнювати процес, якщо вони виконуються лише на CPU. Однак за допомогою Metal, ці операції можуть бути передані на GPU, що дозволяє значно прискорити процес.

Згортка — одна з основних операцій, яка використовується при фільтрації та обробці зображень для OCR. Виконання згортки на GPU значно швидше, оскільки GPU здатен виконувати паралельні обчислення для кожного пікселя в зображенні.

Виявлення контурів — ще одна важлива операція при підготовці зображень для OCR. Методи, що використовують алгоритми, такі як Собель або Канн, можуть бути значно прискорені за допомогою GPU, дозволяючи зменшити час затримки на обробку кожного кадру.

Заміна обробки на GPU замість CPU також має переваги з точки зору енергоефективності. GPU оптимізований для виконання великих обчислень на численних пікселях одночасно, що дозволяє знизити енергоспоживання при виконанні графічних операцій порівняно з CPU, який виконує обчислення по черзі. Це є важливим для мобільних додатків, де батарея є обмеженим ресурсом.

Наприклад, при обробці відео або потокових даних в реальному часі, Metal дозволяє значно зменшити енергоспоживання, оскільки обчислення виконуються на GPU, що є менш енергозатратним для таких задач, ніж обробка на CPU.

OCR, яке працює в реальному часі, вимагає низьких затримок для забезпечення швидкої обробки зображень. Metal дозволяє отримати значно нижчі затримки при обробці відео потоку або зображень в реальному часі, оскільки GPU здатний до паралельного виконання численних операцій. Це є критично важливим для таких додатків, як OCR, де кожен кадр повинен бути оброблений миттєво.

У порівнянні з CPU, GPU здатний швидше обробляти дані через високий рівень паралелізму, що дозволяє виконувати всі необхідні обчислення для OCR (наприклад, фільтрацію зображень або розпізнавання тексту) без значних затримок. (рис. 3.7)

```

0
1 // Обробка відео потоку
2 func processSampleBuffer(sampleBuffer: CMSampleBuffer) {
3     guard let pixelBuffer = CMSampleBufferGetImageBuffer(sampleBuffer) else { return }
4
5     // Перетворення CVPixelBuffer в MTLTexture
6     let texture = createTextureFromPixelBuffer(pixelBuffer)
7
8     // Передача Metal Texture до Vision для OCR
9     processMetalTextureWithVision(texture: texture!)
10 }
11
12 // Перетворення CVPixelBuffer в MTLTexture
13 func createTextureFromPixelBuffer(_ pixelBuffer: CVPixelBuffer) -> MTLTexture? {
14     // Створення текстури з CVPixelBuffer
15     let textureDescriptor = MTLTextureDescriptor()
16     textureDescriptor.pixelFormat = .bgra8Unorm
17     textureDescriptor.width = CVPixelBufferGetWidth(pixelBuffer)
18     textureDescriptor.height = CVPixelBufferGetHeight(pixelBuffer)
19     textureDescriptor.usage = [.shaderRead, .shaderWrite]
20
21     guard let texture = metalDevice.makeTexture(descriptor: textureDescriptor) else {
22         print("Не вдалося створити MTLTexture")
23         return nil
24     }
25
26     // Копіювання пікселів з CVPixelBuffer до MTLTexture
27     let region = MTLRegionMake2D(0, 0, CVPixelBufferGetWidth(pixelBuffer), CVPixelBufferGetHeight(pixelBuffer))
28     CVPixelBufferLockBaseAddress(pixelBuffer, .readOnly)
29     let baseAddress = CVPixelBufferGetBaseAddress(pixelBuffer)
30
31     texture.replace(region: region, mipmapLevel: 0, withBytes: baseAddress!, bytesPerRow: CVPixelBufferGetBytesPerRow(pixelBuffer))
32     CVPixelBufferUnlockBaseAddress(pixelBuffer, .readOnly)
33
34     return texture
35 }
36

```

Рисунок 3.7- обробка фото за допомогою Metal

Один з основних аспектів використання Metal для обробки зображень — це можливість застосовувати шейдери для маніпулювання пікселями зображення. Шейдери дозволяють ефективно виконувати операції над текстурами (наприклад, для покращення контрасту, зміни яскравості або фільтрації шумів). Оскільки ці операції можуть бути виконані паралельно на кожному пікселі, це дозволяє досягти високої швидкості обробки, що важливо для OCR у реальному часі.

Щоб зменшити час затримки при доступі до камери, можна зберігати деякі параметри сесії, такі як налаштування відео-вхідного потоку та попередні налаштування, у кеші або в пам'яті. Це дозволить уникнути повторного налаштування камери при кожному запуску додатку.(рис 3.8)

```

func loadCameraSettings() {
    if let cameraType = userDefaults.string(forKey: selectedCameraKey) {
        print("Завантажено налаштування для камери: \(cameraType)")
    }
}

```

Рисунок 3.8 — Завантаження параметри сесії в пам'яті

3.4 Проведення дій з прискорення роботи Vision в мобільному додатку

Тепер потрібно оптимізувати роботу OCR алгоритму через оптимізацію з роботою Vision. Розпізнавання тексту працює через об'єкт класу VNRecognizeTextRequest. Це клас для ініціалізації запиту на розпізнавання тексту в зображенні або відео. Він використовує попередньо натреновану модель для

виявлення тексту у зображеннях. Це основний метод і вже на цьому етапі ми можемо обрати тип `.fast` для більш швидкого розпізнавання. Цей рівень розпізнавання тексту оптимізований для швидкості. Використовується більш легкий підхід до аналізу зображення, що дозволяє значно зменшити час обробки, але може призводити до менше точної обробки тексту, особливо в складних випадках (наприклад, при деформаціях або нестандартних шрифтах).

Коли ви використовуєте `.fast`, система використовує менш складні моделі для розпізнавання тексту (рис 3.9). Це означає, що Vision не буде проводити глибокий аналіз шрифтів або контексту, а обмежиться базовим розпізнаванням, що дозволяє заощаджувати час. використовує алгоритми, які дають результат за менший час, часто в обмін на точність. Наприклад, швидкий аналіз може обмежуватися виявленням великих блоків тексту без детального вивчення дрібних деталей. що для кожного кадру або зображення потрібно виконати менше обчислень. Алгоритми для швидкого розпізнавання тексту не використовують додаткові ресурси, такі як глибокі нейронні мережі для аналізу контексту і структури тексту.

```
visionRequest = VNRecognizeTextRequest(completionHandler: { (request, error) in
    guard let results = request.results else { return }
    for observation in results {
        if let textObservation = observation as? VNTextObservation {
            // Обробка виявленого тексту
            //print(textObservation.topCandidates(1).first?.string ?? "No text found")
        }
    }
})
visionRequest.recognitionLevel = .fast
```

Рисунок 3.9— Встановлення рівня розпізнавання

Це зменшує навантаження на CPU і GPU, що дозволяє досягти швидшої обробки зображень або відео, а також знижує споживання енергії, що є важливим на мобільних пристроях.

Наступним методом оптимізації є зменшення роздільної здатності зображень є одним із важливих методів оптимізації в процесі OCR, особливо коли ви працюєте з відеопотоком або великою кількістю зображень, що надходять від камери в реальному часі. В умовах обмежених ресурсів, таких як мобільні пристрої, зменшення роздільної здатності дозволяє значно зменшити навантаження на процесор і графічний процесор, одночасно зберігаючи достатній рівень точності розпізнавання тексту. Це особливо важливо в контексті OCR, коли точність і швидкість виконання обробки мають велике значення. зменшення роздільної здатності

зображення — це процес, при якому ви зменшуєте кількість пікселів, які необхідно обробляти для виконання OCR. Цей підхід дозволяє зменшити навантаження на обчислювальні ресурси: Коли роздільна здатність зображення зменшується, відповідно зменшується кількість пікселів, що потрібно обробити. Це означає, що процесору (CPU) і графічному процесору (GPU) потрібно обробляти менше даних, що значно знижує загальне навантаження на систему.(рис 3.8)

Надає перевагу в швидкій обробці зображен через зменшення кількість пікселів означає, що алгоритми, які використовуються для розпізнавання тексту, будуть працювати швидше. Це критично важливо для додатків, де потрібно виконувати OCR в реальному часі, наприклад, для сканування документів або розпізнавання тексту з відео потоку.

Також зменшення роздільної здатності зображення також зменшує кількість пам'яті, необхідної для збереження зображення. Це особливо важливо для мобільних пристроїв, де пам'ять є обмежен

Масштабуючи зображення до більш низької роздільної здатності (наприклад, до 640x480 або 320x240 пікселів), ви зменшуєте кількість пікселів, які потрібно обробити для розпізнавання тексту. Це дозволяє алгоритмам OCR працювати швидше, оскільки кожен кадр або зображення має менший обсяг даних для обробк

Алгоритми OCR зазвичай працюють швидше, коли зображення має менше пікселів. Для задач, де текст добре видно навіть на зображеннях низької роздільної здатності, така оптимізація дозволяє досягти значного прискорення обробки, не втрачаючи значної точності.

Зменшена кількість пікселів означає, що алгоритми, які використовуються для розпізнавання тексту, будуть працювати швидше. Це критично важливо для додатків, де потрібно виконувати OCR в реальному часі, наприклад, для сканування документів або розпізнавання тексту з відео потоку.

```
// Масштабування UIImage до нової роздільної здатності
func scaleImageForOCR(ciImage: CIImage, targetWidth: CGFloat, targetHeight: CGFloat) -> CIImage {
    let scaleTransform = CGAffineTransform(scaleX: targetWidth / ciImage.extent.size.width,
                                             y: targetHeight / ciImage.extent.size.height)
    let scaledImage = ciImage.transformed(by: scaleTransform)
    return scaledImage
}
```

Рисунок 3.10 — Функція для зменшення роздільної здатності зображення

Вибіркове оброблення кадрів — це стратегія, яка дозволяє значно знизити навантаження на систему та прискорити процес розпізнавання тексту в реальному часі, що особливо важливо для застосунків, що працюють з відео потоком або великими обсягами зображень. Цей підхід дозволяє економити ресурси пристрою та підвищити ефективність без суттєвих втрат в точності для більшості реальних випадків OCR.

Кожен кадр відео або зображення, що надходить з камери, потребує обробки — аналізу пікселів, визначення контурів тексту, класифікації символів тощо. Це може бути дуже ресурсоємно, особливо для OCR, яке потребує значних обчислювальних потужностей. Якщо обробляти кожен кадр, система буде постійно виконувати ці операції, що сильно навантажує CPU та GPU.

Вибіркове оброблення кадрів дозволяє обробляти лише кожен N-й кадр, що зменшує кількість обчислень на кожному етапі. Наприклад, якщо ви обробляєте кожен 3-й кадр, це означає, що ви зменшуєте кількість кадрів для обробки на 66%. Замість того, щоб виконувати складні операції на кожному кадрі, система виконує обробку лише на вибраних кадрах, тим самим значно зменшуючи загальне навантаження.

Зменшення кількості кадрів, які потрібно обробляти, дозволяє значно зменшити загальний час обробки. Наприклад, в умовах реального часу, де важливо отримувати результат OCR швидко, обробка кожного кадру може зайняти більше часу, ніж це необхідно. Замість того, щоб чекати на кожен результат для кожного кадру, вибіркове оброблення дозволяє значно прискорити загальний процес:

- Якщо обробка кожного кадру займає 100 мс, обробка кожного 3-го кадру займе лише 33 мс на кадр, що значно покращує ефективність.
- Це дозволяє знижувати час затримки між отриманням зображення і розпізнаванням тексту, що важливо для взаємодії з користувачем в реальному часі.

Оскільки мобільні пристрої мають обмежені ресурси батареї, зменшення обсягу обробки дозволяє значно знизити енергоспоживання. Вибіркове оброблення кадрів дозволяє пристрою працювати менш інтенсивно, що зменшує навантаження на CPU, GPU та інші компоненти, що в свою чергу знижує споживання енергії.

- Менше оброблених кадрів означає менше використаної енергії, що особливо важливо для додатків, які працюють довгий час або в мобільному середовищі.

- Це дозволяє зберегти заряд батареї, що робить додаток більш ефективним з точки зору енергоспоживання, особливо при роботі з камерами або відеопотоками.

Вибіркове оброблення кадрів не тільки зменшує навантаження на ресурси, але й дозволяє зменшити затримки при обробці відео потоку. Оскільки не потрібно обробляти кожен кадр, система може швидше реагувати на нові кадри, забезпечуючи менше затримок і кращу плавність обробки.

При обробці кожного кадру час затримки між кадрами може накопичуватися, особливо в додатках, де потрібно обробити відеопотік з високою частотою кадрів. Вибіркове оброблення дозволяє значно зменшити ці затримки.

Це особливо важливо для мобільних пристроїв або веб-додатків, де частота кадрів і затримка є критичними параметрами для зручності користувача.

Для OCR важливо мати стабільне та чітке зображення для досягнення високої точності розпізнавання. Вибіркове оброблення кадрів може бути корисним в ситуаціях, коли зміни на зображенні між кадрами незначні. Наприклад, для статичних або повільно змінюваних сцен (коли текст не змінюється швидко), розпізнавання тексту на кожному кадрі не є необхідним. Обробка кожного N-го кадру може дати вам точний результат, оскільки текст на кадрі залишається стабільним, і система має достатньо часу для точного розпізнавання. Алгоритм роботи показаний на рисунку 3.11

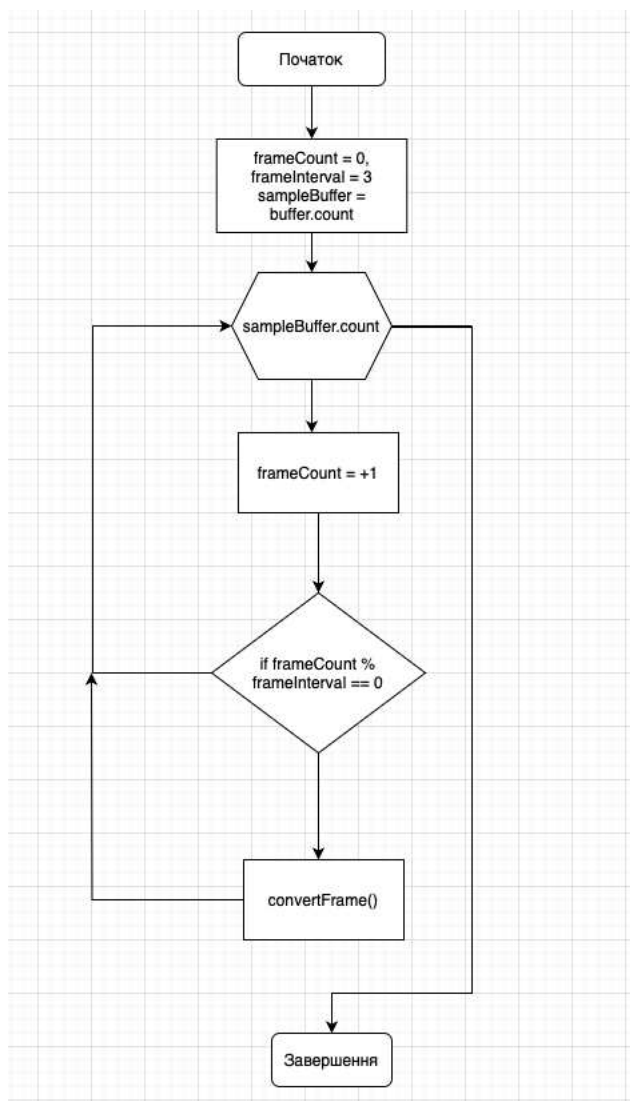


Рисунок 3.11 – Блок-схема обчислення кадрів

Наступні наші дії по оптимізації будуть крок схожий при налаштуванні камери, щоб уникнути зупинки основного додатку. OCR для кожного кадру може зайняти деякий час, особливо при високій роздільній здатності. Виконання цих операцій асинхронно дозволяє уникнути блокування основного потоку, що особливо важливо для UI-додатків, де необхідно зберігати плавність взаємодії з користувачем. У випадку, коли вам потрібно обробляти відео потік або серію кадрів в реальному часі, асинхронність дає змогу забезпечити мінімальну затримку між отриманням кадру і його обробкою (OCR). Якщо ці операції будуть виконуватись синхронно, то система може не встигати обробляти кадри вчасно, що призведе до затримок, а, можливо, і до пропусків кадрів, що буде впливати на точність і швидкість розпізнавання. Оскільки OCR є ресурсоємною операцією, виконання її асинхронно дає змогу використовувати більше ніж один потік для обробки даних, дозволяючи

знизити загальне навантаження на основний потік. Це дозволяє краще управляти використанням ресурсів і зберігати належну продуктивність додатку.(рис. 3.12)

```
DispatchQueue.global(qos: .userInitiated).async {
    let textRequest = VNRecognizeTextRequest(completionHandler: { (request, error) in
        guard let results = request.results else { return }
        DispatchQueue.main.async {
            for observation in results {
                if let textObservation = observation as? VNTextObservation {
                    print(textObservation.topCandidates(1).first?.string ?? "No text found")
                }
            }
        }
    })
}
```

Рисунок 3.12 – Виконання OCR асинхронно в фоновому потоці

Однією з важливих стратегій оптимізації в реальному часі, особливо при обробці відео потоку для OCR, є обробка лише потрібних областей зображення (ROI - Region of Interest). Вона дозволяє значно зменшити час обробки, знизити навантаження на систему і підвищити ефективність розпізнавання тексту. Цей підхід полягає в тому, щоб обробляти тільки ті частини зображення, які містять корисну інформацію (наприклад, текст), і ігнорувати інші частини, що не містять тексту або не мають значення для розпізнавання.

OCR алгоритми працюють шляхом аналізу пікселів зображення для пошуку тексту. Якщо обробляти все зображення, процес може бути повільним, особливо при високій роздільній здатності або великих зображеннях. Обмеження обробки тільки на області, де ймовірно є текст, значно скорочує кількість пікселів, що підлягають обробці, і прискорює процес.

Обробка всього зображення вимагає великих обчислювальних ресурсів, особливо якщо зображення містить багато деталей, які не є корисними для OCR (наприклад, фони, предмети, що не містять тексту). Зменшивши площу для аналізу, система витрачає менше ресурсів (CPU, GPU), що покращує загальну продуктивність.(рис 3.13)

Сфокусувавшись на малих ділянках зображення, де точно очікується текст (наприклад, на заголовках, абзацах або у визначених полях документа), можна прискорити процес розпізнавання, оскільки OCR не потрібно “перевіряти” кожен піксель на зображенні.

ROI визначає частину зображення, на якій ймовірно знаходиться текст. Це може бути, наприклад, область навколо текстового блоку, який потрібно розпізнати (наприклад, текст на документах, таблицях або знаках).

Замість того, щоб обробляти все зображення, ви працюєте лише з тими ділянками, де знаходиться текст. Це значно зменшує час обробки та знижує навантаження на систему. OCR для великого зображення вимагає значних ресурсів. Обробка тільки ROI дозволяє зменшити кількість обчислень і витрати пам'яті, оскільки ви працюєте тільки з малими частинами зображення.

У реальному часі, де важлива швидкість обробки, обробка тільки ROI дозволяє значно зменшити затримки, що важливо для додатків, таких як сканери документів або додатки для розпізнавання тексту в відео. Оскільки OCR обробляє лише текстові області, ймовірність помилок зменшується, оскільки фонові елементи або інші частини зображення не будуть заважати в процесі розпізнавання.

```
func detectTextRegions(in ciImage: CIImage) {
    let textDetectionRequest = VNDetectTextRectanglesRequest { request, error in
        guard let observations = request.results as? [VNTextObservation] else { return }

        for observation in observations {
            let boundingBox = observation.boundingBox

            let croppedImage = ciImage.cropped(to: boundingBox)

            self.processOCR(on: croppedImage)
        }
    }

    let requestHandler = VNImageRequestHandler(ciImage: ciImage, options: [:])
    try? requestHandler.perform([textDetectionRequest])
}
```

Рисунок 3.13 – Обробка тексту окремими блоками

Також в данній технічній реалізації всі моделі для OCR знаходяться на мобільному пристрою і виконуються локально. Але також потрібно розглянути варіант, коли у користувача є доступ до інтернету. Тоді можна використати варіант в якому частина моделей завантажується через інтернет. Дана реалізація представлена на діаграмі розгортання(рис. 3.14)

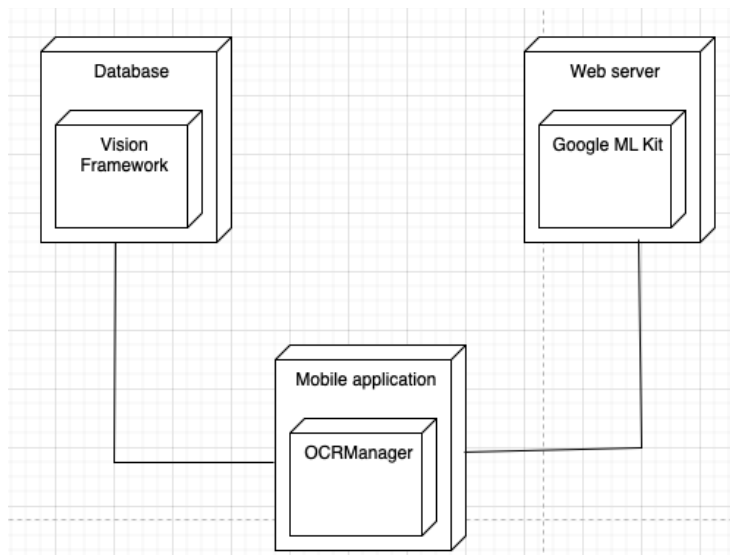


Рисунок 3.14 – Діаграма розгортання

3.5 Результат роботи над оптимізацією та заміри

В результаті розробки було реалізовано мобільний додаток, яке забезпечує перетворення візуального формату даних на текстовий тип

Програмний продукт розрахований на роботу у трьох вікнах де перший екран показує результат нашого сканування, другий екран бере на себе роль камери для зчитування фото з камери і останній екран допомагає обробляти фото і виділяти потрібну частину тексту. Програмний продукт має зрозумілий інтерфейс для користувачів.

Функції OCR викликаються за допомогою засобів інтерфейсів –кнопками. На рисунках 3.15 – 3.22 зображена робота додатку при різних типів вхідного фото та тексту на якому вона зображена.



Рисунок 3.15 – Сканування тексту з звичайним шрифтом

Завдяки проробленої оптимізації структури OCR ми можемо побачити, що для нашого алгоритму це не є проблемою.

TermoPasty
FLUX
RF800
NO CLEAN FLUX
ISO 29454-1 typ 1.1.3 A / DIN 8511 typ F-SW 32
Topnik RF800 do lutowania SMD
Flux RF800 for soldering SMD
Flussmittel RF800 für SMD-Löten
Флюс RF800 для SMD пайки
Flux RF800 yia cuy kóMAAndn SMD

[Scan document](#)

Рисунок 3.16 – Результат сканування звичайного шрифту

Як бачимо ніяких проблем звичайний шрифт для OCR не доставив. Розглянемо більш складний варіант з рукописним текстом в зошиті.



Рисунок 3.17 – фото рукописного тексту в зошиті.

Для версії без оптимізації – це могло виникнути проблеми, але наш алгоритм OCR впорався.



Рисунок 3.18 – Результат обробки рукописного тексту.

Давайте розглянемо нашу оптимізацію з відділенням головного блоку тексту, який саме нам потрібен.

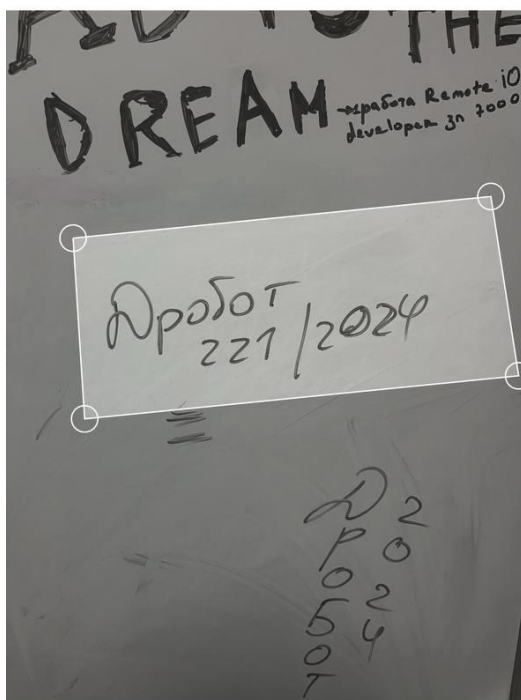
[< Back](#)[Retake](#)[Keep Scan](#)

Рисунок 3.19 – Обработка потрібного блоку тексту

03:37

19

Дробот
221/2029[Scan document](#)

Рисунок 3.20 – Результат роботи з виділеним текстом

Тепер головна задача -це визначення тексту на змішаному фоні де є картинка та використовується складний фон.



Рисунок 3.21 – Фото змішаного тексту

Завдяки алгоритму через Metal Foundation нам вдалось обробити такий тип.

03:38



B.255.
 "onl
 TE SANDERN
 10681
 Bille
 PORTKEY GAMES
 NOGWARTS
 LEGACY
 12
 www.pegi.info
 AVALANCHE
 UB
 WARNER BROS.
 GAMES

Scan document

Рисунок 3.22 – Результат змішаного фото.

3.6 Висновок

Розроблено новий алгоритм оптимізації обробки відео та зображень для оптичного розпізнавання тексту (OCR), що використовує сучасні технології Vision та Metal. Проведена практична реалізація обробки відеопотоку з камери для виконання OCR, включаючи визначення текстових блоків на зображеннях і розпізнавання тексту в реальному часі.

Розроблено архітектуру інформаційної системи, що має модульну структуру, яка складається з наступних основних модулів:

- Захоплення відео та зображень з камери.
- Обробка зображень за допомогою Metal для прискорення процесу.

- Визначення областей тексту на зображеннях за допомогою Vision Framework.
- Виконання OCR та виведення результатів.
- Візуалізація результатів розпізнавання тексту (графічне представлення).

Розробка має потенціал для застосування в широкому спектрі завдань, таких як автоматизовані системи для аналізу документів, сканування текстів і забезпечення доступу до інформації для людей з обмеженими можливостями. Вона може бути також інтегрована в комплекси автоматизації для обробки відео і зображень на підприємствах та в наукових дослідженнях.

У майбутньому система може бути адаптована для інтеграції з іншими технологіями автоматизації, наприклад, для контролю якості в промислових процесах або в медичних додатках, де важливо швидко і точно зчитувати текст з документів або зображень.

РОЗДІЛ 4. ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вступна частина

Під час виконання дипломного проекту необхідно було мобільний додаток пошуку та реєстрації місць. Розрахунок економічної ефективності дозволить обґрунтувати доцільність розробки.

Техніко-економічне обґрунтування (ТЕО) – це комплект розрахунково-аналітичних документів, що містять основні технічні і організаційні рішення, розрахунково-кошторисні, оціночні та інші показники, що дозволяють розглядати доцільність і ефективність інвестиційного проекту.

Техніко-економічне обґрунтування є необхідним дослідженням, в ході якого проводиться ряд робіт по вивченню і аналізу всіх складових інвестиційного проекту і розробці термінів повернення вкладених в бізнес засобів. Метою техніко-економічного обґрунтування є аргументація доцільності вибору нового устаткування, технології виробництва або процесу.

ТЕО може проводитися по самих різних науково-технічних заходах, які можуть здійснюватися як на конкретному підприємстві, так і в масштабах всього господарства країни.

Розробка ТЕО будь-якого проекту завжди проводиться перед ухваленням рішення про проектування і серійне виробництво певного устаткування або при ухваленні рішення про створення якого-небудь виробництва, орієнтованого на певний вид продукції.

4.2 Розрахунок витрат на створення й експлуатацію програми

Керуючись тим, що середня основна і додаткова заробітна плата (20% від основної) заробітна плата розробників програмного забезпечення складає 12000 грн/міс, а середня вартість сучасного ПК складає 15000 грн, і вартість кіловат-години електроенергії складає 1,68 грн, розрахуємо вартість розробки програми. Таблиця витрат на допоміжні матеріали приведені в таблиці 4.1.

Пункти витрат	Сума, грн
---------------	-----------

Інтернет	110,00
Папір	90,00
Заправка картриджа	90,00
Інші витрати	1000
Всього матеріали	1210

Таблиця 4.1 - Витрати на допоміжні матеріали

Вартість програми розраховується по формулі 5.1:

$$C_{\text{пр}} = (Z_{\text{зп}} + Z_{\text{сз}} + Z_{\text{зв}} + Z_{\text{е}})T + Z_{\text{м}}, \text{ де} \quad (4.1)$$

T – тривалість розробки, міс.;

$Z_{\text{зп}}$ – основна і додаткова заробітна плата персоналу, грн.;

$Z_{\text{сз}}$ – відрахування на соціальні заходи (30% від основної і додаткової заробітної плати, грн.);

$Z_{\text{зв}}$ – загальногосподарські витрати (10% від основної заробітної плати, грн.);

$Z_{\text{е}}$ – витрати на електроенергію;

$Z_{\text{м}}$ – витрати на основні і допоміжні матеріали;

$Z_{\text{е}}$ при споживанні потужності 240 Вт.

Кількість календарних днів - 30, вихідних – 8, тривалість роботи – 8 годин.
Тривалості роботи на місяць складає $(30-8)*8 = 176$ годин, вартість кіловат-години електроенергії 1,68 грн., з цього слідує:

$$Z_{\text{е}} = 176 * 1.68 * 0,24 = 70,96 \text{ грн.};$$

$$C_{\text{пр}} = (12000 + 3600 + 1200 + 70,96) * 3 + 1210 = 84812,88 \text{ грн.};$$

Витрати на створення та експлуатацію програми $Z_{\text{се}}$ в 1-ий рік експлуатації обчислюється по формулі 5.2:

$$Z_{\text{се}} = C_{\text{пр}} + Z_{\text{ев}}, \text{ де} \quad (4.2)$$

$C_{\text{пр}}$ – вартість створення та впровадження програми;

$Z_{\text{ев}}$ – експлуатаційні витрати для ПК за рік (формула 5.3).

$$Z_{\text{ев}} = A_{\text{об}} + Z_{\text{м}} + Z_{\text{е}} \quad (4.3)$$

$A_{\text{об}}$ – амортизаційні відрахування на устаткування;

$Z_{\text{м}}$ – витрати на основні і допоміжні матеріали;

$Z_{\text{е}}$ – витрати на електроенергію.;

Таблиця витрат на розробку програми представлені в таблиці 4.2.

Найменування витрат	Одиниця вимірювання	Кількість
Термін розробки	міс	3
Основна та допоміжна заробітна плата	грн.	12000,00
Відрахування на соціальні заходи	грн.	15000,00
Загальногосподарчі витрати	грн.	3000,00
Витрати на головні та допоміжні матеріали	грн.	1210,00
Витрати на електроенергію	грн.	70,96

Таблиця 4.2 – Витрати на розробку програми

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{об} = 15000 * 0,6 = 9000,00 \text{ грн.};$$

$$З_{ев} = 9000,00 + 1210,00 + 70,96 = 10280,96 \text{ грн.};$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$ВМ = 15000 * 0,05 = 7500,00 \text{ грн.};$$

Річний обсяг робіт ПК у годинах визначається за формулою 5.4:

$$\Phi_m = 249 * TC, \text{ де} \quad (5.4)$$

ТС – це середнє місячне завантаження устаткування (близько 8 год.), де 249 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПК складе:

$$\Phi_m = 249 * 8 = 1992 \text{ годин};$$

Витрати на електроенергію $З_e$ при 1992 годинах роботи устаткування в рік складуть:

$$З_e = 1992 * 0,24 * 1,68 = 803,17 \text{ грн.};$$

Експлуатаційні витрати для ПЕОМ за рік складуть:

У перший рік витрати на створення й експлуатацію програми складуть:

$$З_{сз} = 84812,88 + 803,17 = 86826,05 \text{ грн.};$$

4.3 Розрахунок економічної ефективності

Визначимо пряму економічну ефективність ґрунтуючись на тім, що впровадження програмного продукту вивільняє 0,7 працівника, з урахуванням затрат на супроводження програми.

Заробітна платня 0,7 працівника в рік складає:

$$(6400 * 12) * 0,7 = 53760,00 \text{ грн.};$$

4.4 Розрахунок часу окупності програми

Розрахуємо за яку кількість місяців програмний продукт окупиться за формулою 5.6:

$$T_{\text{ок}} = \frac{\Delta K}{\Delta C} = \frac{19039,48}{49880,26} = 0,38 \text{р.} \quad (4.6)$$

Таким чином витрати на створення програмного продукту окупляться приблизно за 4 місяці експлуатації.

4.5 Висновок

Виходячи з розрахунків мобільний додаток окупить себе за 4 місяці. Даний додаток є більш зручним в користуванні. Він дозволяє автоматизувати роботу та прискорити процес реєстрацію місць і, таким чином, збільшує ефективність роботи. Усе це вказує на те, що існує необхідність створення даного програмного додатку.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

5.1 Вступ

Згідно статі 1 Закону України «Про охорону праці» охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Даний розділ дипломного проекту присвячений розгляду питань охорони праці при роботі за комп'ютером.

5.2 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці програмного комплексу

5.2.1 Організація робочого місця

Конструкція робочого місця користувача ПК має забезпечувати підтримання оптимальної робочої пози з такими ергономічними характеристиками: ступні ніг – на підлозі або на підставці для ніг; стегна – в горизонтальній площині; передпліччя – вертикально; лікті – під кутом 70 – 90 град. до вертикальної площини; зап'ястя зігнуті під кутом не більше 20 град. відносно горизонтальної площини, нахил голови – 15 – 20 град. відносно вертикальної площини.

Якщо користування ПК є основним видом діяльності, то ПК і його периферійні пристрої (принтер, сканер тощо) розміщується на основному робочому столі, як правило, з лівого боку.

Якщо використання ПК є періодичним, то він, як правило, розміщується на приставному столі, переважно з лівого боку від основного робочого столу.

Кут між поздовжніми осями основного та приставного столів має бути 90 – 140 град.

Висота робочої поверхні столу для ПК має бути в межах 680 – 800 мм, а ширина – забезпечувати можливість виконання операцій в зоні досяжності моторного поля.

Рекомендовані розміри столу: висота – 725 мм, ширина – 600 -1400 мм, глибина – 800 – 1000 мм.

Робочий стіл для ПК повинен мати простір для ніг висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм, на рівні

витагнутої ноги – не менше 650 мм. Робочий стіл для ПК, як правило, має бути обладнаним підставкою для ніг шириною не менше 300 мм та глибиною не менше 400 мм, з можливістю регулювання по висоті в межах 150 мм та кута нахилу опорної поверхні – в межах 20 град. Підставка повинна мати рифлену поверхню та бортик на передньому краї заввишки 10 мм. Застосування підставки для ніг тими, у кого ноги не дістають до підлоги, є обов’язковим.

Робоче сидіння (сидіння, стілець, крісло) користувача ПК повинно мати такі основні елементи: сидіння, спинку, стаціонарні або знімні підлокітники.

Монітор та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів. Розташування монітору має забезпечувати зручність зорового спостереження у вертикальній площині під кутом ± 30 град. від лінії зору працівника.

Клавіатуру слід розміщувати на поверхні столу або на спеціальній, регульований за висотою, робочій поверхні окремо від столу на відстані 100 – 300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах 5 – 15 град.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість монітору, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля: по висоті 900 – 1300 мм, по глибині 400 – 500 мм.

5.2.2 Мікроклімат робочої зони програміста

Приміщення для роботи з персональними комп’ютерами мають бути обладнані системами опалення, кондиціонування повітря, або припливно-втяжною вентиляцією. У приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури,

відносної вологості й рухливості повітря у відповідності до санітарних норм

Таблиця 5.1 - мікроклімат температури, відносної вологості й рухливості повітря у відповідності до санітарних норм

Пора року		Температура повітря, град. С	Відносна вологість повітря, %	Швидкість руху повітря, м/с

	Категорія робіт	Оптимальна	Оптимальна	Оптимальна
Холодна	Легка 1-а	22-24	40-60	0,1
	Легка 1-б	21-23	40-60	0,1
Тепла	Легка 1-а	23-25	40-60	0,1
	Легка 1-б	22-24	40-60	0,2

Рівні позитивних та негативних іонів у повітрі мають відповідати санітарно – гігієнічним нормам №2152 – 80.

Таблиця 5.2 - «Рівні позитивних та негативних іонів у повітрі мають відповідати санітарно – гігієнічним норм»

Мінімально необхідні	400	600
Оптимальні	1500-3000	3000-500
Максимально допустимі	5000	5000

Для підтримки допустимих значень мікроклімату та концентрації позитивних та негативних іонів необхідно передбачати установки або прилади зволоження та/або штучної іонізації, кондиціонування повітря.

5.2.3 Освітлення робочого місця

Відносно вікон робоче місце повинно бути розміщено так, щоб природне світло було збоку, переважно з лівого та забезпечувати коефіцієнт природної

освітленості не нижче 1,5 %. Робоче місце, обладнане ПК повинно бути розташоване так, щоб уникнути попадання в очі прямого світла. Джерела штучного світла рекомендується розташувати з обох сторін від екрану паралельно напрямку зору. Щоб уникнути світових блисків від екрану, клавіатури, освітлювальних пристроїв, сонця в напрямку очей необхідно застосовувати антиблискові сітки, спеціальні фільтри для екрану, захисні козирки, жалюзі на вікнах. Вікна приміщень повинні мати регульовальні пристрої для відкривання.

Штучне освітлення приміщення має бути обладнане системою загального рівномірного освітлення. У приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення). Загальне освітлення має бути виконане у вигляді суцільних або переривчатих ліній світильників, що

розміщуються збоку від робочих місць (переважно зліва) паралельно лінії зору працівників. Застосування світильників без розсіювачів та екранних сіток забороняється.

Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. Допускається у світильниках місцевого освітлення застосовувати лампи розжарювання. Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 град. до 90 град. відносно вертикалі в подовжній і поперечній площинах повинна складати не більше 200 кд/кв. м , а захисний кут світильників повинен бути не більшим за 40 град. Коефіцієнт запасу для освітлювальної установки загального освітлення слід приймати рівним 1,4. Коефіцієнт пульсації повинен не перевищувати 5 %. Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300 – 500 лк.

У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж до 300 лк. Світильники місцевого освітлення повинні мати напівпрозорий відбивач світла з захисним кутом не меншим за 40 град. Необхідно передбачити обмеження прямої близькості від джерела природного та штучного освітлення, при цьому яскравість поверхонь, що світяться (вікна, джерела штучного світла) і перебувають у полі зору, повинна бути не більшою за 200 кд/кв. м. Необхідно обмежувати відбиту близькість шляхом правильного вибору типів світильників та розміщення робочих місць відносно джерел природного та штучного освітлення. При цьому яскравість відблисків на моніторі не повинна перевищувати 40 кд/кв. м, яскравість стелі при застосуванні системи відбивного освітлення не повинна перевищувати 200 кд/кв. м. Необхідно обмежувати нерівномірність розподілу яскравості в полі зору осіб, що працюють з монітором, при цьому відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1, а робочих поверхонь і навколишніх предметів (стіни, обладнання) – 5:1.

5.2.4 Вплив шуму на програміста

Рівні шуму та вібрації на робочих місцях осіб, що працюють з ПК, визначаються відповідно до ДсанПіН 3.3.2-007-98.

Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот представлені в таблиці 30.

Таблиця .3 - Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот

Вид трудової діяльності, робочі місця	Рівні звукового тиску в дБ в октавних смугах із середньогометричними частотами, Гц									Рівні звуку, еквівалентні рівні звуку, дБА/дБАекв.
	31,5	63	125	250	500	1000	2000	4000	8000	
Програмісти	86	71	61	54	49	45	42	40	38	50
Оператори в залах обробки інформації на ПК та оператори комп'ютерного набору	96	83	74	68	63	60	57	55	54	65
В приміщеннях для розташування шумних агрегатів	103	91	83	77	73	70	68	66	64	75

5.2.5 Виробничі випромінювання

Теплові апарати, які використовуються на підприємствах, є джерелами інфрачервоного випромінювання. За фізичною природою інфрачервоне випромінювання являє собою електромагнітні хвилі та потік квантових фотонів.

Ефект дії інфрачервоного випромінювання на людину залежить від довжини хвилі.

- короткохвильове інфрачервоне випромінювання з довжиною хвилі від 0,76 до 1,4 мкм має більшу здатність проникати через шкіру;
- довгохвильове інфрачервоне випромінювання з більшою довжиною хвилі поглинається в основному в епідермісі;

– видиме – кров'ю у шарах дерми та підшкірною жировою клітковиною.

Поглинання інфрачервоних променів різними шарами шкіри призводить до їх нагрівання. Внаслідок цього можливе порушення теплового балансу організму людини. Інфрачервоне випромінювання негативно впливає на функціональний стан центральної нервової системи, виникають зміни у серцево-судинній системі.

Вплив інфрачервоного випромінювання на очі нерідко викликає кон'юнктивіти, помутніння рогівки, спазм зіниць, помутніння кришталика, опік сітчатки, «снігову» сліпоту. Під час опромінення очей випромінюванням інтенсивністю $4,2 \text{ кВт/м}^2$ температура рогівки може досягати 40°C і більше. Постійна дія такого випромінювання на очі викликає професійне захворювання – катаракту.

Ступінь біологічного впливу електромагнітних полів на організм людини залежить від частоти коливань, напруженості та інтенсивності поля, тривалості його впливу

Підвищений рівень електромагнітних випромінювань шкодить здоров'ю людини. Від цього страждає передусім нервова і серцево-судинна системи, виникають головний біль і перевтома, знижується точність робочих рухів, порушується сон. Електромагнітне випромінювання викликає зміни тиску крові, гіпотонію або гіпертонію.

Захистні властивості екранів базуються на ефекті послаблення напруженості електричного поля в просторі поблизу заземленого металевго предмета. Екрани виготовляються у вигляді металевої сітки, розміщеної між екранним простором та джерелом електричного поля; організаційними заходами (проведенням дозиметричного контролю, медичних оглядів, додатковою відпусткою, скороченням робочого часу).

Професійний ризик операторів, які обслуговують комп'ютери, пов'язаний також з можливим опроміненням. Катодне, ультрафіолетове, інфрачервоне, а також радіочастотне випромінювання екрана також можуть становити певну небезпеку. Однак з приводу її серйозності думки спеціалістів розходяться. Одні вважають, що випромінювання від дисплея, як від звичайного телевізора, не перевищує допустимих норм. Інші наполягають, що шкода від дисплейної техніки порівняно з

телевізором значно більша. Останнє пояснюється близькістю екрана та тривалістю роботи з ним.

Усі наведені типи професійного ризику під час роботи з дисплейною технікою, накладаючись на стресові ситуації, нерідко спричиняють нервові, психічні та захворювання серцево-судинної системи. В Україні розроблені й діють нормативні документи, що регламентують роботи з візуальними дисплейними терміналами (ДНАОП 0,00-1,31-99) і затверджені наказом Держнаглядохоронпраці від 10.02.1999 р. № 2 за умови додержання Державних санітарних правил і норм роботи із візуальними дисплейними терміналами електронно-обчислювальних машин 3.3.2.007-98. Одним з найбільш ефективних напрямів вирішення зазначених питань є широке і швидке поширення атестації та раціоналізації робочих місць, пов'язаних із візуальними дисплейними терміналами й електронно-обчислювальною технікою.

5.2.6 Електробезпека. Статична електрика.

Лінія електромережі для живлення персональних комп'ютерів, їх периферійних пристроїв (принтер, сканер тощо) виконується як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Використання нульового робочого провідника як нульового захисного провідника забороняється. Не допускається підключення на щиті до одного контактного затискача нульового робочого та нульового захисного провідників.

У приміщенні, де одночасно експлуатується або обслуговується більше п'яти персональних комп'ютерів, на помітному та доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення.

Персональний комп'ютери, периферійні пристрої повинні підключатися до електромережі тільки з допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників повинні мати спеціальні контакти для підключення нульового захисного провідника. Конструкція їх має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше ніж приєднання

фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Необхідно унеможливити з'єднання контактів фазових провідників з контактами нульового захисного провідника. Неприпустимим є підключення комп'ютерів, периферійних пристроїв до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв.[6]

Електромережі штепсельних з'єднань та електророзеток для живлення персональних комп'ютерів, периферійних пристроїв слід виконувати за магістральною схемою, по 3 – 6 з'єднань або електророзеток в одному колі.

Доточувати проводи можна тільки шляхом паяння з наступним старанним ізолюванням місць з'єднання.

Є неприпустимими:

- експлуатація кабелів та проводів з пошкодженою або такою, що втратила захисні властивості за час експлуатації, ізоляцією; залишення під напругою кабелів та проводів з неізольованими провідниками;
- користування пошкодженими розетками, розгалужувальними та з'єднувальними коробками, вимикачами та іншими електровиробами, а також лампами, скло яких має сліди затемнення або випинання;
- підвішування світильників безпосередньо на струмопровідних проводах, обгортання електроламп і світильників папером, тканиною та іншими горючими матеріалами, експлуатація їх зі знятими ковпаками (розсіювачами);
- використання електроапаратури та приладів в умовах, що не відповідають вказівкам (рекомендаціям) підприємств-виготовлювачів.

5.3 Ергономіка та організація робочого місця

Після проведення аналізу робочого місця програміста були записані певні вимоги :

- Висота робочої поверхні повинна складати 720мм;
- Розмір робочої поверхні 1600 x 1000 мм. Простір для ніг знаходиться під робочою поверхнею з розмірами по глибині 650 мм. Клавіатури розташована на відстані 300мм від краю робочої поверхні. Між екраном монітору та очима повинна зберігатися відстань приблизно 40мм;

– Стілець включає в себе підйомно-поворотний механізм. Висота сидіння повинна бути виставлена в межах 400-500 мм. Кут нахилу спинки регулюється в межах 90 - 110°.

5.4 Висновки

Додержання встановлених правил щодо охорони праці має пряме відношення до ефективності роботи з програмним забезпеченням, оскільки при роботі не повинно виникати дискомфорту.

Слід пам'ятати про мікрокліматичні параметри, про норми освітлення, шуму, вібрації та іонізуючих випромінювань, так як дані чинники впливають на самопочуття людини та як результат, на якість виконуваної роботи.

Окрім цього, необхідно дотримуватись ергономічних вимог до робочого місця, що забезпечить не тільки більш продуктивну роботу працівника, а й може зекономити енергію та час роботи.

Проаналізувавши інформацію щодо сприятливих умов для роботи працівників з ПК, можна зробити висновок, що додержання установлених правил є необхідною складовою для продуктивної роботи працівника

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Забруднення навколишнього середовища надлишками праці

Надмірне використання трудових ресурсів, або надлишки праці, є однією з головних причин забруднення навколишнього середовища в сучасному світі. Інтенсивне виробництво, яке базується на максимальному використанні робочої сили, спричиняє значне навантаження на природні ресурси, призводить до збільшення відходів, підвищення рівня викидів в атмосферу, забруднення води та ґрунтів. В результаті, забруднення довкілля стає серйозною проблемою для здоров'я людини та існування екосистем, порушуючи екологічний баланс.

Збільшення робочих годин і праці без належного відпочинку спричиняє не лише фізичне, а й психічне перевантаження людини, що знижує ефективність праці та здоров'я працівників. Це в свою чергу, впливає на якість праці і спричиняє більші викиди шкідливих газів та відходів у виробництві, погіршуючи умови для життя і навколишнє середовище.

Особливо серйозні наслідки цього явища спостерігаються в індустріальних зонах і великих містах, де висока концентрація виробничих потужностей призводить до значного забруднення атмосферного повітря і водних ресурсів. Забруднення повітря шкідливими газами, такими як вуглекислий газ, оксиди азоту, сірчані з'єднання, призводить до підвищення температури в містах, погіршення якості повітря, що, в свою чергу, знижує якість життя населення.

Забруднення водних ресурсів виникає через масове скидання неочищених стічних вод, що містять важкі метали та хімічні сполуки, що вбивають флору і фауну. Забруднення ґрунтів відбувається через великі обсяги відходів, які не утилізуються належним чином і забруднюють землю. Виснаження природних ресурсів, таких як вода та енергія, також є наслідком надмірної трудової активності, оскільки все більше ресурсів витрачається на безперервне виробництво і задоволення потреб населення.

Загроза, яку несуть надлишки праці, полягає в порушенні не тільки природних, але й соціальних систем. Погіршення здоров'я працівників, збільшення кількості хронічних захворювань, соціальна напруга та зниження якості життя — все це наслідки надмірної трудової експлуатації. В умовах, коли робочі ресурси не мають

чітких обмежень, зростає ризик енергетичних криз, погіршення умов праці і життя для значної частини населення.

Особливу увагу слід звернути на проблему зниження навантаження на екосистеми через раціональне використання трудових ресурсів і збереження природних ресурсів. Це дозволить уникнути надмірного забруднення, а також знизити навантаження на здоров'я людей, забезпечити стійке економічне зростання та гармонійний розвиток суспільства. Рішенням може бути перехід до сталого розвитку, де виробничі потужності працюють з урахуванням екологічних вимог, а організація праці адаптована до природних ресурсів і екосистем.

Контроль за забрудненням навколишнього середовища надлишками праці здійснюється через різні органи та установи, як на національному, так і на міжнародному рівнях. В Україні національні органи, такі як Міністерство охорони навколишнього середовища, Держекоінспекція, Міністерство праці та соціальної політики, а також органи місцевого самоврядування, виконують контроль за дотриманням екологічних і трудових норм. Ці установи забезпечують моніторинг рівнів забруднення, встановлення лімітів на використання ресурсів і викиди, а також сприяють запровадженню екологічно чистих технологій у виробничу діяльність.

На міжнародному рівні існують численні організації, які займаються питаннями охорони навколишнього середовища та трудових прав. Серед них — Організація Об'єднаних Націй (ООН), Міжнародна організація праці (МОП), Всесвітня організація охорони здоров'я (ВООЗ), а також Європейський Союз. Ці організації розробляють міжнародні стандарти і рекомендації щодо управління природними ресурсами, використання трудових ресурсів і контролю за їхнім впливом на навколишнє середовище. Норми і регламенти цих організацій мають за мету збереження екологічної рівноваги і забезпечення здорових умов праці для всіх верств населення.

Що стосується конкретних приладів для вимірювання забруднень, то для контролю забруднення навколишнього середовища надлишками праці використовуються різноманітні інструменти та технічні засоби, які дозволяють точно вимірювати рівні забруднення. Для вимірювання електромагнітного забруднення, наприклад, застосовують прилади такі як МТМ-01, ВЕ-МЕТР-АТ-003,

BE-50, ПЗ-41, АТТ-2593, Norda NBM-550, Norda SRM-3000, Norda SRM-3006. Ці прилади дозволяють вимірювати напруженість магнітного поля, щільність електромагнітного випромінювання, а також інші параметри, які є критичними для визначення рівня забруднення навколишнього середовища техногенними факторами.

Також для контролю викидів забруднюючих речовин в атмосферу та водні ресурси використовуються спеціалізовані системи моніторингу, що включають прилади для вимірювання концентрації шкідливих газів (наприклад, газоаналізатори) та системи для моніторингу стану водних і ґрунтових ресурсів. Інструменти, такі як аналізатори газів і датчики рівня забруднень, відіграють важливу роль у забезпеченні дотримання норм безпеки та зниженні впливу людської діяльності на навколишнє середовище.

Таким чином, для ефективного контролю за забрудненням навколишнього середовища надлишками праці необхідна злагоджена робота на рівні держави та міжнародних організацій, впровадження систем моніторингу, встановлення чітких норм і стандартів, а також активне сприяння інноваціям в галузі сталого розвитку і ресурсозбереження. Це дозволить забезпечити гармонійне співіснування людини і природи в умовах сучасної економіки.

ВИСНОВКИ

1. Проаналізовано можливості та методи обробки даних за допомогою сучасних технологій Vision та Metal. Було визначено, що для використання основних методів обробки зображень і відео, таких як OCR (оптичне розпізнавання тексту), користувачу не потрібно заглиблюватися в складні математичні процеси. Достатньо розуміти принципи роботи технологій і алгоритмів. Щоб забезпечити ефективність і зручність використання цих методів, програмне забезпечення повинно відповідати кільком важливим критеріям: підтримка обробки зображень і відео в реальному часі, можливість інтеграції з різними системами, зручний інтерфейс, високий рівень точності розпізнавання, а також доступність для кінцевого користувача.

2. Досліджено методології та підходи до обробки зображень і відео в реальному часі для OCR. Проаналізовано принципи роботи з Vision Framework для виконання OCR на основі відеопотоку з камери, а також роль Metal в оптимізації продуктивності при роботі з зображеннями. Були розглянуті ключові етапи обробки даних, методи масштабування, виявлення текстових блоків та їх подальша обробка для отримання текстової інформації. Визначено, що обробка кожного кадру відео може зайняти значний час, тому важливо застосовувати асинхронні методи для забезпечення швидкості роботи додатка.

3. Розроблено новий оптимізований мобільний додаток для сканування тексту з камери оптимізовану під мобільну платформу iOS інформаційну, проведено практичну реалізацію з виводом тексту в консоль, описані всі кроки і наслідки оптимізації.

4. Проаналізовано тенденції підвищення ефективності, на підставі чого з урахуванням непрямих чинників визначено показники економічної ефективності впровадження нового оптимізованого рішення.

5. Розглянуто заходи щодо охорони праці на автоматизованому робочому місці користувача програмного забезпечення.

6. Проаналізовано шкідливі фактори щодо забруднення навколишнього середовища від наслідків людської діяльності; розглянуто заходи щодо охорони навколишнього середовища.

Наукова новизна. В даній роботі вирішується завдання створення ефективної системи для обробки відео і зображень з використанням передових технологій Vision та Metal, що дозволяє автоматизувати процес OCR в реальному часі. Розроблена система є інноваційною, оскільки поєднує швидкість обробки з високою точністю розпізнавання тексту та оптимізацією ресурсоемних операцій за допомогою GPU.

Наукове значення роботи. Отримані результати сприяють розвитку технологій комп'ютерного зору, особливо в області реального часу, де важливою задачею є швидкість і точність обробки даних. Вони також вносять вклад у розвиток методів оптимізації обробки зображень та відео в мобільних і настільних системах.

Практичне значення розроблена система має широке застосування в багатьох сферах, таких як автоматизація процесів у сферах документообігу, перекладу, допомоги людям з обмеженими можливостями (для читання тексту з зображень) і в інших областях, де необхідно проводити швидку обробку відео і зображень з метою извлечения информации.

Апробація результатів досліджень. Основні положення і результати роботи представлені науково-практичній інтернет конференції «Світ наукових досліджень. Випуск 36». Автор: Партас Віктор Кирилович, кандидат технічних наук, доцент, Національний університет кораблебудування, м. Миколаїв; Дробот Володимир Вадимович, магістрант, Національний університет кораблебудування, м. Миколаїв. Дата: 17.12.2024 – 18.12.2024. Місто Миколаїв.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація SWIFT. URL: <https://swift.org/documentation/> (дата звернення 21.11.2024)
2. Головний сайт Apple : <https://www.apple.com/ru/swift/> (дата звернення 15.05.202)
3. Інтернет-ресурс Medium. URL: <https://medium.com> (дата звернення 15.04.2021)
4. Документація Apple Vision Framework. URL: <https://developer.apple.com/documentation/vision>
5. Документація Apple Metal. URL: <https://developer.apple.com/documentation/metal>
6. Дослідження в області OCR: <https://github.com/tesseract-ocr/tesseract>
7. D. C. S. W. (2017). "Optical Character Recognition: A Literature Review". International Journal of Computer Applications.
8. Z. Zhang, S. Hwang, X. Xu, Y. B. (2019). "Deep Learning for OCR."
9. Szeliski, R. (2010). "Computer Vision: Algorithms and Applications." Springer.
10. Gonzalez, R. C., & Woods, R. E. (2008). "Digital Image Processing." Pearson
11. Bai, X., & Liu, Z. (2019). "A Comprehensive Review of Deep Learning for OCR". Journal of Computational Vision and Image Processing : <https://www.sciencedirect.com/science/article/pii/S1877055820300215>
12. Jafari, M., & Naveh, N. (2017). "High-Performance Image and Video Processing with Metal". Apple Developer Technical Documentation : <https://developer.apple.com/library/archive/documentation/Metal/Conceptual/MetalPerformanceShaders/Introduction.html>
13. Дослідження ефективності алгоритмів OCR для обробки тексту з зображень у Swift-додатках : <https://www.economy-confer.com.ua/full-article/5935/>