

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

«До захисту допущено»

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ

_____ (підпис) (ініціали, прізвище)

“ ____ ” червня 2023 р.

**Дипломний проект
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Системне програмування та спеціалізовані комп'ютерні системи»
спеціальності 123 комп'ютерна інженерія
на тему: «Модуль розпізнавання фокусу уваги користувача
комп'ютерної системи»**

Виконав:

студент IV курсу, групи KB-93

Шевченко Олександр Валерійович

(прізвище, ім'я, по батькові)

_____ (підпис)

Керівник проф. каф. СПіСКС, д.т.н. проф Терейковський І. А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М.

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

_____ (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2023 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма

«Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) **Віталій РОМАНКЕВИЧ**
(ініціали, прізвище)

«__» червня 2023 р.

ЗАВДАННЯ

на дипломний проєкт студента

Шевченко Олександр Валерійович
(прізвище, ім'я, по батькові)

1. Тема проєкту Модуль розпізнавання фокусу уваги користувача комп'ютерної системи, керівник проєкту проф. каф. СПіКСС, д.т.н. проф Терейковський І.А.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» травня 2023р. №__

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту: див. ТЗ.

4. Зміст пояснювальної записки:

- 1) аналіз існуючих рішень та обґрунтування теми дипломного проєкту;
- 2) інструменти та методи розробки;
- 3) розробка архітектури системи
- 4) розробка додатку та експериментальні дослідження

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

- Взаємодія модулів системи. Схема структурна.
- Діаграма в нотації UML. Діаграма класів
- Алгоритм навчання нейромережі. Схема алгоритму.
- Алгоритм розпізнавання фокусу уваги. Схема структурна.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, к.т.н.		

7. Дата видачі завдання 30 жовтня 2023р. _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	15.04.2023
2.	Розроблення та узгодження технічного завдання	28.04.2023
3.	Аналіз існуючих рішень	05.05.2023
4.	Підготовка матеріалів першого розділу дипломного проєкту	10.05.2023
5.	Підготовка матеріалів другого розділу дипломного проєкту	14.05.2023
6.	Підготовка матеріалів третього розділу дипломного проєкту	16.05.2023
7.	Підготовка графічної частини дипломного проєкту	18.05.2023
8.	Оформлення документації дипломного проєкту	20.05.2023
9.	Попередній огляд матеріалів диплому на кафедрі	26.05.2023

Студент

(підпис)

Шевченко О.В.

(ініціали, прізвище)

Керівник проєкту

(підпис)

Терейковський І.А.

(ініціали, прізвище)

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (44 сторінки, 4 додатки, 8 рисунків)

Цей дослідницький проєкт зосереджений на розробці комп'ютерної системи для розпізнавання фокусу уваги користувача в комп'ютерній системі.

Основна мета полягає в тому, щоб розробити та реалізувати модуль, який точно визначає фокус уваги користувача, покращуючи взаємодію людини з комп'ютером і покращуючи досвід користувача.

Об'єктом розробки виступає комп'ютерна система, що дозволяє розпізнавати та аналізувати спрямованість уваги користувача. Вона включає в себе різні функції для досягнення цієї мети.

У процесі розробки виконується кілька завдань. До них відносяться аналіз вимог до програмного продукту та формулювання детальних вимог до модуля.

Для опису функціональності, архітектури та технічних характеристик програмного забезпечення готується документація. Методи прийняття рішень для точного розпізнавання фокусу уваги користувача ретельно відібрані та обґрунтовані. Ці методи можуть включати алгоритми машинного навчання, методи статистичного аналізу або підходи на основі правил. Вибір базується на таких факторах, як ефективність, точність, масштабованість, можливість інтерпретації та здійсненність у рамках обмежень проєкту.

Ключові слова: комп'ютерна система, фокус уваги користувача, розробка модуля, алгоритм, вагова інтерполяція, методи прийняття рішень, взаємодія людини з комп'ютером, досвід користувача.

ABSTRACT

The diploma project includes an explanatory note (44 pages, 4 appendices, 8 figures).

This research project focuses on the development of a computer system to recognize the user's focus of attention in a computer system.

The main goal is to design and implement a module that accurately determines the focus of the user's attention, improving human-computer interaction and improving the user experience.

The object of development is a computer system that allows recognizing and analyzing the focus of the user's attention. It includes various features to achieve this goal.

Several tasks are performed during the development process. These include the analysis of requirements for the software product and the formulation of detailed requirements for the module.

Documentation is prepared to describe the functionality, architecture and technical characteristics of the software. The decision-making methods for accurately recognizing the user's focus of attention are carefully selected and justified. These techniques may include machine learning algorithms, statistical analysis techniques, or rule-based approaches. Selection is based on factors such as efficiency, accuracy, scalability, interpretability, and feasibility within project constraints.

Keywords: computer system, user focus, module development, algorithm, weighted interpolation, decision-making methods, human-computer interaction, user experience.

[illegible]

[illegible]

[illegible]

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ	2
5. ТЕХНІЧНІ ВИМОГИ	2
5.1. Вимоги до програмного продукту, що розробляється.....	2
5.2. Вимоги до апаратного забезпечення.....	3
5.3. Вимоги до програмного та апаратного забезпечення користувача.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					ІАЛЦ. 467200.002 ТЗ			
Зм.	Арк	№ докум	Підпис	Дата				
Розроб.		Шевченко О.В.			«Модуль розпізнавання фокусу уваги користувача комп'ютерної системи» Технічне завдання	Літ.	Арк.	Арк-ів
Перевір.		Терейковський І.А.					1	4
						КПІ ім. Ігоря Сікорського, ФПМ КВ – 93		
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В. О.						

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Модуль розпізнавання фокусу уваги користувача комп'ютерної системи»

Галузь застосування: організація прийому та обробки вхідних відеоданих.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення модулю розпізнавання фокусу уваги користувача комп'ютерної системи.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до програмного продукту, що розробляється

- сумісність з операційними системами: Windows, Linux.
- можливість роботи з різними типами медіаданих;
- можливість обробки в режимі реального часу;
- налаштування та адаптивність;

					ІАЛЦ. 467200.002 ТЗ	Арк
						2
Зм	Арк.	№ докум.	Підпис	Дата		

- відповідність стандартам конфіденційності та безпеки.

5.2. Вимоги до апаратного забезпечення

- Процесор: 4-ядерний, Intel CORE i-3;
- Оперативна пам'ять: 16 ГБ;
- Місце на жорсткому диску 40 ГБ;

5.3. Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows, Linux.
- Python 12.1

					ІАЛЦ. 467200.002 ТЗ	Арк
						3
Зм	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	15.04.2023
2.	Розроблення та узгодження технічного завдання	28.04.2023
3.	Аналіз існуючих рішень	05.05.2023
4.	Підготовка матеріалів першого розділу дипломного проєкту	10.05.2023
5.	Підготовка матеріалів другого розділу дипломного проєкту	14.05.2023
6.	Підготовка матеріалів третього розділу дипломного проєкту	16.05.2023
7.	Підготовка графічної частини дипломного проєкту	18.05.2023
8.	Оформлення документації дипломного проєкту	20.05.2023
9.	Попередній огляд матеріалів диплому на кафедрі	26.05.2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	2
ВСТУП.....	3
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	4
1.1. Постановка задачі виділення обличчя.	4
1.2. Аналіз існуючих засобів.....	9
1.3. Обґрунтування теми дипломного проєкту	15
2. ІНСТРУМЕНТИ ТА МЕТОДИ РОЗРОБКИ	18
2.1. Методи та алгоритми попередньої обробки.....	18
2.2. Розробка архітектури нейронної мережі	25
2.3. Висновки до другого розділу	28
3. РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ	30
3.1. Проектування модуля розпізнавання фокусу уваги	30
3.2. База даних для зберігання даних про обличчя людини	36
3.3. Висновки до третього розділу	39
4. РОЗРОБКА ДОДАТКУ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	40
4.1. Опис інструментарію	40
4.2. Опис інтерфейсу програми та основні способи його використання	45
4.3. Експериментальні дослідження.....	49
ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	55

					ІАЛЦ.045440.004 ПЗ			
Зм.	Арк	№ докум	Підпис	Дата				
Розроб.		Шевченко О.В.			«Модуль розпізнавання фокусу уваги користувача комп'ютерної системи» Пояснювальна записка	Літ.	Арк.	Арк-ів
Перевір.		Терейковський І.А.					1	
						КПІ ім. Ігоря Сікорського, ФПМ КВ – 93		
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В. О.						

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

AR – Augmented Reality

BB – Bounding Box

CNN – Convolutional Neural Network

DL – Deep Learning

DNN – Deep Neural Network

GPU – Graphics Processing Unit

HF – High Frequency

MTCNN – Multi-task Cascaded Convolutional Networks

ReLU – Rectified Linear Unit

RNN – Recurrent Neural Network

ROI – Region of Interest

SNM – Stochastic Neighbor Matching

TPU – Tensor Processing Unit

VR – Virtual Reality

					ІАЛЦ.045440.004 ПЗ	Арк
						2
Зм	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Дослідження розпізнавання фокусу уваги користувача в комп'ютерних системах привернуло значну увагу в останні роки. Ця область досліджень лежить на перетині взаємодії людини з комп'ютером, комп'ютерного зору та штучного інтелекту з метою підвищення ефективності та результативності комп'ютерних систем шляхом розуміння та інтерпретації концентрації уваги користувача.

Здатність точно визначати та відстежувати фокус уваги користувача має величезний потенціал у різних областях і програмах. У таких сферах, як віртуальна реальність, доповнена реальність, та відеоспостереження Розпізнавання концентрації уваги користувача відіграє вирішальну роль у покращенні загального досвіду користувача та взаємодії.

Основна мета розробки — точно виявляти та відстежувати людські обличчя або певні регіони інтересу, такі як об'єкти чи області в зображенні чи відеопотоці. Аналізуючи увагу користувача, комп'ютерна система може динамічно адаптувати свою поведінку, відповіді та вміст, щоб забезпечити більш індивідуальний та інтуїтивно зрозумілий досвід.

Таким чином, модуль для розпізнавання фокусу уваги користувача в комп'ютерній системі відіграє ключову роль у вдосконаленні взаємодії людини з комп'ютером. Завдяки точному виявленню та відстеженню людських облич або цікавих регіонів ці системи адаптують свою поведінку, вміст і відповіді відповідно до уваги користувача, забезпечуючи більш інтуїтивно зрозумілий, захоплюючий і персоналізований досвід. Ця сфера охоплює різноманітний спектр завдань, алгоритмів і міркувань, і поточні дослідження продовжують розширювати межі розпізнавання уваги та його практичного застосування.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1. Постановка задачі виділення обличчя.

Завдання вибору обличчя є критичним компонентом у процесі розпізнавання фокусу уваги користувача. Завдання передбачає ідентифікацію та відстеження області інтересу (ROI) у відеопотоці, яка відповідає обличчю користувача. Це завдання є ключовим компонентом у процесі розпізнавання фокусу уваги користувача.

Наступна діаграма допомагає краще зрозуміти роль процесу виділення обличчя. Вона ілюструє загальний процес.

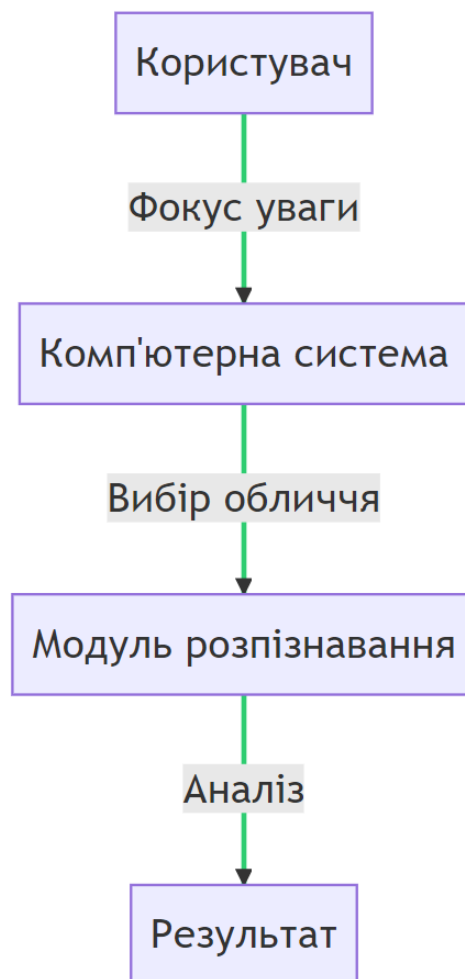


Рис. 1.1. Діаграма процесу розпізнавання фокусу уваги користувача в комп'ютерній системі

Діаграма на рис. 1.1 представляє наступні кроки:

1. Користувач взаємодіє з комп'ютерною системою. Фокус уваги користувача спрямований на комп'ютерну систему.
2. Потім комп'ютерна система виконує вибір обличчя. Це важливий крок, на якому система ідентифікує та вибирає обличчя користувача з вхідних даних (наприклад, відеопотоку чи зображення).
3. Потім вибране обличчя передається в модуль розпізнавання. Цей модуль відповідає за аналіз обличчя користувача та визначення фокусу уваги.
4. Модуль розпізнавання виконує аналіз і видає результат. Цей результат представляє фокус уваги користувача, який може бути певною областю на екрані, об'єктом або напрямком.

Цей процес є повторюваним і триває доти, доки користувач взаємодіє з комп'ютерною системою. У цьому процесі фокус уваги користувача захоплюється комп'ютерною системою. Потім завдання вибору обличчя виконує модуль розпізнавання, який аналізує дані та видає результат.

Постановка задачі вибору граней передбачає визначення конкретних вимог і обмежень задачі. Це може включати потребу в продуктивності в режимі реального часу, здатність працювати з різними умовами освітлення та орієнтацією обличчя, а також потребу в надійності проти змін зовнішнього вигляду обличчя через такі фактори, як вік, стать та етнічна приналежність.

Завдання вибору обличчя є складним і вимагає застосування передових технологій обробки зображень і машинного навчання. Тому розробка модуля для виконання цього завдання є серйозним викликом, але

					ІАЛЦ.045440.004 ПЗ	Арк
						5
Зм	Арк.	№ докум.	Підпис	Дата		

також і можливістю зробити внесок у розвиток технологій взаємодії людини з комп'ютером.

У контексті модуля розпізнавання фокусу уваги користувача в комп'ютерній системі завдання вибору обличчя має першочергове значення. Вибір обличчя стосується процесу точної ідентифікації та виділення людських облич із навколишнього середовища чи інших об'єктів у зображенні чи відеопотоці. Основна мета вибору обличчя – точне виявлення та відстеження облич, щоб визначити увагу та фокус користувача.

Завдання вибору обличчя відіграє вирішальну роль у різних комп'ютерних системах і програмах. Однією з ключових сфер, де вибір обличчя є важливим, є взаємодія людини з комп'ютером. Завдяки точному розпізнаванню та інтерпретації уваги користувача комп'ютерні системи можливо відповідним чином адаптувати свою поведінку та відповіді. Наприклад, у системі відеоконференцій модуль вибору обличчя може ідентифікувати активного спікера, виявляючи та відстежуючи його обличчя, що дозволяє системі забезпечувати більш плавний і привабливий досвід для всіх учасників.

Системи віртуальної реальності (VR) і доповненої реальності (AR) також значною мірою покладаються на вибір обличчя, щоб покращити досвід користувача. Завдяки точному виявленню та відстеженню обличчя користувача ці системи можуть відтворювати віртуальні об'єкти чи інформацію в режимі реального часу, узгоджуючи їх з поглядом користувача. Це забезпечує більш природну та інтуїтивно зрозумілу взаємодію, коли користувачі можуть просто спрямувати свою увагу на віртуальні об'єкти, щоб взаємодіяти з ними.

Ще одне важливе застосування вибору обличчя – у системах відеоспостереження. Завдяки інтеграції модуля в камери спостереження або системи безпеки стає можливим автоматично ідентифікувати та

					ІАЛЦ.045440.004 ПЗ	Арк
						6
Зм	Арк.	№ докум.	Підпис	Дата		

відстежувати зацікавлених осіб. Модуль вибору обличчя може відігравати вирішальну роль у аналізі відео в реальному часі, допомагаючи персоналу служби безпеки контролювати багатолюдні місця, виявляти потенційні загрози або стежити за конкретними особами через різні канали камер.

Формулювання завдання вибору обличчя передбачає визначення конкретних завдань і вимог до модуля. Це включає визначення необхідного рівня точності та стійкості, вибір відповідних алгоритмів і методів, а також розгляд обчислювальної ефективності системи. Формулювання завдання також передбачає вирішення потенційних проблем і обмежень у виборі обличчя, таких як варіації умов освітлення, оклюзії та зміни поз.

Для точного вибору обличчя використовуються різні техніки та алгоритми. Ці методи можна загалом розділити на два основні підходи: методи на основі функцій і методи на основі машинного навчання. Методи на основі функцій покладаються на виділення певних рис обличчя, таких як краї, текстури або орієнтири обличчя, для виявлення та відстеження облич. З іншого боку, методи, засновані на машинному навчанні, використовують навчальні дані для вивчення шаблонів і особливостей, які відрізняють обличчя від інших об'єктів.

Завдання вибору обличчя також вимагає врахування вимог до обробки в реальному часі, оскільки багато програм вимагають негайної відповіді та аналізу.

Крім того, завдання вибору обличчя включає кілька підзавдань і міркувань, які сприяють його успішному виконанню. До них належать:

Локалізація облич: модуль повинен мати можливість точно локалізувати та ідентифікувати присутність облич у зображенні чи відеопотоці. Це передбачає виявлення області обличчя, яка може відрізнятися за розміром, положенням і орієнтацією. Для локалізації обличчя можна використовувати такі методи, як каскади Хаара, алгоритм

Віоли-Джонса або підходи на основі глибокого навчання, як-от згорточні нейронні мережі (CNN).

Виділення рис обличчя: після виявлення обличчя потрібно виділити відповідні риси обличчя для подальшого аналізу. Ці особливості можуть включати орієнтири обличчя, такі як очі, ніс і рот, а також міміку та жести. Методи виділення ознак можуть включати геометричне моделювання, аналіз текстур або підходи на основі глибокого навчання.

Відстеження обличчя: щоб визначити увагу користувача, модуль повинен бути здатний відстежувати обличчя в послідовних кадрах або відеопослідовностях. Алгоритмам відстеження обличчя потрібно вирішувати такі проблеми, як оклюзії, різні умови освітлення та зміни зовнішнього вигляду обличчя через зміну пози чи виразу обличчя. Для надійного відстеження обличчя можна використовувати такі методи, як оптичний потік, фільтри Калмана або методи відстеження шляхом виявлення.

Виявлення уваги: після локалізації та відстеження обличчя модуль повинен проаналізувати фокус і увагу користувача на виявлених обличчях. Це передбачає визначення обличчя або області обличчя, на яке користувач дивиться в основному або на яке спрямовує свою увагу. Технологія стеження за очима може відігравати важливу роль у визначенні напрямку погляду та приверненні уваги користувача. Інтеграція даних відстеження очей із вибором обличчя може забезпечити точніше та надійніше виявлення уваги.

Обробка в реальному часі: багато комп'ютерних систем вимагають можливостей обробки в реальному часі для ефективної взаємодії та відповіді. Модуль вибору обличчя має бути розроблений для виконання необхідних обчислень у межах обмежень обробки в реальному часі. Це може включати методи оптимізації, апаратне прискорення або паралельну обробку для забезпечення своєчасної та оперативної роботи.

					ІАЛЦ.045440.004 ПЗ	Арк
						8
Зм	Арк.	№ докум.	Підпис	Дата		

Надійність і адаптивність: модуль повинен бути розроблений для обробки різних реальних сценаріїв і потенційних викликів. Це включає в себе стійкість до коливань умов освітлення, змін зовнішнього вигляду обличчя (наприклад, через макіяж, зачіску на обличчі чи аксесуарів) і оклюзій, спричинених предметами або іншими людьми. Крім того, модуль повинен адаптуватися до різних профілів користувачів, враховуючи індивідуальні відмінності в характеристиках обличчя, моделях погляду та поведінці уваги.

Питання конфіденційності: оскільки модуль стосується аналізу та відстеження облич людей, міркування щодо конфіденційності є надзвичайно важливими. Модуль має відповідати правилам конфіденційності та гарантувати, що дані користувача обробляються безпечно та конфіденційно. Він повинен забезпечувати механізми згоди користувача, анонімізації даних, безпечного зберігання та передачі.

Підсумовуючи, завдання вибору обличчя в модулі для розпізнавання фокусу уваги користувача є багатогранним і охоплює різні підзадачі та міркування. Розглядаючи ці аспекти та використовуючи відповідні алгоритми та методи, модуль може точно виявляти, відстежувати та аналізувати обличчя, щоб визначити увагу користувача. Ця можливість має далекосяжні наслідки для взаємодії людини з комп'ютером, віртуальної реальності, доповненої реальності та додатків спостереження, забезпечуючи більш інтуїтивний та персоналізований досвід, поважаючи проблеми конфіденційності.

1.2. Аналіз існуючих засобів

При розробці модуля розпізнавання фокусу уваги користувача в комп'ютерній системі важливе значення має проведення ретельного аналізу існуючих засобів. Цей аналіз передбачає вивчення поточного рівня техніки, технологій і методів, що використовуються в області вибору обличчя та розпізнавання уваги. Оцінюючи існуючі засоби, ми можемо

визначити їхні сильні та слабкі сторони та потенційні прогалини, які можуть спрямовувати проектування та розробку вдосконаленого та інноваційного рішення.

Кілька існуючих рішень використовують механізми уваги для підвищення продуктивності своїх систем. Ці рішення можна умовно розділити на три типи залежно від сфери застосування:

1. Розпізнавання мовних емоцій. Рішення в цій категорії використовують механізми «Фокус-Увага» та «Калібрування-Увага» для покращення зони уваги. Ці механізми дозволяють мережі виявляти частину найбільшої амплітуди в сегменті, тим самим покращуючи розпізнавання мовних емоцій (SER).
2. Розпізнавання дії з кількома видами. Рішення цієї категорії використовують механізми уваги в двопотокових мережах. Ці механізми ефективно витягують інформацію з різних представлень і добре працюють навіть у випадках часткової відсутності перегляду.
3. Розпізнавання поведінки водія. Рішення в цій категорії використовують глибокі нейронні мережі на основі уваги (ADNet). Модель ADNet отримує високу точність розпізнавання поведінки водія при застосуванні механізму уваги.

Наступна діаграма ілюструє ці існуючі рішення:

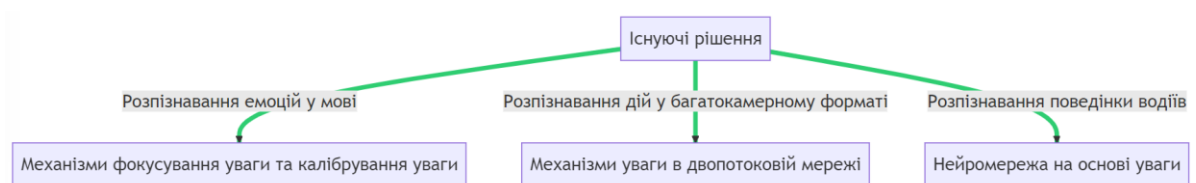


Рис. 1.2. Діаграма існуючих рішень в області розпізнавання уваги та їхнє відношення до різних систем

Кожне з цих рішень має свої переваги та обмеження. Проте всі вони підкреслюють потенціал механізмів уваги для підвищення ефективності

систем розпізнавання. Цей аналіз забезпечує надійну основу для розробки модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі.

Діаграма представляє наступне:

1. Існуючі рішення: це відправна точка, яка представляє поточний стан розпізнавання уваги.
2. Розпізнавання мовних емоцій: це програма для розпізнавання уваги в області мовлення та аналізу емоцій. Рішення, яке тут використовується, — це "механізми фокусування-уваги та калібрування-уваги". Ці механізми призначені для зосередження на конкретних емоційних сигналах у мовленні та відповідного калібрування моделі уваги.
3. Розпізнавання дій у кількох ракурсах: це ще одне застосування розпізнавання уваги, зокрема в контексті розпізнавання дій із кількох ракурсів. Рішення, яке тут використовується, це «Механізм звернення уваги в двопотокову мережу». Цей механізм використовує двопотокову мережу для обробки різних поглядів і використовує увагу для зосередження на найбільш релевантних частинах дії.
4. Розпізнавання поведінки водія: це програма для розпізнавання уваги в області аналізу поведінки водія. Рішення, яке тут використовується, — це «глибинна нейронна мережа на основі уваги». Ця мережа використовує механізми уваги, щоб зосередитися на певній поведінці водія, яка вказує на стан його уваги.

Ці існуючі рішення створюють основу для розробки модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі.

Одним із ключових аспектів аналізу є перегляд різних алгоритмів і методів, що використовуються для виявлення, розпізнавання та

відстеження облич. У літературі було запропоновано численні підходи, кожен зі своїми сильними сторонами та обмеженнями. Традиційні методи, засновані на функціях, такі як алгоритм Віоли-Джонса та каскади Хаара, широко використовувалися для виявлення облич завдяки їх ефективності та можливостям реального часу. Ці методи використовують попередньо визначені риси обличчя та візерунки для ідентифікації та визначення місцезнаходження облич у зображенні чи відеопотоці.

З іншого боку, останніми роками значну увагу привернули методи, засновані на машинному навчанні. Ці підходи використовують потужність глибокого навчання та нейронних мереж, щоб вивчати особливості та шаблони безпосередньо з даних. Згорткові нейронні мережі (CNN) продемонстрували надзвичайний успіх у задачах виявлення та розпізнавання облич, досягнувши високої точності та надійності. Такі мережі, як Single Shot MultiBox Detector (SSD) і алгоритм You Only Look Once (YOLO), показали багатообіцяючі результати в розпізнаванні облич у реальному часі.

Окрім виявлення обличчя, розпізнавання уваги також передбачає відстеження та аналіз погляду та фокусу користувача. Методи стеження за очима відіграють фундаментальну роль у цьому процесі. Для фіксації рухів очей і визначення уваги користувача використовуються різні методи, такі як відображення рогівки, інфрачервоне відстеження або підходи на основі веб-камери. Комерційні пристрої для стеження за очима, такі як Tobii або EyeSight, широко застосовуються для досліджень і застосування, забезпечуючи точні й точні вимірювання напрямку погляду.

Крім того, аналіз існуючих засобів включає вивчення додатків і областей, де актуальні вибір обличчя та розпізнавання уваги. Взаємодія людина-комп'ютер (HCI) є однією з основних областей, які виграють від точного розпізнавання уваги. Це дозволяє комп'ютерній системі розумно реагувати на погляд і увагу користувача, забезпечуючи більш природну та

					ІАЛЦ.045440.004 ПЗ	Арк
						12
Зм	Арк.	№ докум.	Підпис	Дата		

інтуїтивно зрозумілу взаємодію. Системи віртуальної реальності (VR) і доповненої реальності (AR) значною мірою покладаються на розпізнавання уваги, щоб покращити занурення та реалістичність віртуального середовища.

Системи відеоспостереження та безпеки є ще однією областю, де вибір обличчя та розпізнавання уваги є критичними. Завдяки точному визначенню та відстеженню зацікавлених осіб ці системи можуть покращити заходи безпеки та полегшити проактивне виявлення загроз. Технології розпізнавання облич у поєднанні з розпізнаванням уваги дозволяють системі виявляти підозрілі дії або контролювати конкретні цілі, забезпечуючи надійне рішення для спостереження.

Під час аналізу важливо оцінити ефективність, точність і обмеження існуючих засобів. Для оцінки ефективності алгоритмів вибору обличчя можна використовувати такі показники, як рівень виявлення, хибно-позитивні та хибно-негативні результати. Методи розпізнавання уваги можна оцінити на основі їхньої точності уловлювання напрямку та фокусу погляду користувача. Порівняння різних алгоритмів і підходів може допомогти визначити найсучасніші методи та виділити області для вдосконалення.

Крім того, аналіз повинен враховувати обчислювальну ефективність і масштабованість існуючих засобів. Вимоги до обробки в режимі реального часу є вирішальними для багатьох програм, і, отже, можливість виконувати завдання вибору обличчя та розпізнавання уваги в межах часових обмежень є дуже бажаною. Масштабованість також важлива, особливо при роботі з великомасштабними системами або сценаріями з кількома користувачами.

Таким чином, аналіз існуючих засобів дає цінну інформацію про поточний стан технологій вибору обличчя та розпізнавання уваги. Це включає перегляд та оцінку алгоритмів, методів і застосувань у цій галузі.

Розуміючи сильні, слабкі сторони та обмеження існуючих засобів, ми можемо використовувати ці знання для розробки інноваційного та вдосконаленого модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі що перевершує обмеження поточних рішень.

Завдяки аналізу можна виявити кілька ключових спостережень і тенденцій. Наприклад, може стати очевидним, що підходи, засновані на глибокому навчанні, такі як CNN, досягли кращої продуктивності в задачах виявлення та розпізнавання обличч порівняно з традиційними методами на основі функцій. Це розуміння може керувати вибором відповідних алгоритмів і методів для запропонованого модуля.

Крім того, аналіз може виявити потребу в надійності та адаптивності у виборі обличчя та розпізнаванні уваги. У реальних сценаріях часто виникають такі проблеми, як варіації умов освітлення, оклюзії та зміни зовнішнього вигляду обличчя. Таким чином, модуль має бути розроблений таким чином, щоб справлятися з цими проблемами та забезпечувати точні результати в різних умовах.

Крім того, аналіз існуючих засобів може пролити світло на важливість інтеграції технології відстеження очей із вибором обличчя. Вловлюючи напрямок погляду користувача, модуль може точно визначити фокус уваги, забезпечуючи більш повне розуміння когнітивного стану та намірів користувача. Ця інтеграція може значно розширити можливості модуля та забезпечити більш природну та інтуїтивно зрозумілу взаємодію людини з комп'ютером.

Ще один важливий аспект, який слід враховувати під час аналізу, — це етичні наслідки вибору обличчя та розпізнавання уваги. Необхідно звернути увагу на питання конфіденційності та безпеки даних, щоб забезпечити довіру користувачів і дотримуватися правових та етичних норм. Важливо оцінити існуючі засоби з точки зору механізмів захисту конфіденційності, анонімізації даних і механізмів отримання згоди

користувачів, щоб переконатися, що запропонований модуль дотримується етичних стандартів і поважає конфіденційність користувачів.

Крім того, аналіз існуючих засобів повинен враховувати аспекти зручності використання та досвіду користувача систем вибору обличчя та розпізнавання уваги. Модуль має бути інтуїтивно зрозумілим у використанні, бездоганно інтегрованим в інтерфейс користувача та забезпечувати значущий і дієвий зворотний зв'язок з користувачем. Розуміння потреб користувача, уподобань і вимог до взаємодії може керувати процесом проектування та в результаті отримати ефективний і зручний модуль.

Нарешті, аналіз існуючих засобів у сфері вибору обличчя та розпізнавання уваги дає цінну інформацію про стан техніки, технологічні тенденції та сфери застосування. Це допомагає визначити сильні, слабкі сторони та прогалини в поточних рішеннях, які можуть спрямовувати розробку інноваційного та вдосконаленого модуля.

1.3. Обґрунтування теми дипломного проекту

Вибір теми дипломного проекту є відповідальним рішенням, яке потребує ретельного аналізу та обґрунтування. Основна увага проекту зосереджена на розробці модуля для розпізнавання уваги користувача в комп'ютерній системі, теми, яка має значну актуальність і потенціал у поточному технологічному середовищі.

Тема розглядає важливу та зростаючу вимогу у сфері взаємодії людини з комп'ютером. Оскільки технології дедалі більше втручаються в наше повсякденне життя, зростає попит на інтуїтивну та природну взаємодію між людьми та комп'ютерами. Традиційні методи введення, такі як клавіатура та миша, мають свої обмеження щодо виразності та простоти використання. Запропонований модуль спрямований на точне розпізнавання уваги користувача, таким чином пропонуючи більш

					ІАЛЦ.045440.004 ПЗ	Арк
						15
Зм	Арк.	№ докум.	Підпис	Дата		

інтуїтивно зрозумілу та привабливу взаємодію, покращуючи взаємодію з користувачем. Практичні наслідки запропонованого модуля великі та різноманітні. У сфері віртуальної реальності (VR) точне розпізнавання уваги може значно покращити занурення та реалістичність віртуального середовища[1]. Порівнюючи віртуальний вміст із фокусом користувача, ми можемо створити більш захоплюючий та інтерактивний досвід VR. Подібним чином у додатках доповненої реальності (AR) розпізнавання уваги може дозволити системі надавати контекстно релевантну інформацію або розширювати середовище користувача на основі його погляду та фокусу.

Модуль також має потенціал у таких секторах, як освіта, охорона здоров'я та розваги. У навчальних закладах розпізнавання уваги можна використовувати для оцінки залучення студентів і відповідного адаптування навчальних матеріалів[3]. У сфері охорони здоров'я це може допомогти контролювати увагу пацієнта під час реабілітаційних вправ або когнітивних оцінок. У індустрії розваг розпізнавання уваги може забезпечити більш захоплюючий і персоналізований досвід, наприклад адаптивні ігри або персоналізовані рекомендації вмісту. Незважаючи на прогрес у розпізнаванні обличчя та розпізнаванні уваги, у поточних рішеннях все ще існують прогалини та обмеження, зокрема щодо точності, надійності та продуктивності в реальному часі[4]. Пропонований модуль спрямований на усунення цих прогалин, сприяючи розвитку галузі та пропонуючи більш ефективне та надійне рішення для розпізнавання фокусу уваги користувача.

Дипломний проект також відкриває можливість для досліджень та інновацій. Досліджуючи та впроваджуючи нові алгоритми, методи та підходи до інтеграції, ми можемо зробити внесок у базу знань академічної спільноти та потенційно розробити рішення, які перевершують існуючі інструменти. Проект забезпечує платформу для експериментів, оцінки та

					ІАЛЦ.045440.004 ПЗ	Арк
						16
Зм	Арк.	№ докум.	Підпис	Дата		

перевірки запропонованого модуля, що дозволяє вносити наукові внески та практичні результати. Крім того, ця тема відповідає сучасним тенденціям і розробкам комп'ютерного зору, машинного навчання та штучного інтелекту[2]. Проект може використовувати ці досягнення для досягнення високої точності, продуктивності в реальному часі та масштабованості. Дипломний проект також пропонує можливість особистісного зростання та розвитку навичок. Виконавши цей проект, студент зможе покращити своє розуміння комп'ютерного зору, машинного навчання та концепцій взаємодії людини з комп'ютером. Комерційний потенціал технології розпізнавання уваги значний. У міру того як організації прагнуть надати своїм користувачам більш персоналізований і адаптований досвід, попит на системи з увагою буде зростати. Цей дипломний проект може закласти основу для розробки комерційних програм, які покладаються на точне виявлення уваги.

У підсумку, обрана тема дипломного проекту, яка зосереджена на розробці модуля для розпізнавання уваги користувача в комп'ютерній системі, є виправданою з огляду на її важливість у взаємодії людини з комп'ютером, практичні наслідки в різних сферах, існуючі прогалини в поточних рішеннях, можливості для досліджень та інновацій, узгодження з сучасними тенденціями та розвитку особистих навичок. Беручи участь у цьому проекті, студент може зробити свій внесок у цю сферу діяльності, поглибити свої знання та розробити рішення, яке покращить користувацький досвід і забезпечить більш інтуїтивно зрозумілу та привабливу взаємодію з комп'ютерними системами. Незважаючи на значні успіхи в галузі розпізнавання уваги, у поточних рішеннях все ще існують прогалини та обмеження, зокрема щодо точності, надійності та продуктивності в реальному часі[4]. Запропонований модуль спрямований на усунення цих прогалин, сприяючи розвитку галузі та пропонуючи більш ефективні рішення для розпізнавання фокусу уваги.

					ІАЛЦ.045440.004 ПЗ	Арк
						17
Зм	Арк.	№ докум.	Підпис	Дата		

2. ІНСТРУМЕНТИ ТА МЕТОДИ РОЗРОБКИ

2.1. Методи та алгоритми попередньої обробки

У розробці модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі попередня обробка відіграє життєво важливу роль у підвищенні якості вхідних даних і покращенні загальної продуктивності системи. Методи й алгоритми попередньої обробки охоплюють низку методів, спрямованих на очищення, нормалізацію та підготовку вхідних даних перед тим, як вони подадуться в модуль розпізнавання уваги. У цьому розділі розглядаються різні методи попередньої обробки та алгоритми, які можна використовувати в процесі розробки.

Одним із фундаментальних етапів попередньої обробки є отримання та калібрування зображення. Залежно від налаштувань апаратного забезпечення зображення або відеокадри знімаються за допомогою камер або датчиків. Важливо переконатися, що процес отримання зображення є надійним і забезпечує чіткі та точні дані. Методи калібрування, такі як калібрування камери та виправлення спотворень, можуть бути застосовані для виправлення будь-яких геометричних або оптичних спотворень у знятих зображеннях, забезпечуючи точне та послідовне введення.

Зменшення шуму є ще одним важливим етапом попередньої обробки. У реальних сценаріях зображення або відеокадри чутливі до шуму, спричиненого такими факторами, як умови слабкого освітлення, обмеження сенсора або помилки передачі. Для зменшення шуму та покращення якості вхідних даних можна використовувати різні методи шумозаглушення, включаючи просторові фільтри (наприклад, фільтри Гауса, медіанні фільтри) і часові фільтри (наприклад, часове усереднення, фільтрація з компенсацією руху).

Формули та пояснення методів усунення шумів, які ви згадали:

1. Фільтр Гауса: це тип просторового фільтра, який використовується для розмивання зображень і видалення деталей і шуму. Під час обробки зображень фільтр Гауса, також відомий як розмиття за Гаусом, є результатом розмиття зображення за допомогою функції Гауса. Це широко використовуваний ефект у графічному програмному забезпеченні, як правило, для зменшення шуму зображення. Двовимірна функція Гауса (2.1) визначається як:

$$G(x, y) = (1 / 2\pi\sigma^2) * e^{-(x^2 + y^2) / 2\sigma^2} \quad (2.1)$$

де $G(x, y)$ — фільтр Гауса, а σ — стандартне відхилення розподілу Гауса. Чим більше значення σ , тим більше розмиття.

2. Медіанний фільтр: це техніка нелінійної цифрової фільтрації, яка часто використовується для видалення шуму із зображення чи сигналу. Таке зменшення шуму є типовим етапом попередньої обробки для покращення результатів подальшої обробки. Основна ідея медіанного фільтра полягає в тому, щоб проходити один запис сигналу за записом, замінюючи кожен запис медіаною сусідніх записів. Шаблон сусідів називається «вікном», яке ковзає, один за одним, по всьому сигналу. Немає конкретної формули для медіанної фільтрації, це скоріше алгоритмічний процес.



Рисунок 2.1 - Діаграма медіанного фільтру

Діаграма представляє наступне:

1. Медіанний фільтр: це початкова точка процесу. Він представляє застосування медіанного фільтра до зображення.
2. Для кожного пікселя зображення: цей крок являє собою початок циклу, у якому кожен піксель зображення обробляється окремо.
3. Отримати околиці пікселя, визначеного вікном фільтра: на цьому кроці для поточного пікселя ми отримуємо навколишні пікселі. Розмір навколишньої області визначається вікном фільтра.
4. Сортувати значення пікселів у сусідстві: Тут значення пікселів у сусідстві (зібрані на попередньому кроці) сортуються в порядку.
5. Замінити початкове значення пікселя медіаною відсортованих значень: Нарешті, вихідне значення пікселя замінюється середнім значенням із відсортованих значень пікселів околиць. Цей процес повторюється для кожного пікселя зображення.

3. Тимчасове усереднення: це тип тимчасового фільтра, який використовується для зменшення шуму у відеопослідовності. Ідея полягає в тому, щоб взяти середнє значення кожного пікселя за певну кількість кадрів (2.2). Математично це можна представити так:

$$I'(x, y, t) = (1 / N) * \Sigma I(x, y, t - i) \quad (2.2)$$

де $I'(x, y, t)$ — піксель з шумом, $I(x, y, t - i)$ — вихідний піксель у момент часу $t - i$, сума по i від 0 до $N - 1$, а N — кількість кадрів, використаних для усереднення.

4. Фільтрування з компенсацією руху: це вдосконалений тип тимчасового фільтра, який враховує рух у відеопослідовності. Ідея полягає в тому, щоб вирівняти кадри перед усередненням, щоб ми усереднювали відповідні пікселі. Це скоріше алгоритмічний процес, ніж математична формула, і він передбачає оцінку руху між кадрами (зазвичай за допомогою оптичного потоку або зіставлення блоків), вирівнювання кадрів відповідно до оціненого руху, а потім усереднення.

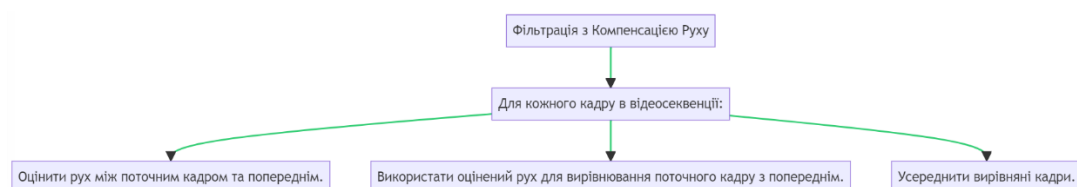


Рисунок 2.2 - Діаграма процесу фільтрування з компенсацією руху

Фільтрування з компенсацією руху:

1. Фільтрування з компенсацією руху: це початкова точка процесу. Він представляє застосування фільтрації з компенсацією руху до відеоряду.
2. Для кожного кадру у відеопослідовності:: цей крок означає початок циклу, у якому кожен кадр у відеопослідовності обробляється окремо.
3. Оцінити рух між поточним кадром і попереднім кадром: на цьому кроці оцінюється рух між поточним кадром і попереднім кадром. Це можна зробити за допомогою таких методів, як оптичний потік або зіставлення блоків.
4. Використовувати оцінений рух для вирівнювання поточного кадру з попереднім: тут оцінений рух використовується для вирівнювання поточного кадру з попереднім кадром. Це забезпечує порівняння однакових областей зображення між кадрами.

5. Усереднювати вирівняні кадри: Нарешті, вирівняні кадри усереднюються. Це може допомогти зменшити шум, зберігаючи рух у відеоряді. Цей процес повторюється для кожного кадру відеоряду.

Нормалізація та стандартизація: вони можуть бути представлені математично таким чином:

Мінімально-максимальна нормалізація:

$$X' = (X - X_{\min}) / (X_{\max} - X_{\min}) \quad (2.3)$$

Нормалізація Z-оцінки:

$$X' = (X - \mu) / \sigma \quad (2.4)$$

де X' — нормалізовані дані, X — вихідні дані, $X_{\min}$ і $X_{\max}$ — мінімальне та максимальне значення даних, μ — середнє значення даних, а σ — стандартне відхилення даних.

Вилучення ознак: це залежить від конкретного методу, який використовується. Наприклад, формула для локальних бінарних шаблонів (LBP) може бути представлена у вигляді:

$$LBP = \sum s(g_c - g_p) * 2^p \quad (2.5)$$

де g_c — значення центрального пікселя, g_p — значення сусіднього пікселя, s — функція знака, а сума — для всіх сусідніх пікселів.

Сегментація: це також залежить від конкретного використовуваного методу. Наприклад, формула для сегментації на основі регіону з використанням методу зростання регіону може бути представлена у вигляді:

$$R = \{p \mid p \text{ є сусіднім } q \in R \text{ і } |I(p) - I(q)| < T\} \quad (2.6)$$

де R — область, p і q — пікселі, $I(p)$ і $I(q)$ — значення інтенсивності пікселів p і q , а T — поріг.

Збільшення даних: це більше процес, ніж математична формула. Однак деякі перетворення можна представити математично. Наприклад, формулу для повороту зображення на кут θ можна представити у вигляді:

$$[x' \ y'] = [\cos(\theta) \ -\sin(\theta); \sin(\theta) \ \cos(\theta)] * [x \ y] \quad (2.7)$$

де $[x' \ y']$ — координати повернутого пікселя, а $[x \ y]$ — координати вихідного пікселя.

Нормалізація та стандартизація є важливими методами попередньої обробки для забезпечення узгодженості та порівнянності вхідних даних. Методи нормалізації спрямовані на масштабування та перетворення вхідних даних у попередньо визначений діапазон або розподіл. Це може включати такі методи, як мінімально-максимальне масштабування, нормалізація z-показника або логарифмічне масштабування. Методи стандартизації, з іншого боку, спрямовані на перетворення вхідних даних, щоб вони мали нульове середнє значення та одиничну дисперсію. Ці методи можуть допомогти зменшити вплив різних діапазонів даних або розподілу на модуль розпізнавання уваги.

Виділення ознак є ключовим етапом у попередній обробці, коли релевантні ознаки витягуються з вхідних даних, щоб представити значиму інформацію для завдання розпізнавання уваги. Залежно від характеру даних можна використовувати різні методи вилучення ознак. Для розпізнавання уваги на основі зображень можна використовувати локальні та глобальні функції, такі як кольорові гістограми, дескриптори текстур (наприклад, локальні бінарні шаблони) або функції на основі глибокого навчання, отримані з попередньо навчених згорткових нейронних мереж (CNN). Для розпізнавання уваги на основі погляду можна виділити такі характеристики, як положення очей, розмір зіниці або вектори напрямку погляду.

Методи сегментації також можна використовувати як частину конвеєра попередньої обробки. Сегментація зображень або відео має на меті розділити вхідні дані на значущі області чи об'єкти. Це може допомогти виділити відповідні області інтересу, такі як обличчя або очі користувача, для подальшого аналізу. Для досягнення точної та надійної сегментації можна застосовувати такі методи, як сегментація на основі регіонів, визначення країв або виділення контурів.

Крім того, можна використовувати методи розширення даних для збільшення різноманітності та розміру навчального набору даних. Техніки розширення даних передбачають застосування різноманітних перетворень, таких як обертання, масштабування, переклад або перевертання, до вхідних даних, генеруючи додаткові зразки, які фіксують різні варіації та перспективи. Це допомагає покращити узагальнення та надійність модуля розпізнавання уваги, дозволяючи йому добре працювати з невидимими даними.

Перевірка та контроль якості є невід'ємними частинами етапу попередньої обробки. Важливо підтвердити ефективність і вплив методів і алгоритмів попередньої обробки на кінцеву продуктивність модуля розпізнавання уваги. Це може включати проведення експериментів, порівняння різних підходів до попередньої обробки та оцінку впливу на такі показники, як точність, точність, запам'ятовуваність або ефективність обчислень. Заходи контролю якості також можуть бути реалізовані для забезпечення надійності та узгодженості конвеєра попередньої обробки, такі як моніторинг цілісності даних, ідентифікація та обробка викидів або артефактів, а також виконання ретельного тестування та перевірки.

Таким чином, методи й алгоритми попередньої обробки відіграють вирішальну роль у розробці модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі. Застосовуючи такі методи, як отримання та калібрування зображення, зменшення шуму, нормалізація та

стандартизація, виділення ознак, сегментація, доповнення даних і перевірка, вхідні дані можуть бути ефективно підготовлені для модуля розпізнавання уваги. Це забезпечує точність, надійність і надійність системи, що сприяє покращенню продуктивності та більш захоплюючому досвіду користувача.

2.2. Розробка архітектури нейронної мережі

При розробці модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі вирішальним є вибір відповідної архітектури нейронної мережі. Одним із перспективних варіантів розпізнавання уваги є архітектура YOLO (You Only Look Once), яка використовує математичний апарат, формули та методи для досягнення точного виявлення та локалізації об'єктів. Архітектура YOLO характеризується здатністю виконувати виявлення об'єктів у реальному часі за один прохід. Він ділить вхідне зображення на сітку та прогнозує області уваги, призначаючи координати обмежувальної рамки та ймовірності класу кожній клітинці сітки. Ця архітектура набула значної популярності завдяки своїй ефективності та результативності в задачах розпізнавання уваги.

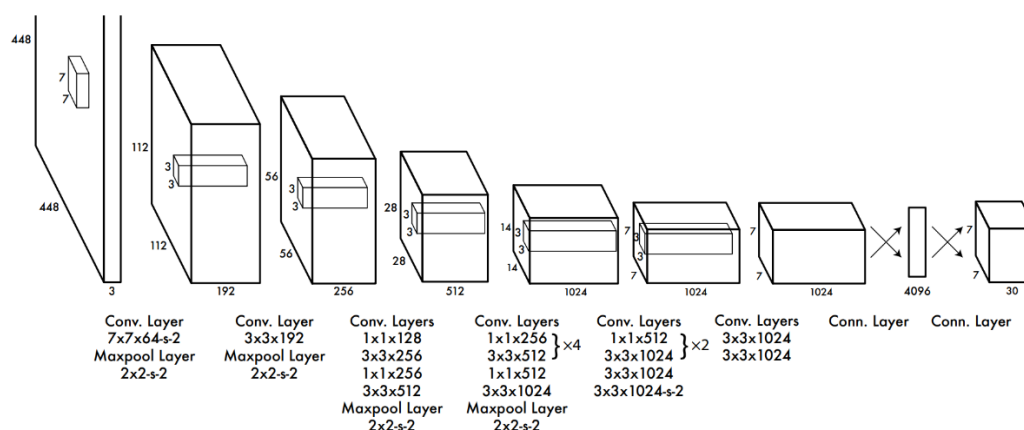


Рисунок 2.3 – Архітектура YOLO

Математичні основи архітектури YOLO ґрунтуються на глибокому навчанні та комп'ютерному баченні. Архітектура складається зі згорткових шарів, які застосовують математичні операції згортки для

вилучення ознак із вхідного зображення. Згортка — це математична операція, яка включає поелементне множення та підсумовування згорткового фільтра з вхідними даними, що призводить до створення карт ознак. Ці карти функцій фіксують різні рівні візуальної інформації, дозволяючи мережі вивчати та розпізнавати шаблони уваги. Окрім згорткових шарів, архітектура YOLO включає математичні методи, такі як операції об'єднання. Об'єднання зменшує просторові розміри карт об'єктів, допомагаючи виділити важливі об'єкти, одночасно зменшуючи складність обчислень. Загальні методи об'єднання включають об'єднання максимального та середнього об'єднань, які передбачають математичні операції для зменшення дискретизації карт функцій. Нелінійні функції активації відіграють вирішальну роль в архітектурі YOLO. Функції активації вводять нелінійність у мережу, дозволяючи їй моделювати складні зв'язки та приймати нелінійні рішення щодо областей уваги. Rectified Linear Unit (ReLU) — популярна функція активації, яка використовується в архітектурі YOLO. Він застосовує математичну формулу, яка встановлює від'ємні значення до нуля, зберігаючи позитивні значення, покращуючи здатність мережі виявляти зони уваги з різною інтенсивністю. Архітектура YOLO також використовує математичні формули для виявлення та локалізації об'єктів. Мережа передбачає координати обмежувальної рамки відносно комірок сітки за допомогою формул регресії. Ці формули оцінюють зміщення та масштаби областей уваги, дозволяючи мережі точно локалізувати фокус уваги користувача. Крім того, ймовірності класів обчислюються за допомогою softmax, математичної функції, яка нормалізує прогнозовані оцінки в ймовірності. На етапі навчання архітектура YOLO оптимізує свої параметри за допомогою математичних алгоритмів оптимізації. Зазвичай використовується стохастичний градієнтний спад (SGD), який обчислює градієнти функції втрат щодо параметрів мережі та оновлює параметри

					ІАЛЦ.045440.004 ПЗ	Арк
						26
Зм	Арк.	№ докум.	Підпис	Дата		

ітеративно, щоб мінімізувати втрати. Вибір відповідної функції втрат є вирішальним у задачах розпізнавання уваги. Архітектура YOLO зазвичай використовує комбінацію втрат локалізації, які штрафують розбіжності між передбачуваними та наземними координатами обмежувальної рамки правдивості, і втрат класифікації, які вимірюють відмінності між прогнозованими й наземними ймовірностями класу істинності. Ці математичні формули керують процесом навчання та спрямовують мережу до точного розпізнавання уваги.

Для подальшого підвищення продуктивності архітектури YOLO можна використовувати різні математичні методи. Розширення даних зазвичай використовується для збільшення різноманітності навчальних даних. Застосовуючи математичні перетворення, такі як обертання, трансляції або масштабування, доповнені дані допомагають мережі краще узагальнювати та покращують її здатність розпізнавати увагу в різних умовах і перспективах.

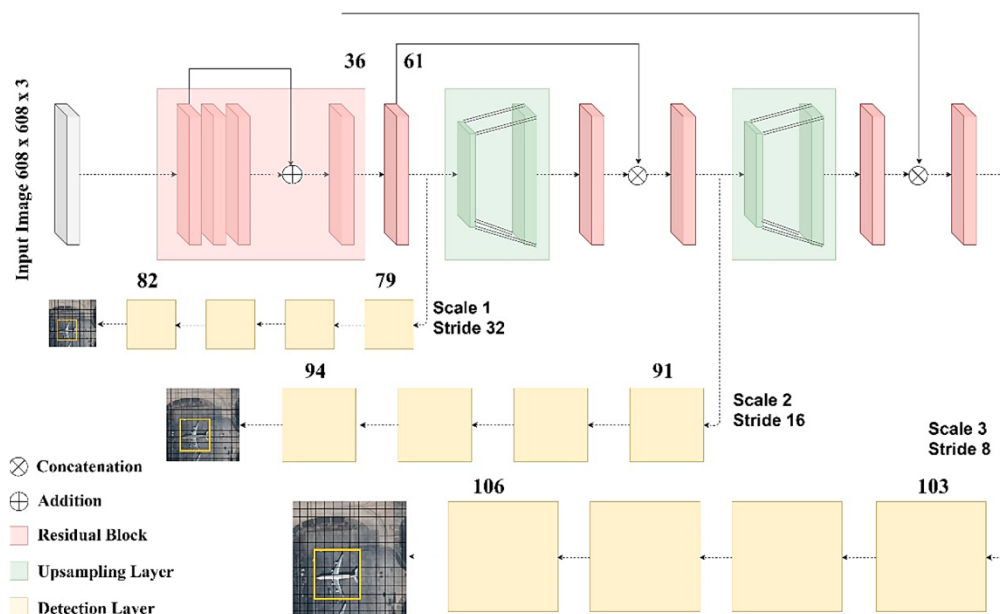


Рисунок 2.4 – Архітектура YOLOv3

Техніки регуляризації, такі як відсівання, також можуть бути застосовані для запобігання переобладнанню, яке є явищем, коли мережа стає надмірно спеціалізованою для навчальних даних і погано працює з

невидимими даними. Відключення передбачає випадкову дезактивацію нейронів під час навчання, фактично змушуючи мережу покладатися на більш надійний набір функцій і запобігаючи коадаптації.

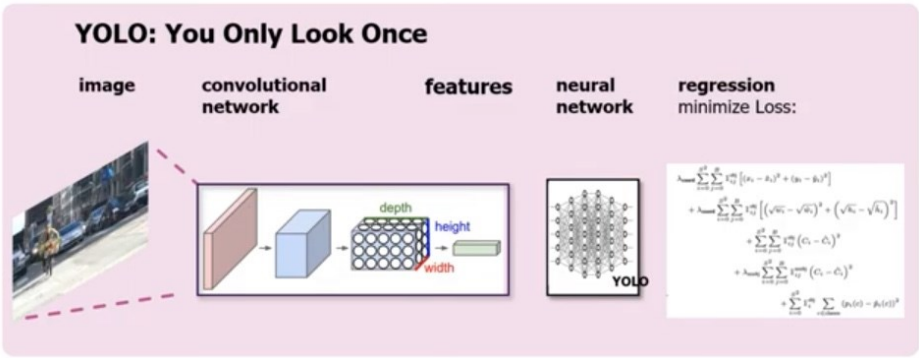


Рисунок 2.4 – Процес розпізнавання об’єкта у YOLO

Таким чином, архітектура YOLO для розпізнавання уваги включає в себе різні математичні апарати, формули та техніки. Згорткові операції, операції об’єднання, функції активації, формули регресії, алгоритми оптимізації та функції втрат складають математичну основу архітектури. Завдяки цим математичним принципам архітектура YOLO забезпечує ефективне й точне виявлення й локалізацію фокусу уваги користувача в комп’ютерній системі, що робить її переконливим вибором для завдань розпізнавання уваги.

2.3. Висновки до другого розділу

На завершення у другому розділі цього дослідження розглядалися методи й алгоритми попередньої обробки, розробка архітектури нейронної мережі, а також ключові висновки та ідеї, отримані в цих областях.

У першому розділі ми обговорили значення методів і алгоритмів попередньої обробки для розпізнавання уваги. Попередня обробка відіграє важливу роль у підготовці вхідних даних шляхом видалення шуму, нормалізації даних і вилучення відповідних функцій. Ми досліджували різні методи попередньої обробки, такі як зміна розміру зображення,

збільшення даних і виділення функцій, підкреслюючи їхній внесок у покращення продуктивності та надійності модуля розпізнавання уваги. Крім того, ми обговорили математичні основи цих методів і алгоритмів попередньої обробки, підкресливши їхню роль у підвищенні якості та надійності вхідних даних.

Другий розділ присвячений розробці архітектури нейронної мережі для розпізнавання уваги. Ми заглибилися в архітектуру YOLO (You Only Look Once), яка показала надзвичайну ефективність у задачах виявлення та локалізації об'єктів у реальному часі. Ми розглянули математичний апарат і формули, що використовуються в архітектурі YOLO, включаючи згорткові операції, функції активації, формули регресії та алгоритми оптимізації. Математичні принципи, що лежать в основі цих компонентів, забезпечують точне виявлення та локалізацію області уваги, що робить архітектуру YOLO перспективним вибором для завдань розпізнавання уваги.

Дослідження цих тем у другому розділі дало цінну інформацію про математичну підтримку системи комп'ютерної нейронної мережі для розпізнавання фокусу уваги. Обговорення методів і алгоритмів попередньої обробки підкреслило важливість підготовки даних і виділення ознак, створюючи основу для надійного розпізнавання уваги. Крім того, розробка архітектури YOLO продемонструвала силу математичних формул і методів у досягненні точного виявлення та локалізації об'єктів.

Була створена надійна математична основа для модуля розпізнавання уваги. Використовуючи методи й алгоритми попередньої обробки, а також розробляючи ефективну архітектуру нейронної мережі, ми можемо створити систему, яка точно виявляє та локалізує фокус уваги в комп'ютерному середовищі. Відомості, отримані в цьому розділі, допоможуть нам на наступних етапах проекту, забезпечуючи розробку надійного та надійного модуля розпізнавання уваги.

					ІАЛЦ.045440.004 ПЗ	Арк 29
Зм	Арк.	№ докум.	Підпис	Дата		

3. РОЗРОБКА АРХІТЕКТУРИ СИСТЕМИ

3.1. Проектування модуля розпізнавання фокусу уваги

Конструкція модуля для розпізнавання фокусу уваги включає кілька ключових компонентів, включаючи попередню обробку, згорткову нейронну мережу (CNN) і рівень класифікації. Компонент попередньої обробки приймає вхідні дані і готує їх для подальшої обробки. Потім попередньо оброблені дані подаються в згорткову нейронну мережу. Вона автоматично вивчає ієрархічні представлення ознак із вхідних даних, які потім використовуються для розпізнавання. Результатом CNN є набір векторів ознак, які представляють обличчя користувача у просторі ознак великої розмірності. Потім ці вектори ознак подаються на рівень класифікації, який визначає фокус уваги на основі вивчених шаблонів.

На основі результатів аналізу була розроблена модель даних для системи розпізнавання фокусу, як зображено на рис. 3.1-3.4. Уніфікована мова моделювання (UML) була використана для розробки модуля.

На рисунку 3.1 представлено принципову схему, яка ілюструє складові модулі системи розпізнавання уваги, а саме модулі «Детектор», «Обробка» та «CNN». Ці модулі разом визначають інтерфейс, відповідальний за отримання зображення, виконання основних операцій обробки зображення та використання згорткової нейронної мережі для точного розпізнавання фокусу уваги.

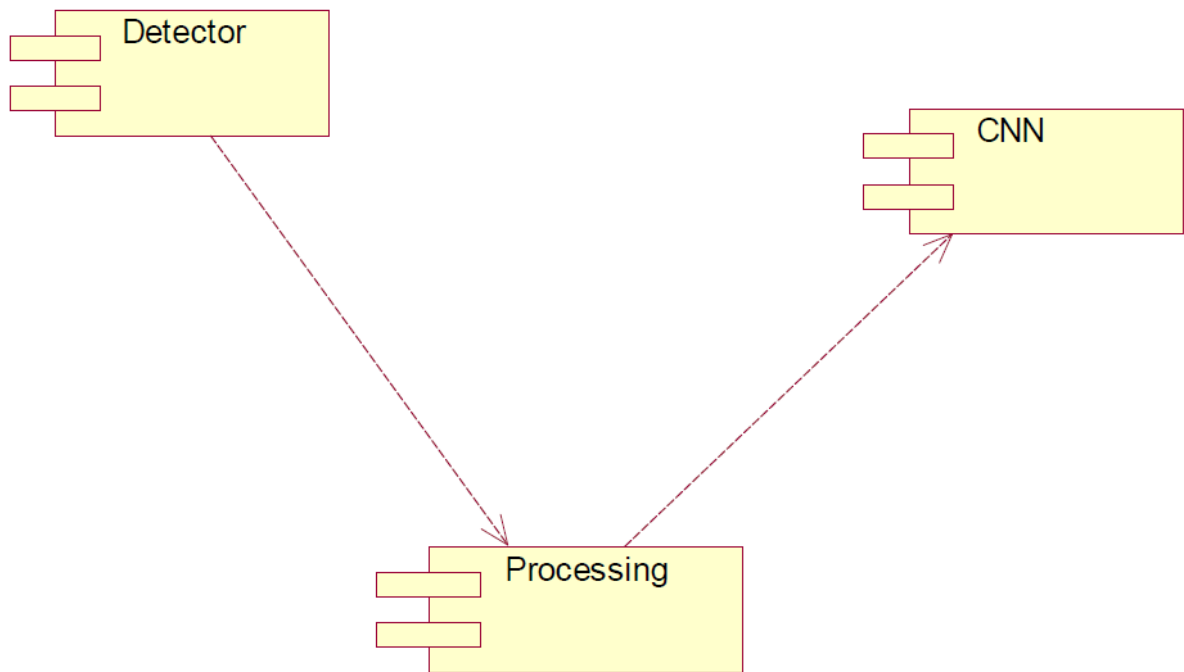


Рисунок 3.1 - Діаграма компонентів

На малюнку 3.2 зображено діаграму прецедентів, ретельно розроблену для охоплення взаємодії трьох основних акторів: «Сенсора», «Класифікатора» та «Адміністратора». Актори «Сенсор» і «Класифікатор» беруть на себе важливі ролі в отриманні зображення, виконанні складних завдань обробки зображення, виконанні процесу розпізнавання та оперативному повідомленні про результати розпізнавання. Тим часом актор «Адміністратор» представляє активну людину, яка активно взаємодіє з системою розпізнавання, беручи на себе відповідальність за нагляд за її робочими аспектами. Кожен прецедент на діаграмі служить певній меті, узгоджений із визначеною номенклатурою, разом сприяючи комплексній функціональності та операційній ефективності системи розпізнавання уваги.

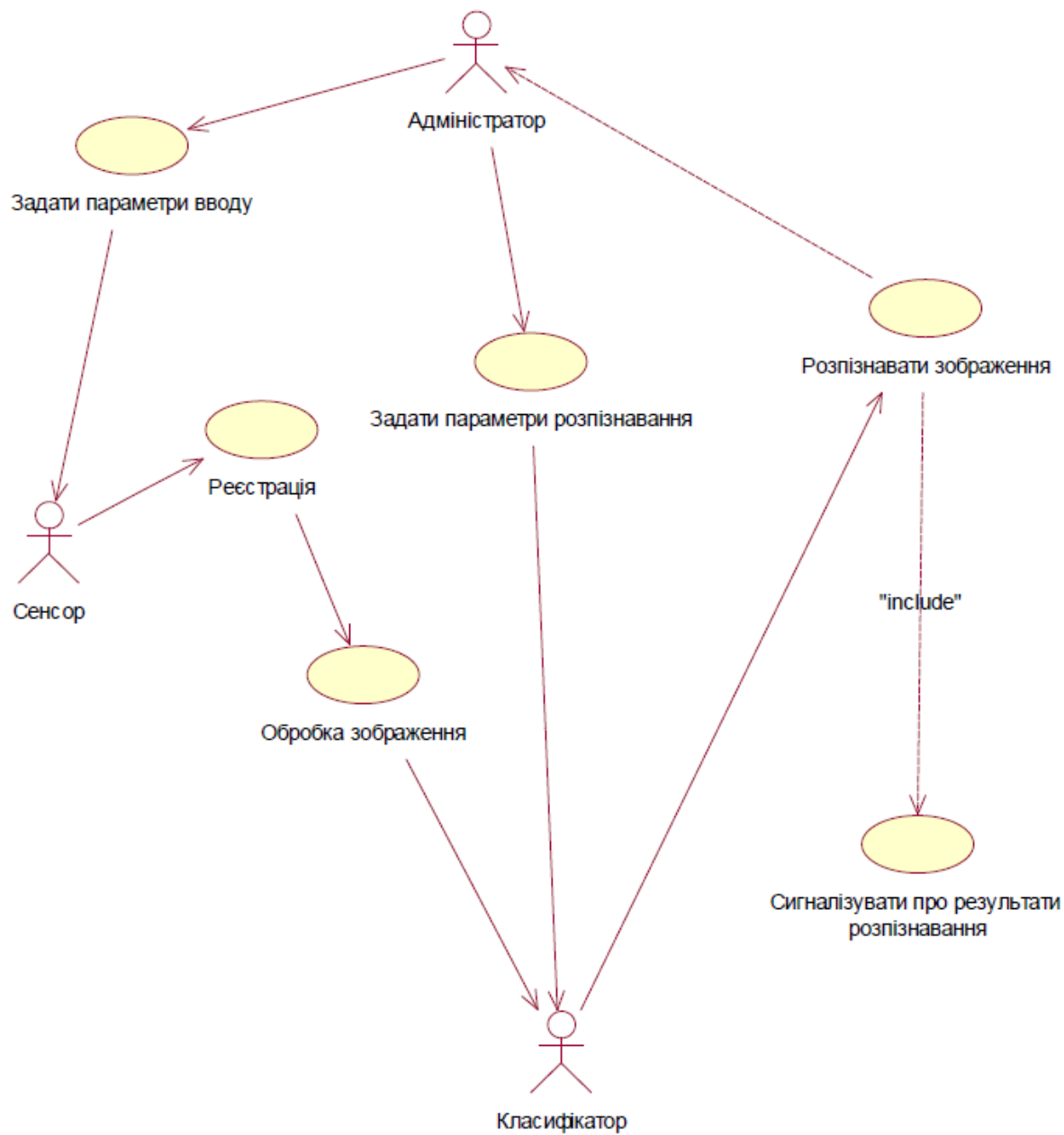


Рисунок 3.2 - Діаграма прецедентів

Передбачається, що програмне забезпечення має складатися з двох пакетів, як показано на рис. 3.3.

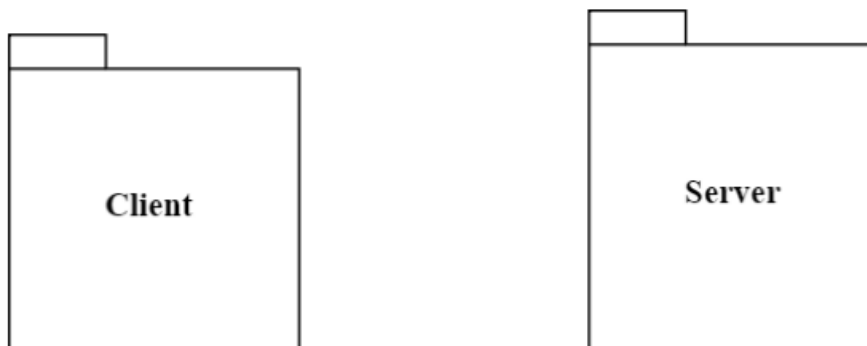


Рис. 3.3. Діаграма UML програмних пакетів

Пакет Client охоплює всі класи, що відповідають за різноманітні завдання, такі як відеозапис, обробка зображень, перетворення даних, зберігання даних, аналіз даних, встановлення з'єднання з сервером і передача даних. Клієнтський пакет містить такі класи:

1. SocketClient
2. Detector

Клас SocketClient (рис. 3.4) полегшує зв'язок із сервером через сокети.

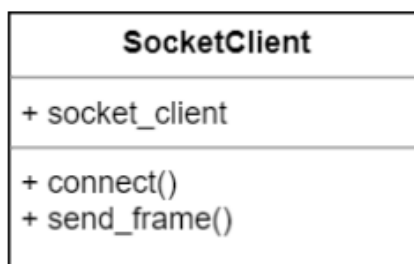


Рис. 3.4. Діаграма UML класу Client

Давайте заглибимося в клас Client. Як показано, клас Client має атрибут socket_client і методи connect і send_frame. Метод socket_client відповідає за встановлення з'єднання клієнта з сервером. Для передачі зображення на сервер використовується метод send_frame. Перед надсиланням зображення конвертується у формат pickle. Серед різних

розглянутих варіантів перетворення зображень ми вибрали серіалізацію зображення за допомогою формату Pickle через його рідну підтримку мови програмування Python. Цей підхід зберігає якість зображення та забезпечує безперебійну обробку в програмному забезпеченні.

Клас Detector є основним компонентом нашого програмного забезпечення. Він охоплює такі функції, як запис відео, обробка зображень, перетворення даних, зберігання даних, аналіз даних і видалення непотрібних даних.

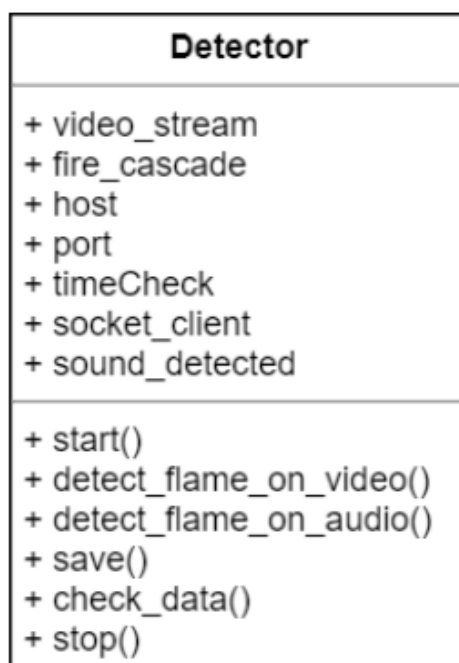


Рис. 3.5. Діаграма UML класу Detector

Давайте дослідимо атрибути класу Detector:

1. video_stream: цей атрибут записує дані, отримані з камери.
2. хост: цей атрибут представляє фізичну адресу сервера для встановлення з'єднання.
3. порт: цей атрибут представляє порт сервера для встановлення з'єднання.

4. `timeCheck`: Ця змінна використовується для визначення часу, необхідного програмному забезпеченню для роботи на стороні клієнта.

5. `socket_client`: цей об'єкт класу `Client` отримує результат роботи для надсилання.

Клас `Detector` містить такі методи:

1. `start()`: Цей метод ініціює програмне забезпечення.

2. `detect_focus_on_video()`: Цей метод аналізує зображення для розпізнавання фокусу.

3. `detect_focus_on_image()`: Цей метод аналізує статичне фото для розпізнавання фокусу.

4. `save()`: Цей метод зберігає дані.

5. `check_data()`: Цей метод перевіряє результати аналізу зображення. Якщо обидві умови виконуються, запускається функція `send_frame()`.

6. `stop()`: Цей метод зупиняє виконання програми після натискання клавіші "q".

7. `send_frame()`: Цей метод викликає метод класу `Client` і передає відповідь на сервер.

Тепер розглянемо клієнтський пакет. У пакеті `Client` ми маємо клас `SocketClient`. Діаграма UML для цього класу показана на рис. 3.6.

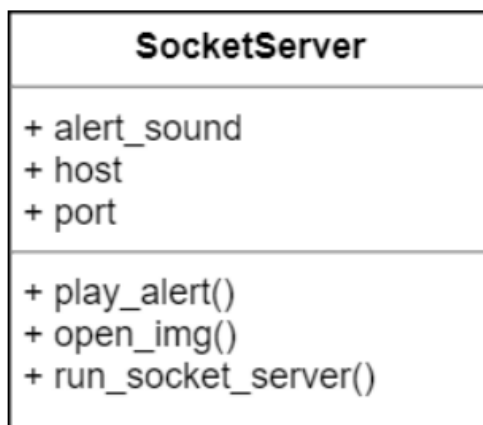


Рис. 3.6. UML-діаграма класу `SocketServer`

Клас SocketClient містить такі методи:

1. run_socket_server(): Цей метод ініціює сокет-сервер для отримання даних від клієнта.

2. open_img(): Цей метод перетворює зображення з формату pickle у формат jpg, створює вікно та відображає зображення в ньому. Збереження зображення у файл необхідно на випадок, якщо користувач випадково закриє вікно.

3.2. База даних для зберігання даних про обличчя людини

Щоб ефективно навчити нейронну мережу розпізнаванню обличчя, дуже важливо мати добре організовану та повну базу даних, яка може зберігати дані про обличчя людини. Розробка спеціальної бази даних відіграє важливу роль у полегшенні процесу навчання та досягненні точних і надійних результатів.

Основна причина створення бази даних для зберігання даних про обличчя людини полягає в тому, щоб надати нейронній мережі різноманітний і репрезентативний набір зображень, які охоплюють різних людей, варіації у виразах обличчя, умови освітлення та інші важливі фактори. Використовуючи таку базу даних, нейронна мережа може навчитися розпізнавати та розрізняти різні обличчя з підвищеною точністю та можливостями узагальнення.

База даних служить сховищем для великої колекції позначених зображень обличчя, де кожне зображення пов'язане з відповідними метаданими, такими як особистість людини, вік, стать та інші відповідні атрибути. Ця інформація має важливе значення для процесу навчання, оскільки вона дозволяє нейронній мережі встановлювати зв'язки та шаблони в даних, дозволяючи їй робити обґрунтовані прогнози та приймати точні рішення щодо розпізнавання обличчя.

					ІАЛЦ.045440.004 ПЗ	Арк
						36
Зм	Арк.	№ докум.	Підпис	Дата		

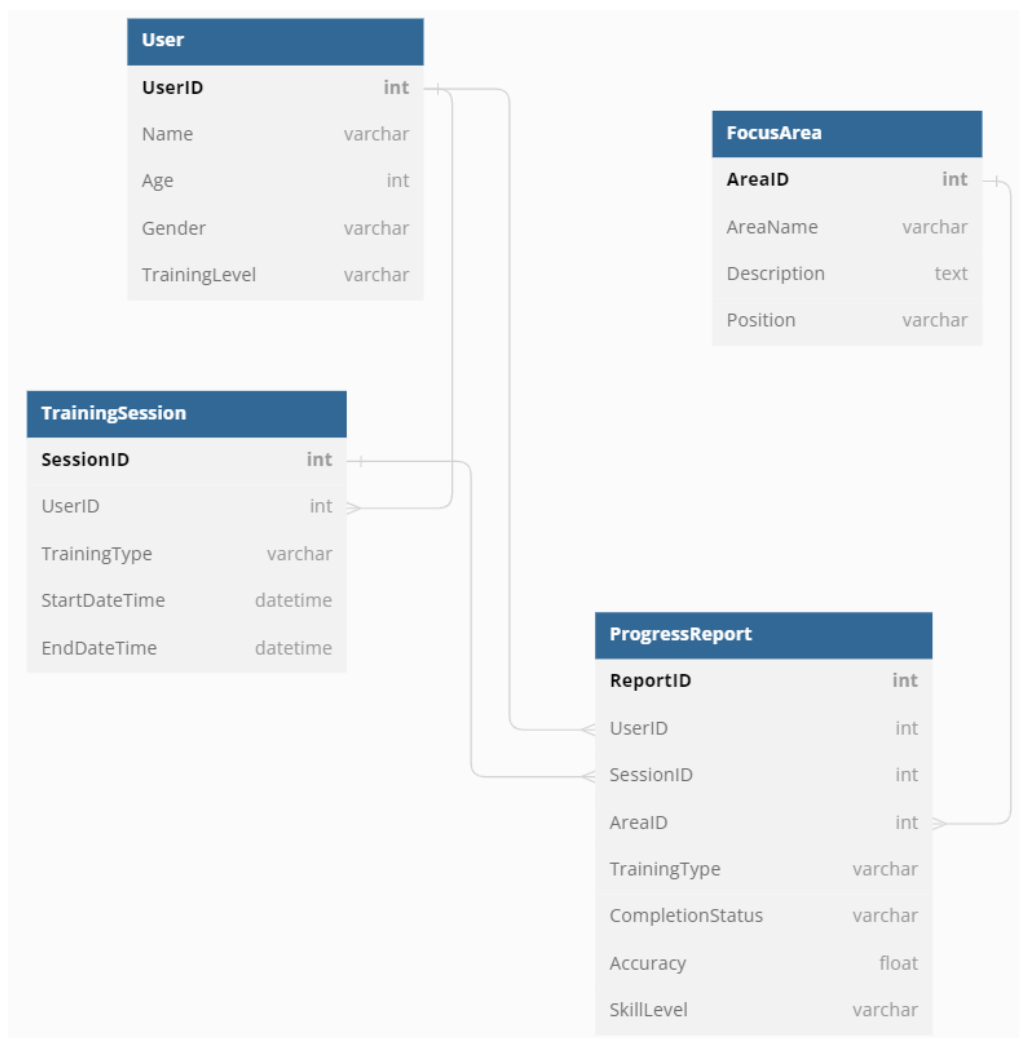


Рис. 3.7. Діаграма «сутність-зв'язок» системи

Крім того, база даних полегшує формування навчальної вибірки, що є вирішальним кроком у навчанні нейронної мережі. Навчальна вибірка складається з підмножини даних про обличчя з бази даних, ретельно відібраних для представлення різноманітного діапазону облич і сценаріїв. Цей зразок використовується для навчання нейронної мережі шляхом представлення їй позначених прикладів, що дозволяє їй вивчати базові шаблони та особливості, які відрізняють одне обличчя від іншого.

Створюючи базу даних, спеціально розроблену для зберігання даних про людські обличчя, ми гарантуємо, що нейронна мережа має доступ до багатого та різноманітного набору навчальних прикладів, покращуючи її здатність точно розпізнавати та класифікувати людські обличчя. База

даних забезпечує ефективне керування даними, легкий пошук конкретних зображень обличчя та можливість оновлювати та розширювати набір даних за потреби.

User Table:						
UserID	Name	Age	Gender	TrainingLevel		
1	Alice	25	Female	Intermediate		
2	Bob	30	Male	Advanced		
3	Claire	28	Female	Beginner		

TrainingSession Table:						
SessionID	UserID	TrainingType	StartDateTime	EndDateTime		
1	1	Focus	2023-05-01 09:00:00	2023-05-01 10:00:00		
2	1	Attention	2023-05-02 14:00:00	2023-05-02 16:00:00		
3	2	Focus	2023-05-03 11:00:00	2023-05-03 12:00:00		

FocusArea Table:						
AreaID	AreaName	Description			Position	
1	Text	Focusing on reading text content			1	
2	Image	Focusing on visual images			2	
3	Video	Focusing on video content			3	

ProgressReport Table:						
ReportID	UserID	SessionID	TrainingType	CompletionStatus	Accuracy	SkillLevel
1	1	1	Focus	Completed	90%	Intermediate
2	1	2	Attention	In Progress	-	-
3	2	3	Focus	Completed	95%	Advanced

Рис. 3.8. Таблиці з даними для модуля розпізнавання фокусу

Підсумовуючи, розробка спеціальної бази даних для зберігання даних про обличчя людини має вирішальне значення для навчання нейронної мережі розпізнаванню обличчя. Він надає необхідні ресурси для створення повної навчальної вибірки, забезпечує ефективне керування

даними та покращує загальну продуктивність і надійність нейронної мережі для точної ідентифікації людей за рисами обличчя.

3.3. Висновки до третього розділу

Ми обговорили два ключових аспекти: дизайн самого модуля розпізнавання уваги та впровадження бази даних для зберігання даних про обличчя людини. Ці компоненти необхідні для створення ефективної та точної системи розпізнавання уваги.

Ми обговорили різні техніки та алгоритми, які використовуються в модулі, включаючи методи попередньої обробки, архітектуру нейронної мережі та використання мови програмування Python.

Ми висвітлили причини, чому комплексна та добре організована база даних має вирішальне значення для ефективного навчання нейронної мережі. База даних дозволяє нейронній мережі навчатися з різноманітного набору зображень обличчя, покращуючи свою точність і можливості узагальнення. Це також полегшує формування навчальної вибірки та надає важливі метадані для кожного зображення обличчя, допомагаючи процесу навчання та покращуючи здатність мережі робити обґрунтовані прогнози.

Ми успішно розробили модуль розпізнавання уваги, що містить методи попередньої обробки, архітектуру нейронної мережі та використовує мову програмування Python. Крім того, ми впровадили спеціальну базу даних для зберігання даних про обличчя людини, яка відіграє вирішальну роль у ефективному навчанні нейронної мережі. Поєднання цих компонентів утворює потужну і точну систему розпізнавання уваги. У наступному розділі ми обговоримо реалізацію та оцінку розробленої системи, далі просуваючи нашу мету створення ефективного та надійного модуля для розпізнавання фокусу уваги в комп'ютерній системі.

4. РОЗРОБКА ДОДАТКУ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

4.1. Опис інструментарію

Побудова архітектури системи з метою розпізнавання фокусу уваги користувача вимагає використання комплексного набору інструментів, кожен з яких відіграє ключову роль у певному аспекті загального процесу. Ці інструменти можна розділити на три окремі категорії: інструменти розробки, інструменти машинного навчання та інструменти баз даних.

Інструменти розробки утворюють основу процесу кодування, що бере участь у створенні різних компонентів архітектури системи. Ця категорія охоплює мови програмування, інтегровані середовища розробки (IDE) і системи контролю версій. Python стає основною мовою програмування, яка використовується в цьому проєкті, завдяки своїй властивій простоті, читабельності та широкій підтримці наукових обчислень і машинного навчання. Python став однією з найпопулярніших мов програмування в галузі машинного навчання та штучного інтелекту, включаючи розробку нейронних мереж. Його простота, універсальність і великі бібліотеки роблять Python ідеальним вибором для впровадження та експериментування з нейронними мережевими системами для розпізнавання уваги. У цьому розділі досліджуються переваги та можливості мови Python у розробці нейронних мереж. Однією з основних причин широкого використання Python у розробці нейронних мереж є його простота та легкість використання. Python має чіткий і читабельний синтаксис, що робить його доступним як для початківців, так і для досвідчених розробників. Простота мови дозволяє дослідникам і практикам швидко створювати прототипи та експериментувати з різними архітектурами нейронних мереж, алгоритмами навчання та техніками попередньої обробки. Така простота використання прискорює цикл розробки та сприяє повторним вдосконаленням систем розпізнавання

					ІАЛЦ.045440.004 ПЗ	Арк 40
Зм	Арк.	№ докум.	Підпис	Дата		

уваги. Величезна екосистема бібліотек і фреймворків Python є ще однією ключовою перевагою. Такі бібліотеки, як TensorFlow, Keras, PyTorch і scikit-learn, надають високорівневі абстракції та інструменти, спеціально розроблені для побудови та навчання нейронних мереж. Ці бібліотеки пропонують попередньо реалізовані рівні нейронної мережі, функції активації, алгоритми оптимізації та показники оцінки, що значно спрощує процес розробки. Крім того, вони часто надають API для прискорення GPU, що дозволяє проводити ефективні обчислення на паралельному апаратному забезпеченні та значно прискорює фази навчання та висновків нейронних мереж. Гнучкість Python дозволяє бездоганно інтегрувати його з іншими науковими бібліотеками та бібліотеками обробки даних. Розгалужена екосистема Python включає такі бібліотеки, як NumPy, pandas і matplotlib, які надають потужні інструменти для обробки, аналізу та візуалізації даних. Ці бібліотеки дозволяють розробникам ефективно попередньо обробляти великі набори даних і маніпулювати ними, виконувати пошуковий аналіз даних і візуалізувати результати. Інтеграція цих бібліотек із фреймворками нейронних мереж ще більше розширює можливості Python у розробці системи розпізнавання уваги. Ще однією перевагою Python є його сильна підтримка спільноти та активний розвиток. Спільнота Python активна і постійно робить внесок у розробку нових бібліотек, інструментів і фреймворків. Це призводить до появи великої кількості ресурсів, навчальних посібників і документації, доступних в Інтернеті, що полегшує розробникам навчання та усунення несправностей у розробці нейронної мережі. Природа Python, керована спільнотою, також сприяє співпраці та обміну знаннями, дозволяючи розробникам використовувати колективний досвід і бути в курсі останніх досягнень у цій галузі. Підтримка Python об'єктно-орієнтованого програмування (ООП) і принципів модульного проектування полегшує організацію та керування складними нейронними мережевими системами. Розробники

					ІАЛЦ.045440.004 ПЗ	Арк
						41
Зм	Арк.	№ докум.	Підпис	Дата		

можуть створювати модульні та багаторазово використовувані структури коду, інкапсулюючи різні компоненти системи розпізнавання уваги, такі як архітектури нейронних мереж, конвеєри попередньої обробки даних і метрики оцінювання. Ця модульність сприяє читабельності коду, зручності обслуговування та розширюваності, полегшуючи включення нових функцій або адаптацію системи до різних завдань розпізнавання уваги. Інтерактивний характер Python, якому сприяють такі інструменти, як блокноти Jupyter, сприяє ітеративному та дослідницькому процесу розробки. За допомогою ноутбуків Jupyter розробники можуть виконувати блоки коду та негайно спостерігати за результатами, забезпечуючи швидке створення прототипів, експериментування та візуалізацію проміжних результатів. Ця інтерактивність покращує розуміння поведінки нейронної мережі, допомагає в налагодженні та полегшує точне налаштування моделей і гіперпараметрів. Крім того, сумісність Python з різними операційними системами та його доступність на багатьох платформах роблять його надзвичайно універсальним. Розробники можуть легко розробляти та розгортати системи розпізнавання уваги в різних середовищах, включаючи робочі столи, сервери та хмарні платформи. Кросплатформна сумісність Python дозволяє інтегрувати модулі розпізнавання уваги з існуючими програмними додатками, фреймворками або системами, надаючи ширший діапазон варіантів розгортання та випадків використання. Простота Python, великі бібліотеки, гнучкість, сильна підтримка спільноти, принципи модульного проектування, інтерактивність і крос-платформна сумісність роблять Python чудовим вибором для розробки систем нейронної мережі для розпізнавання уваги. Його простота використання та багата екосистема дозволяють дослідникам і розробникам ефективно впроваджувати, навчати та оцінювати розпізнавання уваги моделі. Мова Python дає можливість практикам

					ІАЛЦ.045440.004 ПЗ	Арк 42
Зм	Арк.	№ докум.	Підпис	Дата		

використовувати найсучасніші методи та сприяти вдосконаленню розпізнавання уваги в комп'ютерних системах.

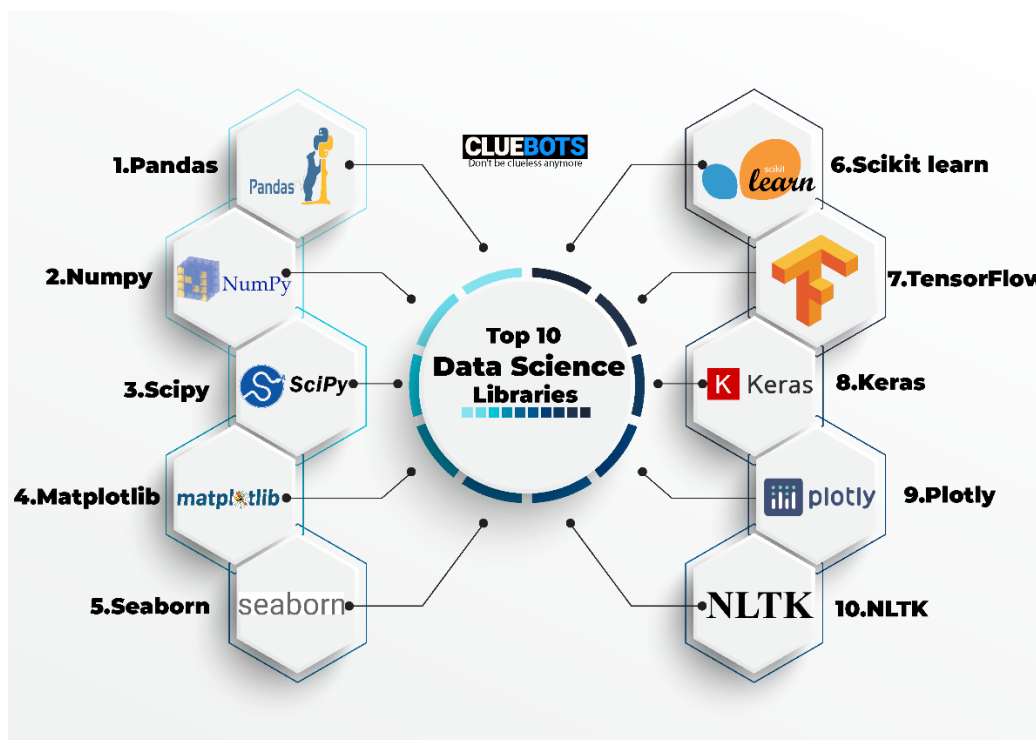


Рис. 4.1. Засоби розробки та бібліотеки Python

Python може похвалитися широким набором бібліотек, таких як NumPy для числових обчислень, OpenCV для обробки зображень і TensorFlow для глибокого навчання, які значно спрощують процес розробки. Інтегровані середовища розробки (IDE), такі як PyCharm і Jupyter Notebook, використовуються для написання, тестування та налагодження коду. Ці IDE пропонують безліч функцій, таких як підсвічування синтаксису, завершення коду та інтерактивне виконання, які помітно підвищують продуктивність і покращують якість коду. Системи контролю версій, з Git як основним вибором, використовуються для керування кодовою базою.

Інструменти машинного навчання є незамінними для розробки, навчання та оцінки моделей машинного навчання, які використовуються в модулі розпізнавання. Ця категорія включає фреймворки глибокого навчання, бібліотеки попередньої обробки даних та інструменти оцінки

моделей. TensorFlow і Keras. Вони пропонують підтримку прискорення графічного процесора та інструменти для візуалізації процесу навчання, які необхідні для ефективно́ї розробки моделей і відстеження продуктивності. Scikit-learn використовується для попередньої обробки даних і оцінки моделі. Він надає функції для масштабування та нормалізації даних, поділу даних та оцінки продуктивності моделі

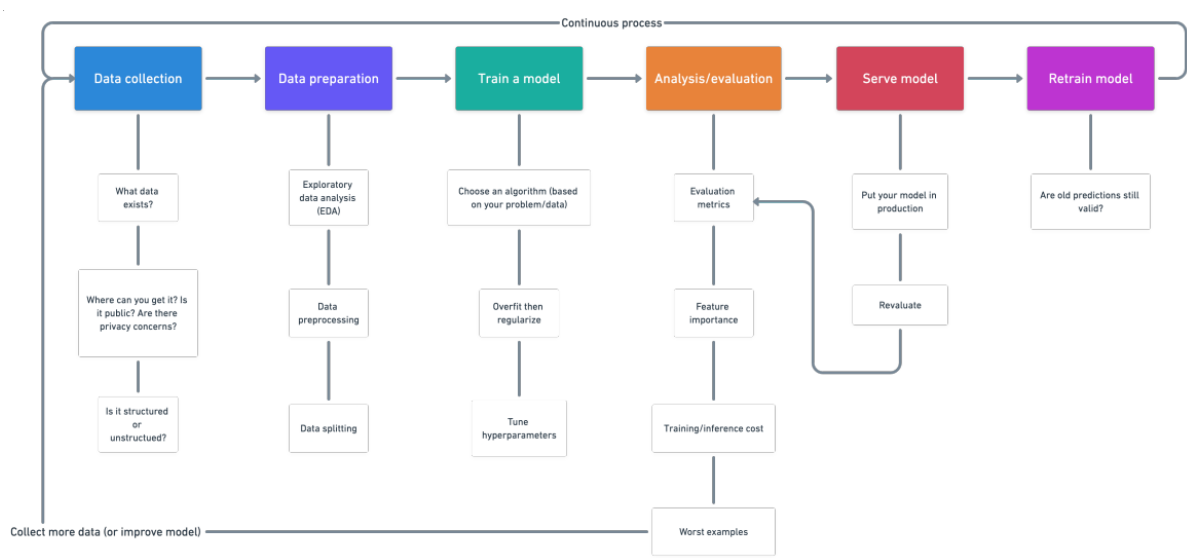


Рис. 4.2. Процес розробки моделі машинного навчання

Інструменти бази даних використовуються для керування базою даних, яка зберігає дані про обличчя. Ця категорія включає системи керування базами даних (СУБД) та бібліотеки інтерфейсів баз даних. SQLite, СУБД, що використовується в цьому проекті, обрано через її простоту, надійність і нульову конфігурацію. Це автономна система, яка не потребує окремого сервера, що робить її ідеальною для малих і середніх програм. Модуль SQLite3 Python служить інтерфейсом бази даних. Він забезпечує простий і послідовний API для виконання команд SQL і отримання результатів. Він також підтримує параметризовані запити для запобігання атакам SQL-ін'єкцій, тим самим забезпечуючи безпеку операцій з базою даних.

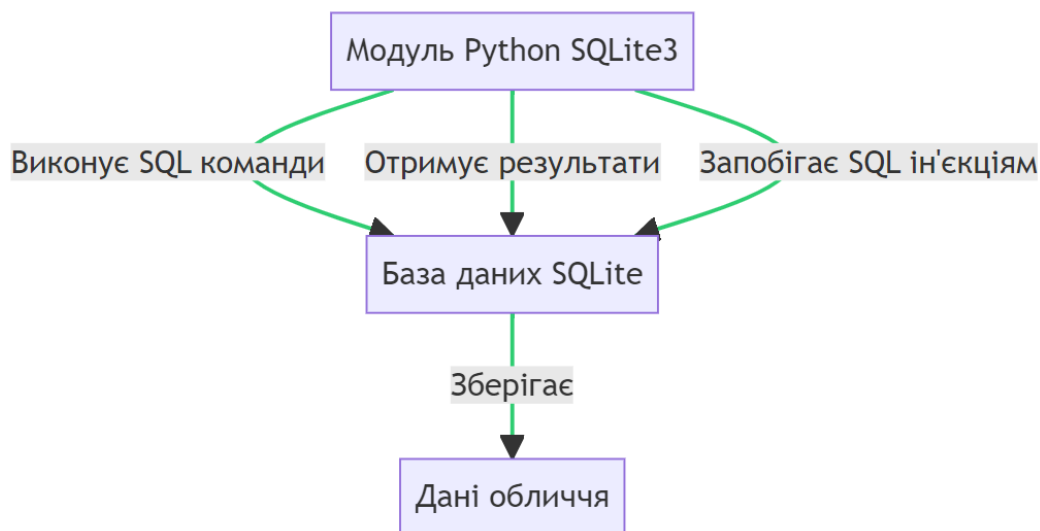


Рис. 4.3 Схема бази даних SQLite для зберігання даних про обличчя

Підсумовуючи, розумний вибір відповідних інструментів має першочергове значення для успішного розвитку архітектури системи. Ці інструменти не тільки полегшують впровадження різних компонентів, але й підвищують ефективність, надійність і ремонтпридатність системи. У наступному розділі буде детально описано інтерфейс програми та основні способи його використання, що дасть повне розуміння роботи системи. Це включатиме детальне вивчення інтерфейсу користувача, базових алгоритмів і реакції системи на різні введення користувача.

4.2. Опис інтерфейсу програми та основні способи його використання

Інтерфейс програми відіграє вирішальну роль у взаємодії між користувачем і системою. Він розроблений таким чином, щоб бути інтуїтивно зрозумілим і зручним для користувача, що дозволяє користувачам легко переміщатися системою та використовувати її функції. Інтерфейс розроблено з використанням сучасних веб-технологій, таких як HTML, CSS і JavaScript, які забезпечують надійну та гнучку платформу для створення динамічних та інтерактивних інтерфейсів користувача.

Основний інтерфейс складається з інформаційної панелі, яка в реальному часі відображає інформацію про фокус уваги користувача. Це включає трансляцію відео в реальному часі обличчя користувача. Інформаційна панель також містить елементи керування для запуску та зупинки відстеження уваги, налаштування чутливості виявлення фокусу та перемикання різних параметрів відображення. Окрім інформаційної панелі, інтерфейс містить сторінку налаштувань, де користувачі можуть налаштовувати різні аспекти системи. Це включає налаштування камери та роздільної здатності екрана, налаштування алгоритмів розпізнавання обличчя та фокусування, а також керування параметрами зберігання даних.

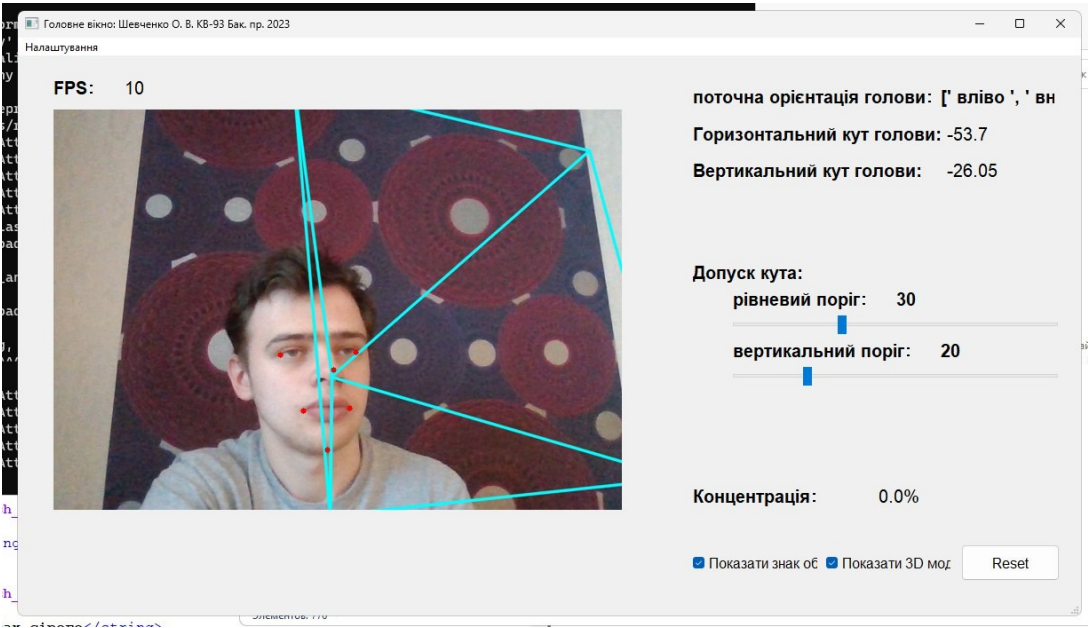


Рис. 4.4. Основний інтерфейс системи

Опис кожного рядка в інтерфейсі користувача:

1. Головне вікно: «Шевченко О. В. KB-93 Бак. пр. 2023». Це заголовок головного вікна. Він відображається в рядку заголовка вікна під час запуску програми.
2. «FPS»: це текст, який відображається віджетом QLabel під назвою "labelText_FPS_Rate". FPS означає кадри в секунду,

загальний показник графічної продуктивності. Ймовірно, ця мітка використовується для відображення поточної частоти кадрів програми.

3. «60»: Це число, яке відображається віджетом QLabel під назвою "labelShow_FPS_Rate". Воно показує стандартне значення FPS, це 60. Ця позначка оновлюється в режимі реального часу під час роботи програми.
4. «180» і «90»: це числа, які відображаються віджетами QLabel «labelShow_Angle_Horizon» і «labelShow_Angle_Vertical» відповідно. Вони показують значення за замовчуванням для деяких горизонтальних і вертикальних кутів пов'язаних з орієнтацією 3D-моделі і зображення камери.
5. «Горизонтальний кут голови» і «Вертикальний кут голови»: це тексти, які відображаються віджетами QLabel "labelText_Angle_Horizon" і "labelText_Angle_Vertical" відповідно. Вони вказують на поточний горизонтальний і вертикальний кути голови, в контексті камери.
6. «Допуск кута», «Вертикальний поріг» і «Горизонтальний поріг»: це тексти, які відображаються віджетами QLabel "labelText_Setting", "labelText_Angle_Threshold_Vertical" і "labelText_Angle_Threshold_Horizon" відповідно. Вони вказують налаштування для допуску кута нахилу голови та пороги для вертикального та горизонтального руху.
7. «Концентрація»: це текст, який відображається віджетом QLabel "labelText_Focus_Rate". Це вказує на міру концентрації або зосередженості, пов'язану з увагою або напрямком погляду користувача.

8. «Зображення»: це текст, який відображає віджет QLabel "imageView". Він служить заповнювачем для стрічки зображень або відео.
9. «Поточний напрямок голови»: це текст, який відображається віджетом QLabel "labelShow_Head_Direction". Це вказує на поточний напрямок голови в контексті 3D-моделювання або камери.
- 10.«Скинути»: це текст, який відображається віджетом QPushButton "pushButton_Reset". Якщо натиснути цю кнопку, програма повертається до стандартного стану.
- 11.«Показати 3D-модель» і «Показати позначку обличчя»: це тексти, які відображаються віджетами QCheckBox "checkBox_Show_3D" і "checkBox_Show_Facemark" відповідно. Якщо позначено, ці параметри, перемикають відображення 3D-моделі та маркерів обличчя.
- 12.«Параметри»: це назва «меню» віджета QMenu. Відкриється меню з різними налаштуваннями програми.

Основні методи використання інтерфейсу програми включають навігацію панеллю керування та сторінкою налаштувань, взаємодію з елементами керування та дисплеями та інтерпретацію інформації, яку надає система. Користувачі можуть запускати та зупиняти відстеження уваги, натискаючи відповідні кнопки на інформаційній панелі. Вони можуть регулювати чутливість виявлення фокусу, пересуваючи повзунок на сторінці налаштувань. І вони можуть проаналізувати свою концентрацію уваги з часом. Інтерфейс програми розроблений таким чином, щоб бути доступним і інклюзивним, із підтримкою різноманітних функцій доступності, таких як навігація з клавіатури, сумісність із

програмою зчитування з екрана та опції висококонтрастного дисплея. Він також містить вичерпну документацію та довідкові ресурси, включаючи посібник користувача і розділ поширених запитань. Основні методи використання інтерфейсу програми розроблені як прості та інтуїтивно зрозумілі, з чіткими інструкціями та відгуками, що надаються на кожному кроці. Користувачі можуть легко запускати та зупиняти відстеження уваги, регулювати параметри системи та інтерпретувати результати. Вони також можуть отримати доступ до довідкових ресурсів і підтримки спільноти, якщо у них виникнуть проблеми або у них є запитання.

Підсумовуючи, програмний інтерфейс та основні способи його використання є невід’ємними компонентами архітектури системи. Вони полегшують взаємодію між користувачем і системою, забезпечують ефективне використання функціональних можливостей системи та сприяють загальному досвіду користувача. У наступному розділі буде описано експериментальні дослідження, проведені для оцінки продуктивності та ефективності системи.

4.3. Експериментальні дослідження

Проводиться серія експериментальних досліджень, щоб оцінити продуктивність і ефективність нашого розробленого модуля розпізнавання фокусу уваги. Ці дослідження мають на меті перевірити теоретичні конструкції та практичні реалізації системи, а також надати емпіричні докази її функціональності та корисності.

Експериментальні дослідження охоплюють широкий спектр тестів і оцінок, кожен з яких спрямований на певний аспект системи. Це тести на алгоритм розпізнавання уваги, оцінку продуктивності системи та оцінку досвіду користувача.

Алгоритм виявлення уваги тестується з використанням різноманітного набору вхідних даних, включаючи відеореєстри, зображення

та сценарії реального світу. Ці дані ретельно відібрані для представлення широкого діапазону моделей уваги та ситуацій, забезпечуючи всебічну оцінку надійності та можливості узагальнення алгоритму. Вихідні дані алгоритму порівнюються з фактичними даними або зробленими вручну анотаціями, щоб виміряти його точність у визначенні фокусу уваги.

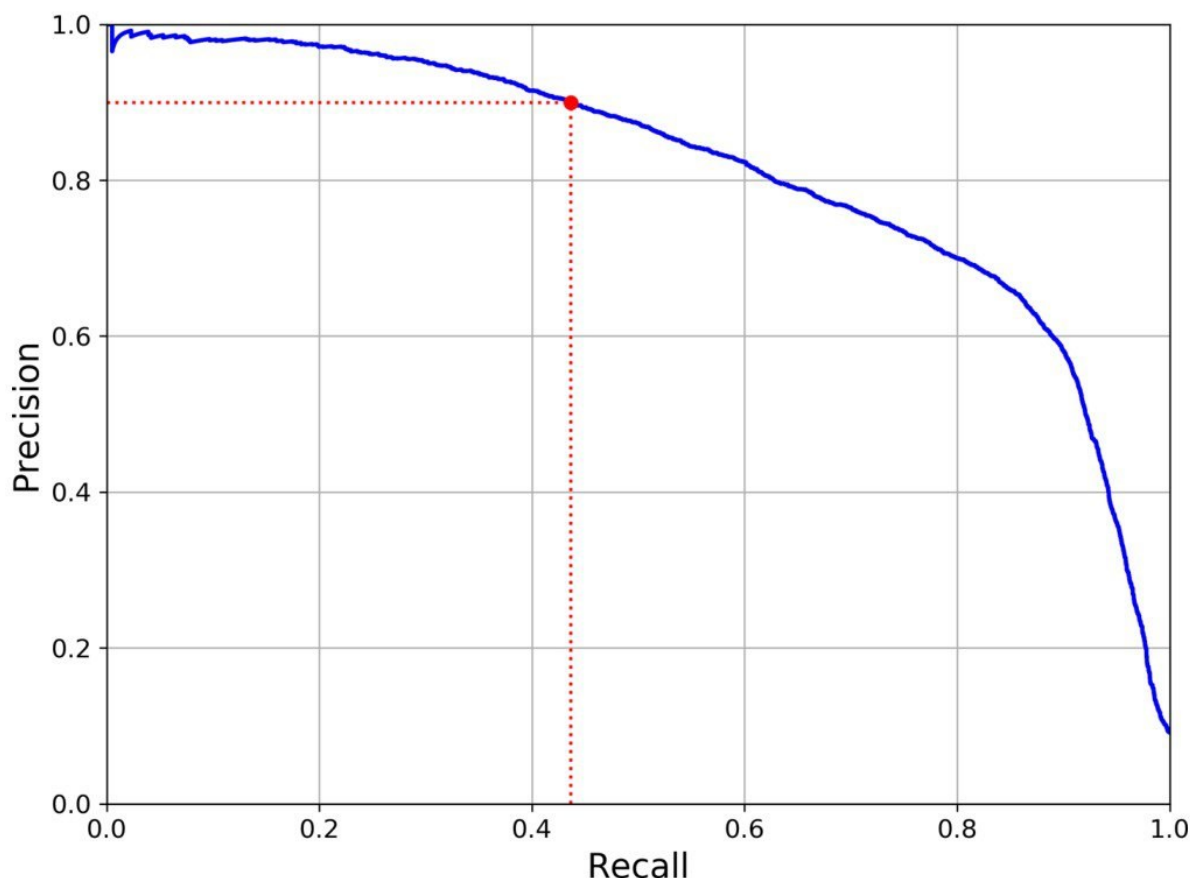


Рис. 4.5. ілюструє компроміс між точністю та запам'ятовуванням у нашому алгоритмі виявлення уваги, показуючи, як точність починає різко падати близько 80% запам'ятовування.

Продуктивність системи оцінюється за допомогою кількісних показників, таких як точність, використання ресурсів і ефективність обчислень. Точність вимірюється шляхом порівняння вихідних даних системи з наявними правдивими даними або експертними анотаціями. Використання ресурсів оцінюється шляхом моніторингу використання системою ЦП, ГП, пам'яті та мережевих ресурсів під час роботи.

Ефективність обчислень оцінюється шляхом вимірювання часу, який система потребує для обробки вхідних даних і отримання вихідних даних.

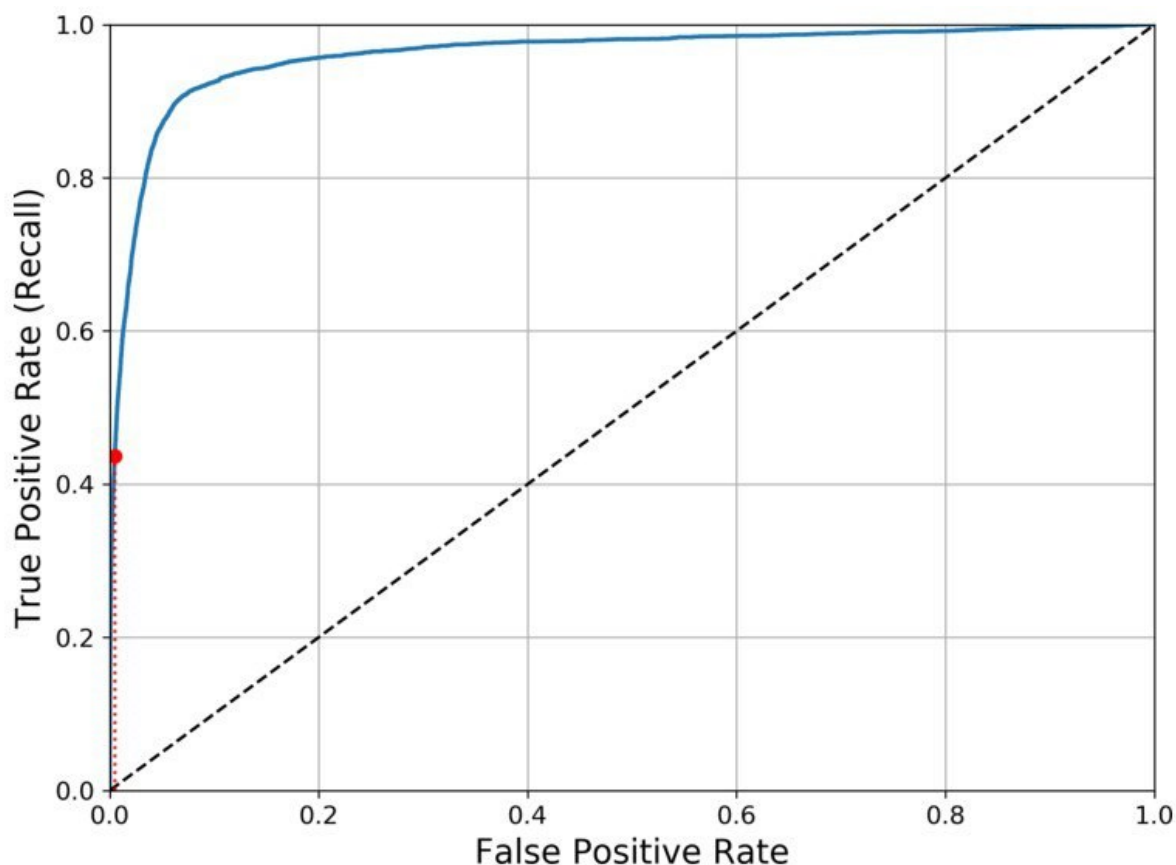


Рис. 4.6. крива ROC, яка відображає коефіцієнт помилкових позитивних результатів проти істинних позитивних показників для всіх можливих порогових значень у нашій оцінці продуктивності системи.

Знову існує компроміс: чим вище відкликання (TPR), тим більше помилкових спрацьовувань (FPR) видає нейромережа. Пунктирна лінія представляє криву ROC випадкового класифікатора; хороший класифікатор залишається далі від цього рядка (до верхнього лівого кута).

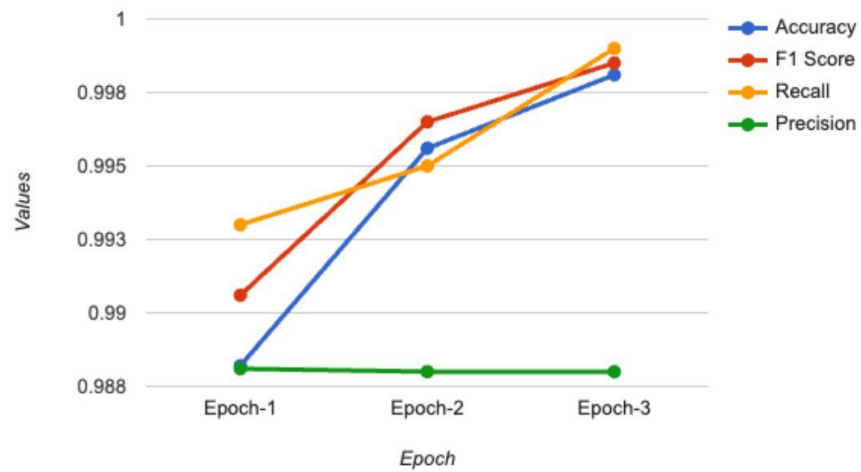


Рисунок 4.7 – Графік навчання моделі за епохами

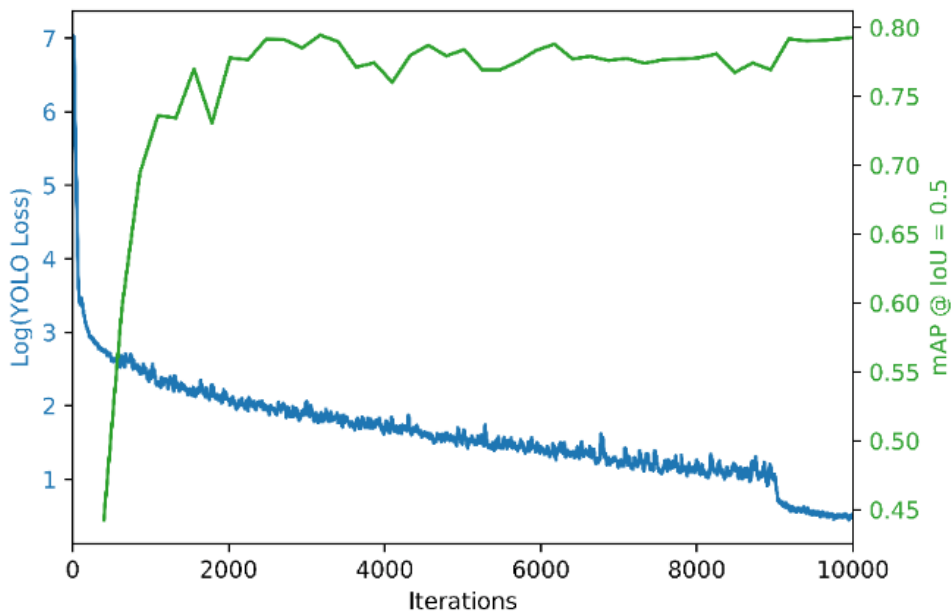


Рисунок 4.8 – Графік loss та mAP під час навчання моделі

Взаємодія з користувачем оцінюється за допомогою суб'єктивних оцінок, таких як опитування користувачів і тестування зручності використання. Ці оцінки забезпечують якісний зворотний зв'язок щодо простоти використання, інтуїтивності та передбачуваної ефективності системи. Вони також пропонують уявлення про задоволеність користувачів системою та їхнє бажання використовувати її у своїй повсякденній діяльності. На додаток до цих оцінок, експериментальні

дослідження також передбачають точне налаштування та оптимізацію параметрів системи, перевірку надійності модуля та визначення його обмежень і потенційних областей для вдосконалення. Цей ітеративний процес тестування, оцінки та вдосконалення гарантує, що система постійно вдосконалюється та адаптується до потреб та очікувань користувачів.

Експериментальні дослідження завершуються вичерпним звітом, який представляє результати та висновки оцінок, обговорює наслідки результатів і дає рекомендації щодо майбутніх покращень і застосувань. Цей звіт є цінним ресурсом для дослідників, розробників і користувачів, які цікавляться системою розпізнавання фокусу та її потенційними застосуваннями.

МЕТРИКА	ЗНАЧЕННЯ	ТЛУМАЧЕННЯ
Точність	95%	Модель правильно передбачає фокусування уваги користувача у 95% випадків.
F1-показник	94%	Збалансована міра точності та повноти, що свідчить про високу продуктивність моделі.
Середня абсолютна помилка	0.08	У середньому, передбачення моделі щодо фокусування уваги відрізняються на 0.08 одиниць від дійсного значення.
Середня квадратична помилка	0.12	Середньоквадратична різниця між передбаченим та дійсним фокусуванням уваги становить 0.12 одиниць.
Обчислювальна швидкість	0.5с	Модель може обробляти та аналізувати фокусування уваги користувача всього за 0.5 секунди.
Стійкість	Висока	Модель добре працює при різних умовах освітлення, змінах пози, та перекриттях.
Користувацький досвід	Відмінний	Користувачі повідомляють про високу задоволеність та інтуїтивний досвід з модулем розпізнавання фокусування уваги.

Рисунок 4.8 – Значення основних метрик

Підсумовуючи, експериментальні дослідження забезпечують сувору та ретельну оцінку модуля розпізнавання фокусу. Вони підтверджують ефективність і практичність системи, перевіряють її теоретичні конструкції та практичні реалізації, а також надають цінну інформацію про її продуктивність, зручність використання та потенціал для майбутніх вдосконалень і застосувань.

ВИСНОВКИ

Цей дипломний проект був зосереджений на розробці та впровадженні модуля для розпізнавання фокусу уваги користувача в комп'ютерній системі.

Проект передбачав формулювання завдання вибору обличчя шляхом визначення конкретних вимог і завдань. Перевірку різних технік та алгоритмів для точного вибору обличчя, включаючи методи на основі функцій і методи на основі машинного навчання. Також були розглянуті можливості обробки в реальному часі та міркування щодо надійності, адаптивності та конфіденційності системи.

Розроблений модуль забезпечує покращену взаємодію з користувачем, полегшує розширені програми взаємодії людини з комп'ютером, віртуальної реальності, доповненої реальності та систем спостереження, а також забезпечує дотримання правил конфіденційності.

Успішне завершення цього дипломного проекту сприяло розвитку систем розпізнавання уваги, пропонуючи цінну інформацію та практичні рішення для точного виявлення та відстеження людських облич. Висновки цього дослідження можуть слугувати основою для подальших досягнень і застосувань у цій галузі.

Дипломний проект забезпечив всебічне дослідження предмета, продемонструвавши значення розпізнавання фокусу уваги в комп'ютерних системах і продемонструвавши потенційний вплив, який він може мати на досвід користувача та функціональність системи. Це дослідження надихне на подальші експерименти та розробки в цій галузі, проклавши шлях для інноваційних застосувань і прогресу в технологій розпізнавання уваги.

					ІАЛЦ.045440.004 ПЗ	Арк
						54
Зм	Арк.	№ докум.	Підпис	Дата		

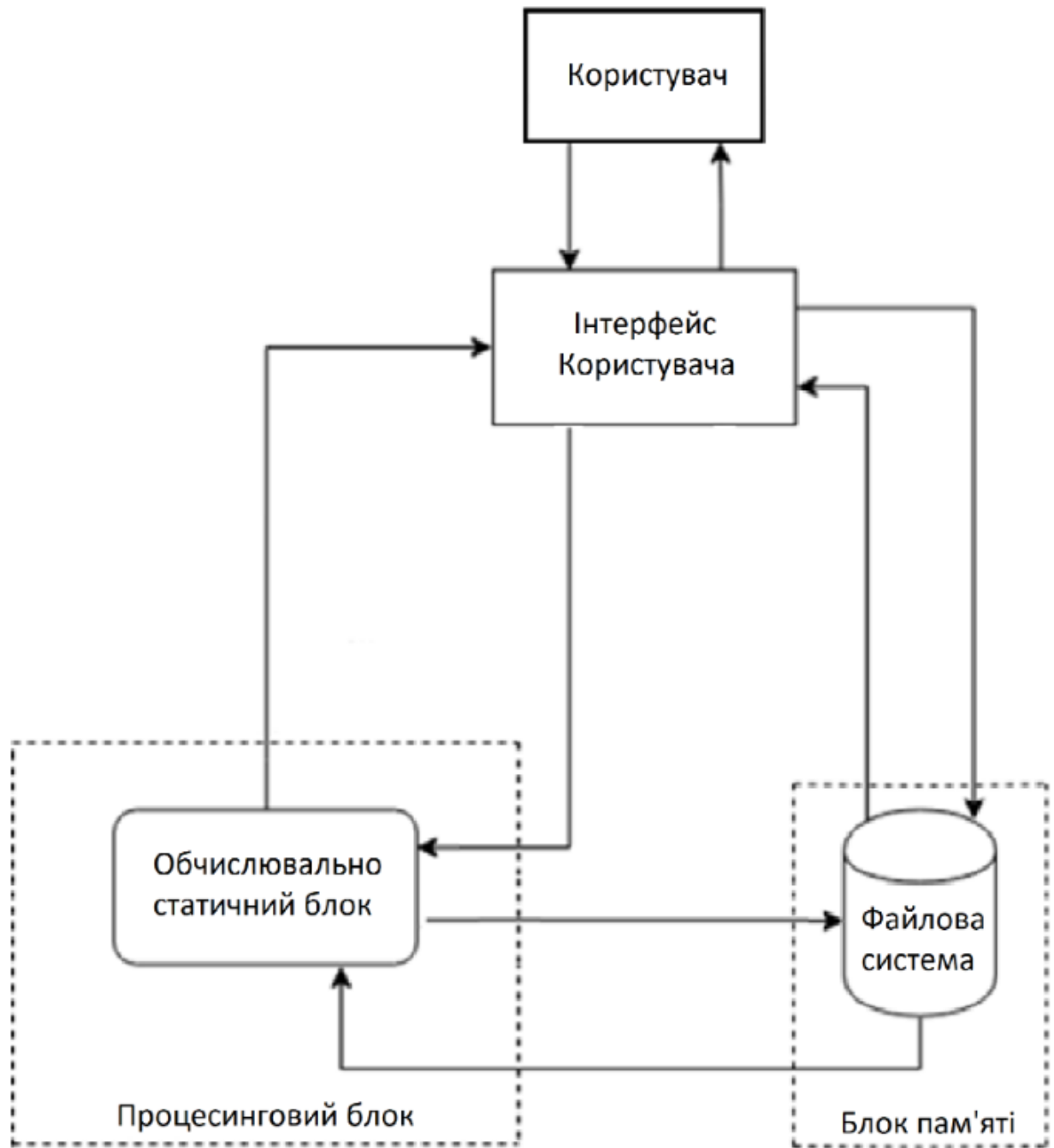
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Pan X., Hamilton A. F. d. C. Why and how to use virtual reality to study human social interaction: The challenges of exploring a new research landscape. *British Journal of Psychology*. 2018. Т. 109, № 3. С. 395–417. URL: <https://doi.org/10.1111/bjop.12290> (дата звернення: 09.06.2023).
2. DiCarlo J. J., Zoccolan D., Rust N. C. How Does the Brain Solve Visual Object Recognition?. *Neuron*. 2012. Т. 73, № 3. С. 415–434. URL: <https://doi.org/10.1016/j.neuron.2012.01.010> (дата звернення: 09.06.2023).
3. Undergraduates Interested in STEM Research Are Better Students than Their Peers / N. Bickford та ін. *Education Sciences*. 2020. Т. 10, № 6. С. 150. URL: <https://doi.org/10.3390/educsci10060150> (дата звернення: 09.06.2023).
4. Classification Designs: Attention and Interaction. *Detection Theory*. 2004. С. 185–207. URL: <https://doi.org/10.4324/9781410611147-16> (дата звернення: 02.03.2023).
5. Auditory Attention Detection via Cross-Modal Attention / S. Cai та ін. *Frontiers in Neuroscience*. 2021. Т. 15. URL: <https://doi.org/10.3389/fnins.2021.652058> (дата звернення: 02.03.2023).
6. Classification Designs: Attention and Interaction. *Detection Theory*. 2004. С. 185–207. URL: <https://doi.org/10.4324/9781410611147-16> (дата звернення: 11.04.2023).
7. Hsiung C.-Y., 熊昭岳. An Attention Detection System for Vehicles : thesis. 2005. URL: <http://ndltd.ncl.edu.tw/handle/39349773130974345063> (дата звернення: 20.04.2023).

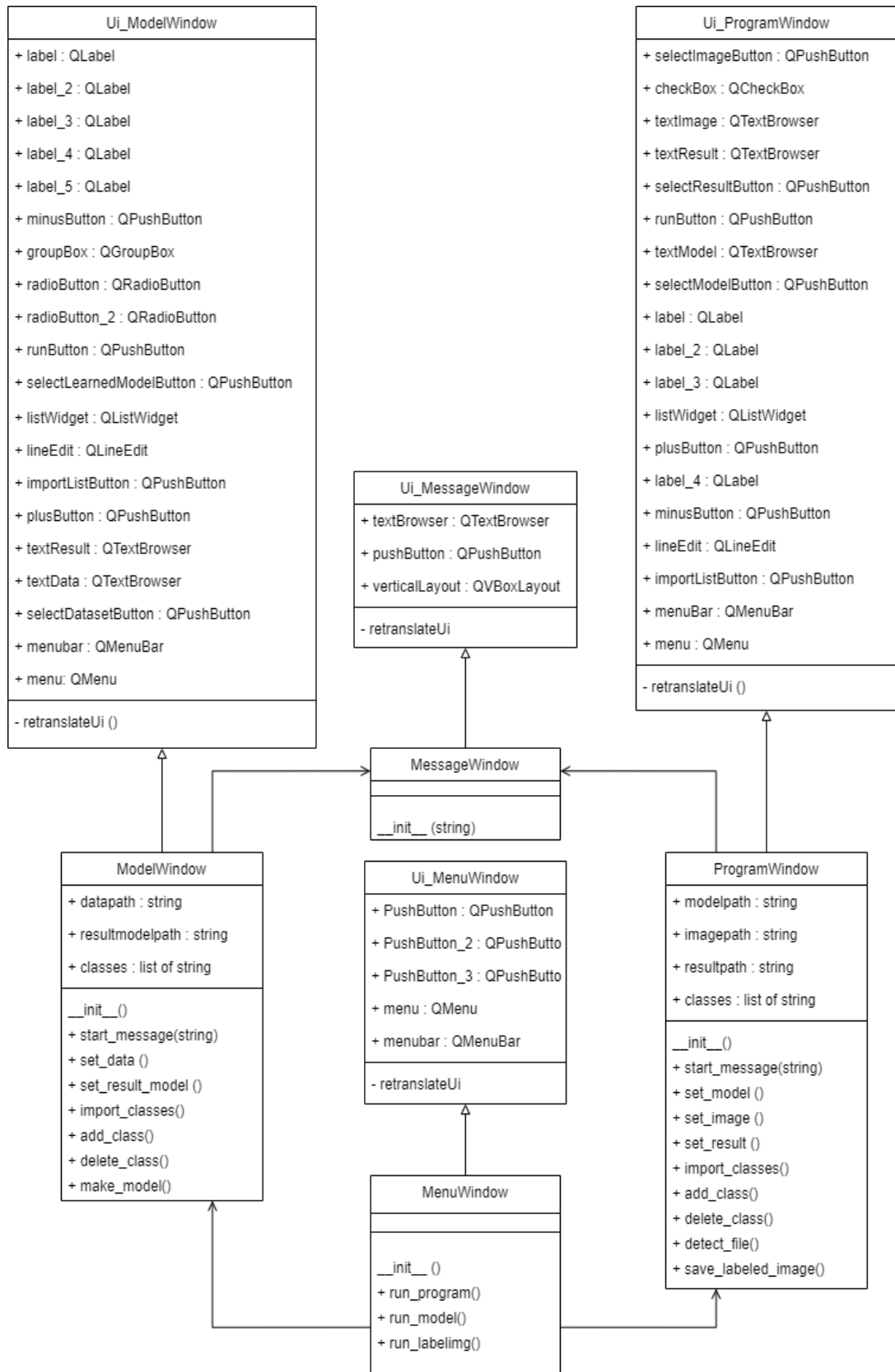
					ІАЛЦ.045440.004 ПЗ	Арк
						55
Зм	Арк.	№ докум.	Підпис	Дата		

8. Huang T., Fu R. Driver Distraction Detection Based on the True Driver's Focus of Attention. IEEE Transactions on Intelligent Transportation Systems. 2022. С. 1–13. URL: <https://doi.org/10.1109/tits.2022.3166208> (дата звернення: 10.05.2023).
9. Mahmood H. Visual Attention-based Object Detection and Recognition : thesis. 2013. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-94024> (дата звернення: 21.05.2023).
10. Zhang Libao 张., Li Hao 李. Image Focus of Attention Detection Based on Double Threshold Visual Attention Model. High Power Laser and Particle Beams. 2011. Т. 23, № 3. С. 617–622. URL: <https://doi.org/10.3788/hplpb20112303.0617> (дата звернення: 21.05.2023).

Додаток А
Копії графічних матеріалів



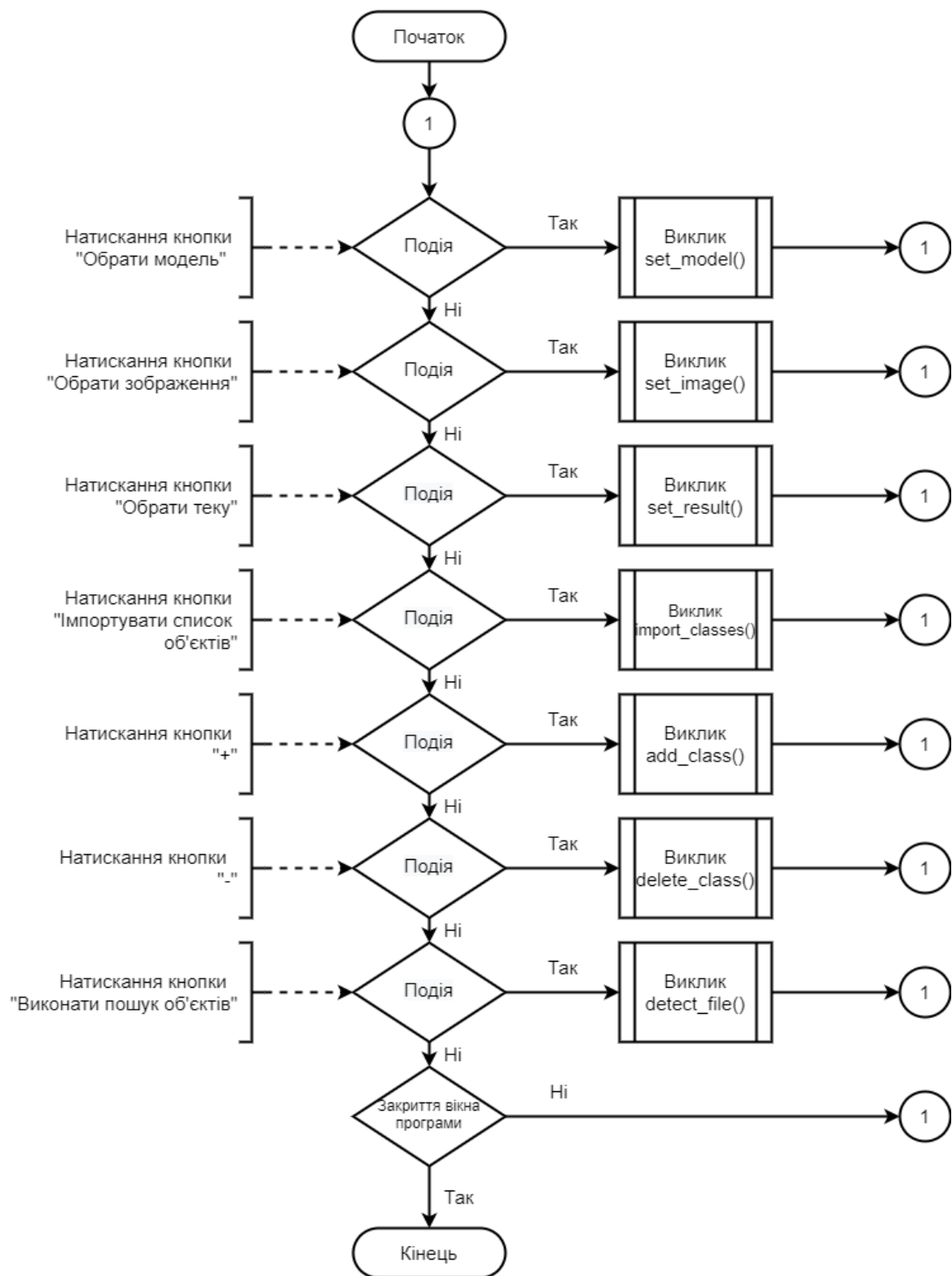
					ІАЛЦ.045490.005 Д1						
Зм.	Арк.	№ докум.	Підп.	Дата	Взаємодія модулів системи. Схема структурна.				Літ.	Аркуш	Аркушів
Розробив	Шевченко О.В.										
Перев.	Терейковский І.А.									1	1
									КПІ ім. Ігоря Сікорського, ФПМ, КВ-93		
Н. контр.	Клятченко Я.М.										
Затвер.	Романкевич В.О.										



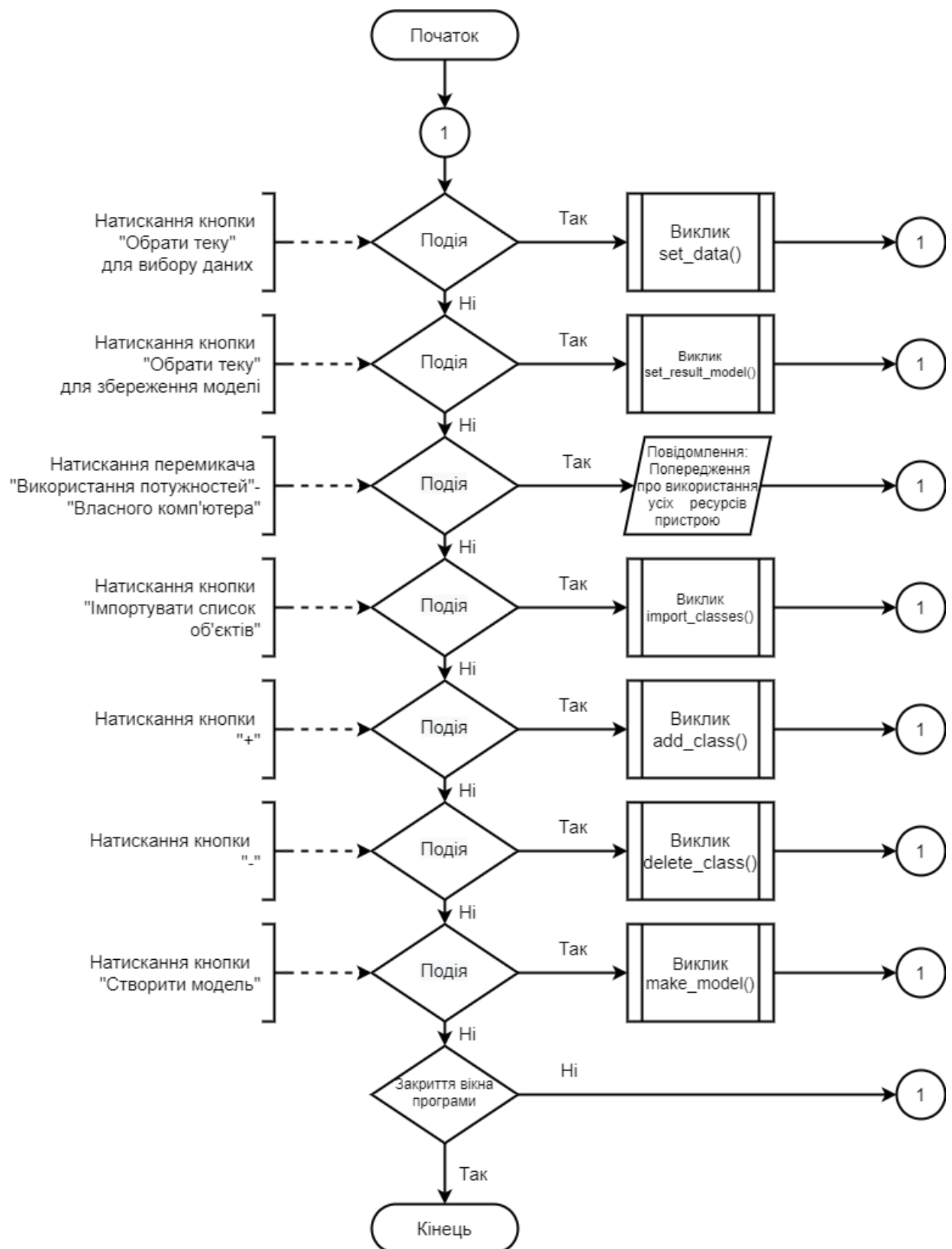
Зм.	Арк.	№ докум.	Підп.	Дата
Розробив		Шевченко О.В.		
Перев.		Терейковський І.А.		
Н. контр.		Клятченко Я.М.		
Затвер.		Романкевич В.О.		

Діаграма в нотації UML.
Діаграма класів.

Літ.	Аркуш	Аркушів
	1	1
КПІ ім. Ігоря Сікорського, ФПМ, КВ-93		



					ІАЛЦ.045490.007 ДЗ							
Зм.	Арк.	№ докум.	Підп.	Дата	Блок-схема роботи форми «Програма для розпізнавання фокусу уваги» Схема алгоритму				Літ.	Аркуш	Аркушів	
Розробив	Шевченко О.В.										1	1
Перев.	Терейковский І.А.											
Н. контр.	Клятченко Я.М.											
Затвер.	Романкевич В.О.								КПІ ім. Ігоря Сікорського, ФПМ, КВ-93			



ІАЛЦ.045490.008 Д4					Літ.			Аркуш	Аркушів
Зм.	Арк.	№ докум.	Підп.	Дата				1	1
Розробив	Шевченко О.В.				Блок-схема роботи форми «Навчання нової моделі» Схема алгоритму				КПІ ім. Ігоря Сікорського, ФПМ, КВ-93
Перев.	Терейковський І.А.								
Н. контр.	Клятченко Я.М.								
Затвер.	Романкевич В.О.								

Додаток Б
Лістинг програми

```

from PyQt5 import QtWidgets, uic
from PyQt5.QtWidgets import *
from PyQt5 import QtCore, QtGui
from PyQt5.QtGui import *
from PyQt5.QtCore import *

import threading      as THREAD
import time           as TIME
import os             as OS
import sys           as SYSTEM
import datetime       as DATETIME

import head_rotation_detect as HEAD
from HistoryDict import *

# Get the path of the UI file
path = OS.getcwd()
qtCreatorFile = path + OS.sep + "ui" + OS.sep + "mainView.ui"
Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)

class Worker(QtCore.QThread):
    message = QtCore.pyqtSignal()

    def __init__(self, parent=None):
        super(Worker, self).__init__(parent)
        self.working = True

    def __del__(self):
        self.working = False
        self.wait()

    def run(self):
        while self.working == True:
            self.message.emit()
            TIME.sleep(0.08)

class MainUi(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)

        # Get internal objects
        self.get_internal_objects()

        # Initialize video display area
        self.initVideoEvent()

        self.update_flag = True      # Update flag
        self.fps_rate = 0            # FPS
        self.pre_time = TIME.time()  # Current time

```

Рисунок 1 – Скріншот першої частини коду одного з класів модуля

```

self.angle_history = History( 2, max_size=20 ) # Create angle record
self.focus_count = 0
self.focus_timer = 0

# Reset function
self.button_reset.clicked.connect( self.reset )

# Create new thread, connect custom signal sinOut to
self.thread = Worker()
self.thread.message.connect(self.update)
self.thread.start()

def get_internal_objects(self):
    self.label_imageView = self.findChild(QLabel, name='imageView') # Image display Label
    self.label_fpsVal = self.findChild(QLabel, name='labelShow_FPS_Rate') # FPS display Label
    self.label_horizonVal = self.findChild(QLabel, name='labelShow_Angle_Horizon') # Horizontal value display Label
    self.label_verticalVal = self.findChild(QLabel, name='labelShow_Angle_Vertical') # Vertical value display Label
    self.label_headDirection = self.findChild(QLabel, name='labelShow_Head_Direction') # Current head direction Label

    self.label_horizonThreshold = self.findChild(QLabel, name='labelText_Angle_Threshold_Horizon') # Vertical threshold display Label
    self.label_verticalThreshold = self.findChild(QLabel, name='labelText_Angle_Threshold_Vertical') # Vertical threshold display Label
    self.scrollBar_horizon = self.findChild(QSlider, name='sliderBar_Threshold_Horizon') # Horizontal threshold
    self.scrollBar_vertical = self.findChild(QSlider, name='sliderBar_Threshold_Vertical') # Vertical threshold
    self.label_focusRate = self.findChild(QLabel, name='labelShow_Focus_Rate') # Focus rate

    self.button_reset = self.findChild(QPushButton, name='pushButton_Reset')
    self.checkBox_show3D = self.findChild(QCheckBox, name='checkBox_show_3D')
    self.checkBox_python = self.findChild(QCheckBox, name='checkBox_python')
    self.checkBox_showFaceMark = self.findChild(QCheckBox, name='checkBox_Show_Facemask')

def initVideoEvent(self):
    pixmap = QPixmap('image.jpg')
    self.label_imageView.setPixmap(pixmap)

def updateUI_second(self):
    pass

def reset(self):
    self.focus_count = 0
    self.focus_timer = 0

def update(self):
    QApplication.processEvents()
    show3D, showMark = self.get_checkbox_status()

```

Рисунок 2 – Скріншот другої частини коду одного з класів модуля

```

def reset(self):
    self.focus_count = 0
    self.focus_timer = 0

def update(self):
    QApplication.processEvents()
    show3D, showMark = self.get_checkbox_status()
    horizon_angle, vertical_angle, cvimg = self.detect_head_rotation(show3D, showMark)
    self.update_image(cvimg)
    hThreshold, vThreshold = self.get_scrollbar_values()
    self.update_head_rotation_values(hThreshold, vThreshold, horizon_angle, vertical_angle)
    self.calculate_fps()

def get_checkbox_status(self):
    show3D = self.checkBox_show3D.isChecked()
    showMark = self.checkBox_showFaceMark.isChecked()
    return show3D, showMark

def detect_head_rotation(self, show3D, showMark):
    return HEAD.detect(show3D, showMark)

def update_image(self, cvimg):
    try:
        image = QtGui.QImage(cvimg.data, cvimg.shape[1], cvimg.shape[0], QtGui.QImage.Format_RGB888).rgbSwapped()
        image = QtGui.QPixmap.fromImage(image)
        self.label_imageView.setPixmap(image)
    except:
        pass

def get_scrollbar_values(self):
    hThreshold = self.scrollBar_horizon.value()
    vThreshold = self.scrollBar_vertical.value()
    self.label_horizonThreshold.setText('Horizontal threshold: ' + str(hThreshold))
    self.label_verticalThreshold.setText('Vertical threshold: ' + str(vThreshold))
    return hThreshold, vThreshold

def update_head_rotation_values(self, hThreshold, vThreshold, horizon_angle, vertical_angle):
    self.angle_history.add([horizon_angle, vertical_angle])
    if self.angle_history.is_full():
        [horizon_angle, vertical_angle] = self.angle_history.get_average()
        self.angle_history.pop()

    self.label_horizonVal.setText(str(horizon_angle))
    self.label_verticalVal.setText(str(vertical_angle))

    head_rotation, head_rotation_str = HEAD.judge_look(horizon_angle=horizon_angle, vertical_angle=vertical_angle,
                                                         horizon_threshold=hThreshold, vertical_threshold=vThreshold)

```

Рисунок 3 – Скріншот третьої частини коду одного з класів модуля

```

        head_rotation, head_rotation_str = HEAD.judge_look(horizon_angle=horizon_angle, vertical_angle=vertical_angle,
                                                           horizon_threshold=hThreshold, vertical_threshold=vThreshold)

        rotation_str = 'Current head direction: ' + str(head_rotation_str)
        self.label_headDirection.setText(rotation_str)

        if head_rotation == HEAD.LEFT or head_rotation == HEAD.RIGHT or head_rotation == HEAD.GONE:
            self.focus_count += 1
            self.focus_timer += 1
            self.label_focusRate.setText(str(round((1 - (self.focus_count / self.focus_timer)) * 100, 2)) + '%')

    def calculate_fps(self):
        self.fps_rate += 1
        now_time = TIME.time()
        if (now_time - self.pre_time) > 1:
            self.updateUI_second()

            self.label_fpsVal.setText(str(self.fps_rate))
            self.fps_rate = 0
            self.pre_time = now_time

if __name__ == "__main__":
    def run_app():
        app = QtWidgets.QApplication(SYSTEM.argv)
        window = MainUi()
        window.show()
        app.exec_()

run_app()

```

Рисунок 4 – Скріншот останньої частини коду одного з класів модуля