

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня бакалавр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Інформаційна технологія розпізнавання емоційного стану співрозмовників під час онлайн-спілкування

Здобувача групи IT-03 Суровцева Артема Віталійовича  
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.



(підпис)

Артем СУРОВЦЕВ

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник Бойко Ольга Василівна, к.т.н.  
(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

**Суми – 2024**

Сумський державний університет  
Факультет електроніки та інформаційних технологій  
Кафедра інформаційних технологій  
Спеціальність 122 «Комп'ютерні науки»  
Освітньо-професійна програма «Інформаційні технології проектування»

**ЗАТВЕРДЖУЮ**  
В. о. зав. кафедри ІТ  
Світлана ВАЩЕНКО  
«\_\_» 2024 р.

## **З А В Д А Н Н Я** **НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

Суровцева Артема Віталійовича

**1 Тема роботи** Інформаційна технологія розпізнавання емоційного стану співрозмовників під час онлайн-спілкування

**керівник роботи** Бойко Ольга Василівна, к.т.н., доцент,

затверджені наказом по університету від 07.05.2024 р. №0482 -VI

**2 Строк подання студентом роботи** 30.05.2024 р.

### **3 Вхідні дані до роботи**

Метою роботи є розробка та впровадження інформаційної технології розпізнавання емоційного стану співрозмовників під час онлайн-спілкування;

– **4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** аналіз предметної області, постановка задачі, моделювання та проектування, програмна реалізація, висновки;

### **5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

порівняння продуктів-аналогів, архітектура програмного продукту, діаграма у нотації IDEF0, діаграма декомпозиції у нотації IDEF0, діаграма варіантів використання сервісу, діаграма бази даних, архітектура десктопного додатку, приклади роботи програмного продукту.

**6. Консультанти розділів роботи:**

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Визначення мети роботи, ознайомлення з базовими поняттями обраної тематики	08.04.2024 – 10.04.2024	
2	Проведення порівняльного аналізу існуючих підходів до автоматичного розпізнавання емоцій	11.04.2024 – 20.04.2024	
3	Знаходження алгоритму та моделі автоматичного розпізнавання емоцій	20.04.2024 – 25.04.2024	
4	Створення інформаційної технології розпізнавання емоцій для онлайн спілкування	25.04.2024– 29.04.2024	
5	Створення кінцевого функціонального додатку	30.04.2024– 22.05.2024	
6	Оформлення пояснювальної записки звіту про виконану роботу	23.05.2024– 30.05.2024	

Студент



Артем СУРОВЦЕВ

Керівник роботи

---

 (підпис)

к.т.н. Ольга БОЙКО

---

 (підпис)

## АНОТАЦІЯ

Тема кваліфікаційної роботи бакалавра «Інформаційна технологія розпізнавання емоційного стану співрозмовників під час онлайн-спілкування».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 23 найменувань, додатків. Загальний обсяг роботи – 67 сторінок, у тому числі 30 сторінок основного тексту, 3 сторінки списку використаних джерел, 29 сторінок додатків.

Актуальність роботи полягає в тому що, дана предметна область є динамічно розвиваюча галузь досліджень. Запропонована інформаційна технологія розпізнавання емоцій дозволить швидко виявляти рівень залучення та емоційний стан співрозмовників під час комунікації в онлайн-середовищах.

Метою роботи є розробка інформаційної технології розпізнавання емоцій для оцінки залучення та емоційного стану співрозмовників під час онлайн-спілкування з використанням класифікатора на основі згорткової нейронної мережі.

Результати роботи були представлені та обговорені на Міжнародній науково-технічній конференції молодих вчених «Інформатика, математика, автоматика, ІМА-2024» (м. Суми, Україна).

**Ключові слова:** CNN, tensorflow, згорткова нейронна модель, розпізнавання емоцій

## ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень .....	8
1.2 Аналіз існуючих продуктів-аналогів.....	10
1.3 Постановка задачі.....	14
1.4 Вибір методів дослідження .....	15
1.4.1 Загальні відомості про модель CNN. ....	15
1.4.2 Набір даних.....	17
1.4.3 Алгоритм автоматичного розпізнавання емоцій.....	17
1.4.4 Вибір стеку технологій для реалізації додатку. ....	19
2 ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ЕМОЦІЙ.....	21
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	26
ВИСНОВКИ.....	35
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	36
ДОДАТОК А Технічне завдання .....	39
ДОДАТОК Б .....	45
ПЛАНУВАННЯ РОБІТ .....	45
ДОДАТОК Г .....	59

## ВСТУП

В останні роки кількість онлайн-спілкувань значно зросла, особливо під час пандемії COVID-19. Відеоконференції, чати та соціальні мережі стали основними засобами взаємодії як у професійному, так і в особистому житті. Емоційний стан є важливим аспектом спілкування, який допомагає зрозуміти контекст і наміри співрозмовника. Технології, що розпізнають емоції, можуть покращити якість взаємодії, допомагаючи уникнути непорозумінь і конфліктів.

Нещодавній прогрес в областях машинного навчання та комп'ютерного зору сприяли створенню розширених систем для автоматизованого визначення емоцій. Такі системи здатні аналізувати міміку, інтонації голосу, рухи тіла та інші біологічні індикатори для точного розпізнавання емоційних станів особи. Впровадження цих технологій відкриває широкі перспективи для покращення інтеракцій між людьми та комп'ютерами, зокрема у сферах робототехніки, відеоігор, цифрової реклами та електронного навчання. Ці інновації мають потенціал істотно вдосконалити онлайн-спілкування, зробивши його більш особистісним та результативним. Тому згортова модель для розпізнавання емоцій є актуальною темою для розробки та впровадження в процес онлайн-спілкування.

**Метою** даної роботи є розробка та впровадження інформаційної технології розпізнавання емоційного стану співрозмовників під час онлайн-спілкування за рахунок розробки класифікатора на основі згорткової нейронної мережі.

Для досягнення поставленої мети необхідно вирішити наступні **задачі**:

- здійснити порівняльний аналіз наявних підходів до автоматичного розпізнавання емоцій під час онлайн спілкування;
- розробити алгоритм та модель розпізнавання емоцій за виразом обличчя, використовуючи сучасні досягнення в машинному навчанні;
- провести структурно-функціональне моделювання інформаційної технології розпізнавання емоцій;

- запропонувати інформаційну технологію розпізнавання емоцій та спроектувати додаток, що використовує її;
- провести тестування додатку.

**Об'єкт дослідження** – процес онлайн-спілкування з використанням інформаційних технологій, що дозволяють розпізнавати емоційний стан співрозмовників.

**Предмет дослідження** – інформаційна технологія розпізнавання емоційного стану співрозмовників, зокрема алгоритми на основі згорткових нейронних мереж для аналізу виразів обличчя під час онлайн-спілкування.

**Практичне значення.** Запропонована інформаційна технологія для розпізнавання емоцій дасть змогу визначати емоційний стан співрозмовників під час онлайн-комунікації. Також дозволить швидко виявляти рівень залучення та емоційний стан співрозмовників під час комунікації в онлайн-середовищах. Це покращить розуміння людей, щодо кращої подачі інформації та зробить онлайн-спілкування більш змістовним.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень

Галузь автоматизованого розпізнавання емоцій динамічно розвивається, метою досліджень даної галузі є розробка моделей та комп'ютерних систем, що здатні інтерпретувати людські емоції. За даними бази даних Scopus у період з 2019 до 2023 року ця тема розкривається у 1682 роботах. На рисунку 1.1 наведено графік, що показує динаміку публікацій за темою. Для пошуку використовувались ключові слова (“Automatic” AND “Emotion” and “Recognition”).

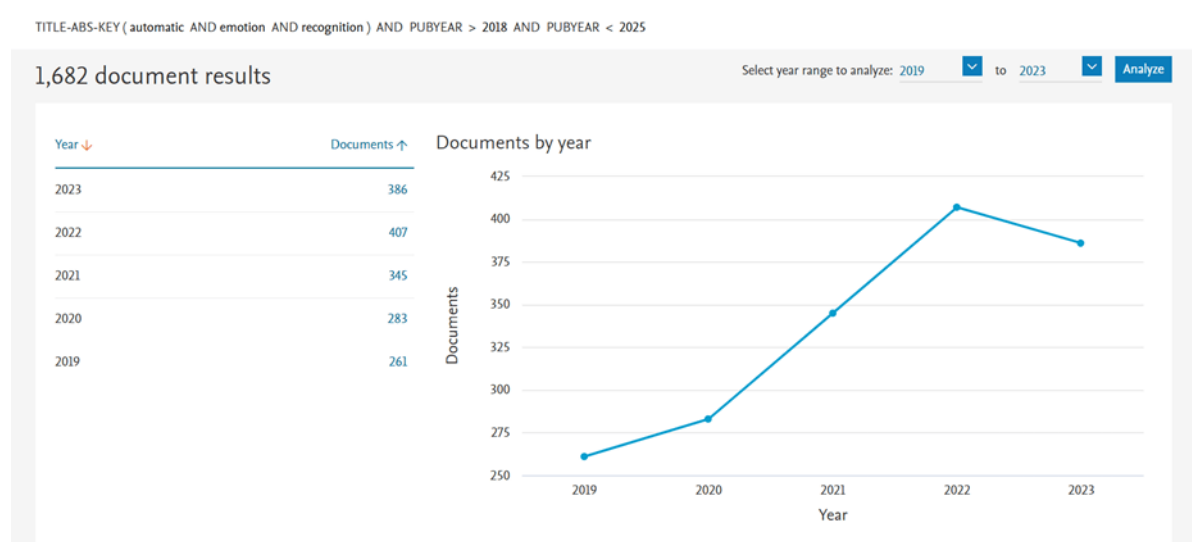


Рисунок 1.1 – Графік динаміки наукових публікацій з галузі автоматичного розпізнавання емоцій у базі даних Scopus за період з 2019 по 2023 рік

За допомогою програми VOSviewer, було проведено аналіз ключових слів та анотацій наукових досліджень. Після аналізу було побудовано мережеву візуалізацію (рис. 1.2). З візуалізації чітко видно, що дослідження в галузі автоматичного розпізнавання емоцій зосереджені в двох основних напрямках [2]:

1. Аналіз емоцій на основі голосу. Ця галузь наукових досліджень зосереджена на вивченні таких акустичних особливостей мовлення, як висота голосу, швидкість мовлення, сила голосу та мелодика, щоб визначити емоційний стан особи, що говорить.



2. Розпізнавання емоцій за виразом обличчя. Ця область досліджень приділяє увагу аналізу жестів обличчя та інших візуальних ознак для ідентифікації емоційної реакції людини.

Варто підкреслити, що ці два напрями досліджень не є ізольованими, а мають значні спільні аспекти. Наприклад, у обох сферах науковці використовують подібні техніки машинного навчання для обробки даних [1]. Крім того, дослідники активно працюють над інтеграцією різних методів розпізнавання емоцій, щоб підвищити точність та надійність результатів [3], [4]. Наявність класів з іншими аспектами емоційної обробки, зокрема, психологічних основ емоцій та їх зв'язків з різними групами людей, підкреслює комплексний і міждисциплінарний характер досліджень у цій галузі.

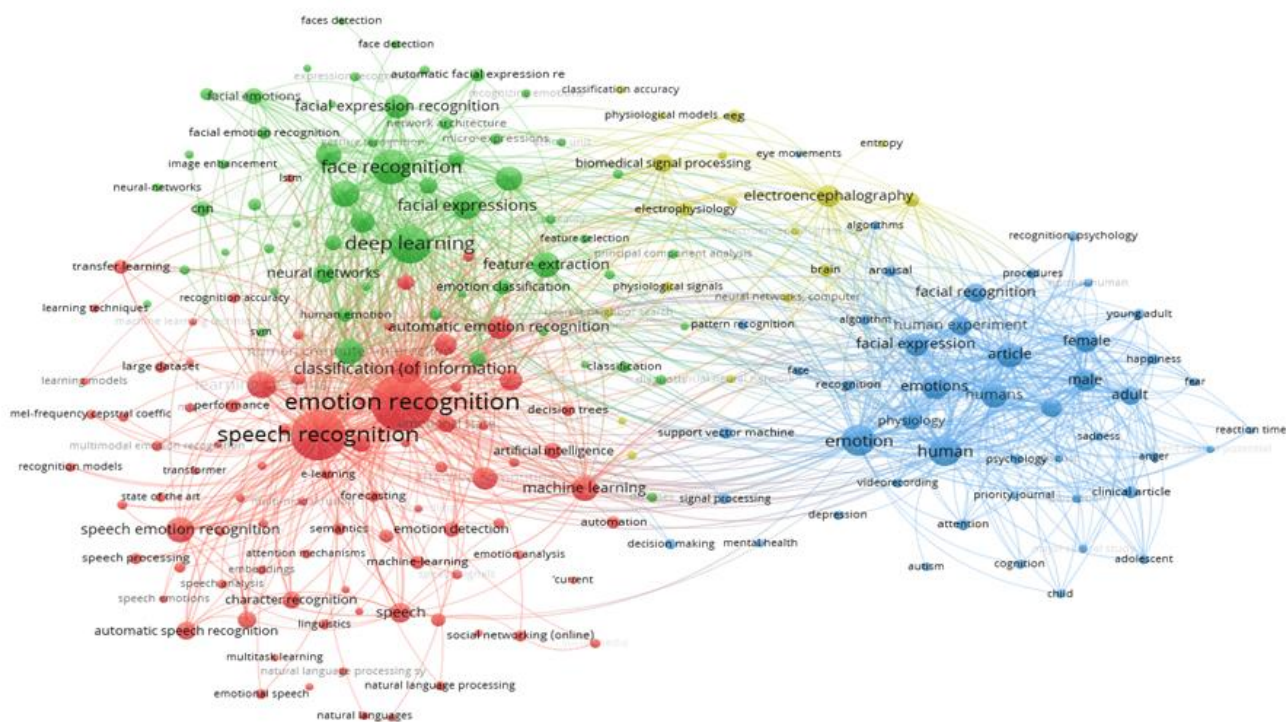


Рисунок 1.2 – Візуалізація мережевої структури тематики досліджень у сфері автоматичного розпізнавання емоцій

Отже, беручи до уваги активний розвиток галузі розпізнавання емоцій можна дійти до висновку, що обрана тема є актуальною для розробки та покращення точності моделі.

## 1.2 Аналіз існуючих продуктів-аналогів

1. FaceReader від Noldus Information Technology [5]: FaceReader – це програмний інструмент, який аналізує вирази обличчя та надає інформацію про емоції. Він використовує методи комп'ютерного зору для виявлення та інтерпретації рухів м'язів обличчя, що допомагає дослідникам та фахівцям розуміти емоційні реакції.

Пошук обличчя. FaceReader знаходить положення обличчя на зображенні за допомогою алгоритму пошуку обличчя, який базується на глибокому навчанні. Цей алгоритм шукає ділянки на зображенні, які мають вигляд обличчя, у різних масштабах.

Моделювання обличчя. FaceReader використовує техніку моделювання обличчя на основі глибоких нейронних мереж. Він створює штучну модель обличчя, яка описує розташування 468 ключових точок на обличчі. Це швидкий метод за один прохід, щоб безпосередньо оцінити повну колекцію орієнтирів на обличчі. Після початкової оцінки програмне забезпечення стискає ключові точки за допомогою аналізу основних компонентів. Це призводить до сильно стиснутого векторного представлення, що описує стан обличчя.

Класифікація обличчя. Навчена глибока штучна нейронна мережа розпізнає візерунки на обличчі, а потім класифікує міміку. FaceReader безпосередньо класифікує вирази обличчя з пікселів зображення. Для навчання штучної нейронної мережі було використано понад 20 000 зображень, анотованих вручну[5].

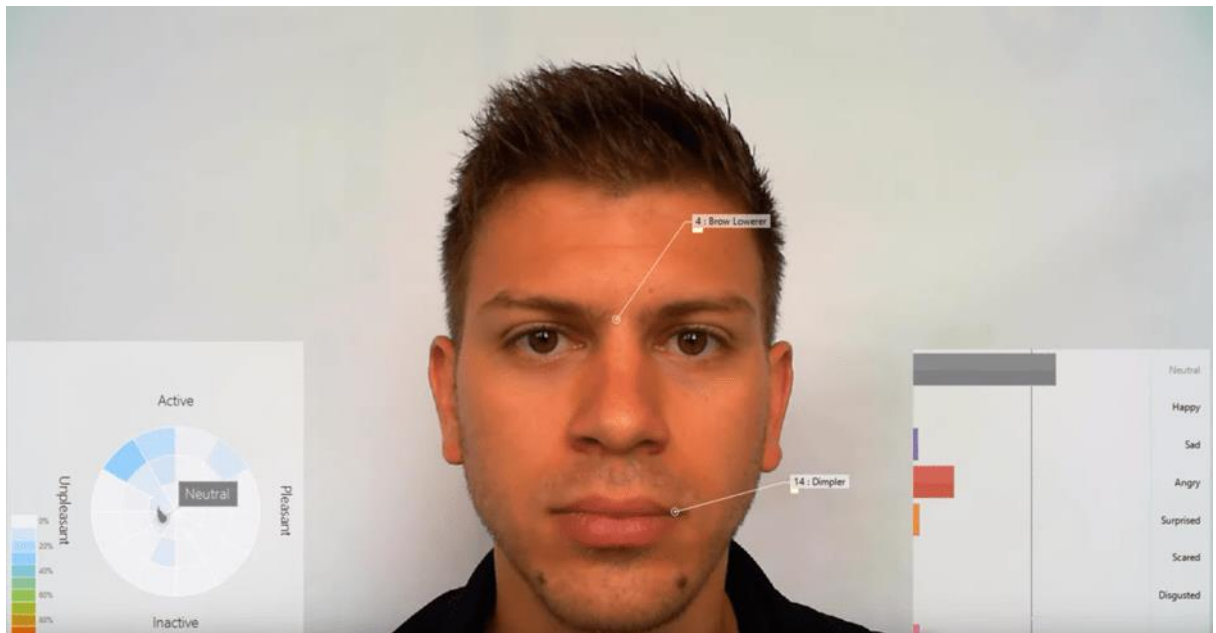


Рисунок 1.3 – Демонстрація роботи додатку FaceReader

2. Microsoft Oxford Project Emotion Recognition: Проект Microsoft спрямований на розпізнавання емоцій на зображеннях. Він використовує моделі глибокого навчання для аналізу рис обличчя та передбачення емоційних станів. Ця технологія є основою для великих проривів, включаючи переклад у реальному часі Skype Translator і персонального помічника Microsoft Cortana. Технологію розпізнавання облич, яка входить до складу Microsoft Project Oxford, також можна використовувати багатьма іншими способами, наприклад групувати колекції фотографій на основі облич людей, які на них зображені[6].

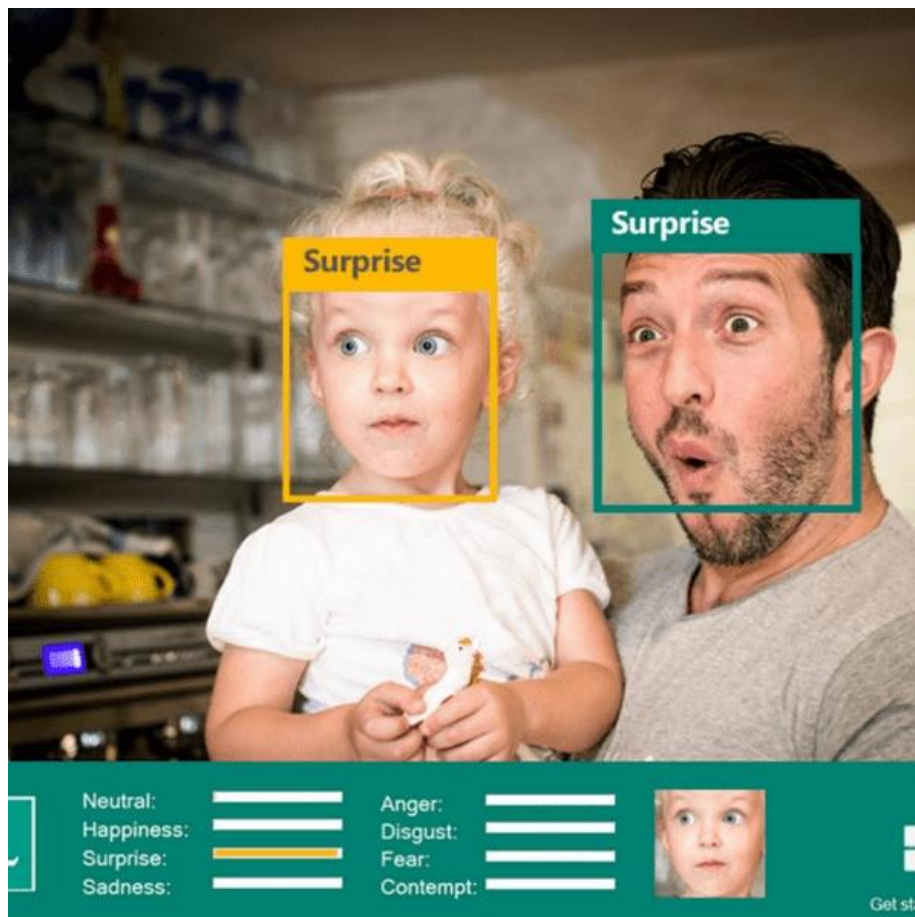


Рисунок 1.4 – Інтерфейс додатку Microsoft Oxford Project Emotion Recognition

3. Visage Technologies: Продукт компанії FaceAnalysis спрямований на розпізнавання емоцій отриманих з зображення, відео чи веб-камери. Програма має весь функціонал, який є звичним для додатків з розпізнавання емоцій. Вона повертає розподіл вірогідностей кожної з 7 універсальних емоцій. Також програма використовує покращене машинне навчання, що дозволяє вірно класифікувати стать та вік людини в режимі реального часу. Можна перевірити розпізнавання емоцій прямо на сайті, також розробникам доступна бібліотека SDK для включення функціоналу в свої проекти[7].



Рисунок 1.5 – Інтерфейс додатку Visage Technologies

Проаналізувавши існуючі аналоги в моїй предметній області, я можу створити їх порівняльну таблицю (табл 1.1).

Таблиця 1.1 – Порівняльна характеристика програм для розпізнавання емоцій людей

Критерії порівняння	Програмний додаток		
	FaceReader	Microsoft Oxford Project Emotion Recognition	Visage Technologies
Наявні додаткові характеристики (вік, стать)	Так	Так	Так
Розпізнає всі види емоцій	Так	Так	Так
Точність розпізнавання	Висока	Висока	Висока
Наявне SDK/API для можливості інтеграції з програмами розробників	Так	Так	Так
Можливість безкоштовного використання	Ні	Доступна пробна версія	Доступна пробна версія

### 1.3 Постановка задачі

Метою даної роботи є розробка та впровадження інформаційної технології розпізнавання емоційного стану співрозмовників під час онлайн-спілкування за рахунок розробки класифікатора на основі згорткової нейронної мережі. Ця технологія буде використовуватися для оцінки рівня залученості та емоційного стану учасників онлайн-зустрічей у відповідній системі, та буде базуватися на моделі штучного інтелекту, яка здатна ефективно розпізнавати емоції за виразом обличчя та враховувати їх у контексті онлайн-спілкування.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести порівняльний аналіз існуючих підходів до автоматичного розпізнавання емоцій при онлайн спілкуванні;
- розробити алгоритм та модель автоматичного розпізнавання емоцій, що базуються на виразі обличчя, враховуючи сучасні досягнення в галузі машинного навчання;
- провести структурно-функціональне моделювання інформаційної технології розпізнавання емоцій;
- запропонувати інформаційну технологію розпізнавання емоцій та спроектувати додаток, що використовує її;
- провести тестування додатку.

Більш детальні вимоги до проекту описані у технічному завданні на розробку проекту (додаток А).

Ось деякі приклади функціональних та нефункціональних вимог до додатку для розпізнавання емоцій:

#### **Функціональні вимоги:**

Розпізнавання емоцій. Додаток повинен мати здатність розпізнавати емоції на основі виразу обличчя користувача.

- Аналіз емоцій. Додаток повинен аналізувати розпізнані емоції та враховувати їх у контексті навчального процесу.
- Збереження даних. Додаток повинен мати можливість зберігати дані про виявлені емоції для подальшого аналізу.
- Захист даних. Додаток повинен гарантувати конфіденційність та захист персональних даних користувачів [8].

#### **Нефункціональні вимоги:**

- Продуктивність. Додаток повинен працювати ефективно та швидко, незалежно від кількості користувачів.
- Безпека. Додаток повинен використовувати сучасні методи шифрування для захисту персональних даних користувачів.
- Надійність. Додаток повинен бути стабільним та надійним, забезпечуючи постійний доступ до своїх функцій.
- Сумісність. Додаток повинен бути сумісний з різними операційними системами та пристроями.
- Приватність. Додаток повинен дотримуватися всіх відповідних законів та нормативів щодо приватності та захисту даних.

## **1.4 Вибір методів дослідження**

### **1.4.1 Загальні відомості про модель CNN.**

Завдяки своїй архітектурі та здатності аналізувати і локальні, і глобальні візуальні характеристики, згорткові нейронні мережі (CNN) стали одним з найдієвіших методів для обробки зображень [9]. У цьому дослідженні ми застосовуємо згорткові нейронні мережі (CNN) для розпізнавання емоцій за виразами обличчя. CNN навчається на наборі даних, що містить зображення облич із позначеними емоціями. Після завершення навчання CNN здатна визначати емоції на нових зображеннях облич.

Типова CNN включає вхідний шар, кілька прихованих шарів, що здійснюють згортку та об'єднання, а також вихідний шар [10], [11] (рис. 1.6).

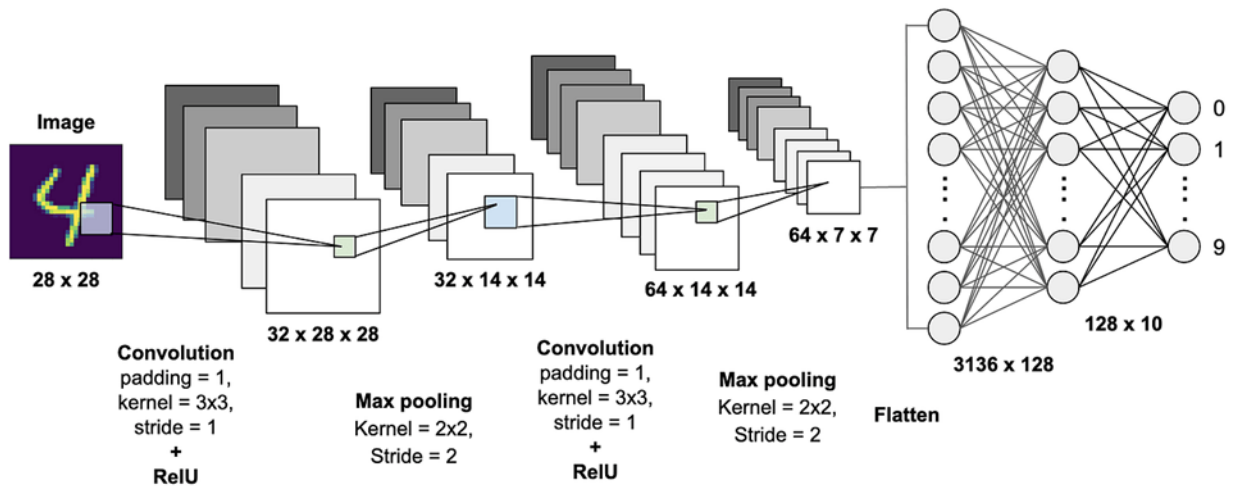


Рисунок 1.6 – Структура згорткової нейронної мережі з двома шарами згорткового пулінгу

*Джерело:* [12]

Вхідний шар отримує зображення як вхідні дані. Згорткові шари застосовують операцію згортки до цього зображення, виявляючи корисні візуальні шаблони. Основу функціонування CNN становить операція згортки, в якій маленьке ядро (наприклад, 3x3 або 5x5) сканує зображення, виявляючи місцеві візуальні відмінності. Це ядро переміщується по всьому зображенню, множачи свої значення на значення пікселів, що дозволяє створити карту ознак. Шари об'єднання виконують функцію зменшення розмірності зображення, що робить його менш чутливим до шуму та варіацій. Це сприяє узагальненню візуальних ознак та підвищує стійкість моделі до перенавчання. Після вилучення ознак за допомогою згорткових та об'єднувальних шарів, для класифікації або регресії використовуються повністю з'єднані шари. Вони схожі на нейронні мережі прямого поширення, де кожен нейрон одного шару з'єднаний з усіма нейронами наступного шару. Вихідний шар видає кінцевий результат, наприклад, ймовірність класу або прогнозоване значення.



### 1.4.2 Набір даних.

Це дослідження базується на використанні відкритого набору даних FER2013 для ідентифікації емоцій за допомогою аналізу виразів обличчя [13]. База даних Facial Expression Recognition 2013 (FER2013), яка була вперше представлена на конкурсі ICML 2013 Challenges in Representation Learning, включає 35 887 зображень з роздільною здатністю  $48 \times 48$ , більшість з яких були зібрані в неконтрольованому середовищі. Набір для навчання містить 28 709 зображень, тоді як валідаційний і тестовий набори містять по 3 589 зображень. Дані FER2013 були зібрані за допомогою API пошуку Google Images, при цьому обличчя на зображеннях автоматично позначалися. Зображення класифіковані за сімома основними типами емоцій, а саме радості, смутку, гніву, страху, огиди, здивування та нейтрального стану.

На відміну від інших наборів даних [14], [15], FER2013 характеризується великим різноманіттям зображень, включаючи оклюзії обличчя (переважно рукою), зображення людей в окулярах, зображення з низькою контрастністю та часткові обличчя.

Зразки зображень із набору даних FER2013 наведені на рисунку 1.7.



Рисунок 1.7 – Фрагменти зображень з датасету FER2013

*Джерело:* [13].

### 1.4.3 Алгоритм автоматичного розпізнавання емоцій.

Суть алгоритму розпізнавання емоцій, що застосовується в даній роботі, можна описати таким чином.

На вхід подається зображення «image», яке підлягає обробці для розпізнавання емоцій. На виході ми отримуємо прогнозовану емоцію «predicted\_emotion», яка була виявлена на зображенні.

Етапи обробки включають:

1. **Підготовка зображення.** Конвертується зображення в формат, що використовується моделлю, перетворюючи його з RGB в Grayscale за формулою 1.1. Цей процес виконується для кожного окремого пікселя на зображенні, з вказаними координатами (x, y) :

$$grayimg[y, x] = 0.2989 \times img[y, x, 0] + 0.5870 \times img[y, x, 1] + 0.1140 \times img[y, x, 2] \quad (1.1)$$

2. **Виявлення обличчя.** Виявлення обличчя. Використовується каскад Хаара для виявлення обличчя за допомогою набору правил, описаних у формулі 1.2 [16].

$$Haarcascade = \sum wi \times fi(x) \quad (1.2)$$

де  $wi$ ,  $fi(x)$  - відповідно вага та ознака i-го правила [17].

3. **Збільшення зображення обличчя.** Вирізається з координатами (x, y, w, h) зображення обличчя з екранного зображення та масштабується за допомогою білінійної інтерполяції (формула 1.3).

$$newimg[y, x] = \sum \sum img[yi, xi] \times h(x - xi, y - yi) \quad (1.3)$$

де  $h(x, y)$  - білінійна інтерполяційна функція.

4. **Визначення емоції.** Передається збільшене зображення обличчя в модель для визначення емоції. Модель використовує CNN для екстракції ознак, які потім використовуються для класифікації емоцій за допомогою повністю зв'язаної нейронної мережі (FCNN).

Функція активації CNN:

$$f(x) = \max(0, x) \quad (1.4)$$

Функція активації FCNN:

$$f(x) = \text{softmax}(x) \quad (1.5)$$

5. **Відображення результату.** Додається текст з назвою емоції на екранне зображення та виводиться на екран.

Список емоцій:

$$emotionlabels = ["Angry", "Disgusted", "Fearful", "Happy", "Neutral", "Sad", "Surprised"] \quad (1.6)$$

Блок-схема алгоритму розпізнавання емоцій на обличчі наведена на рисунку 1.8.



Рисунок 1.8 – Схема алгоритму визначення емоцій на обличчі

#### 1.4.4 Вибір стеку технологій для реалізації додатку.

За мову програмування мною був обраний Python, оскільки він є одним з найпопулярніших та найбільш гнучких інструментів для розробки програмного забезпечення та роботі з машинним навчанням. Python відомий широким спектром бібліотек та фреймворків, які сприяють реалізації різноманітних завдань, і забезпечує зручний синтаксис, що спрощує розробку.

Для виявлення обличчя на екрані було обрано такі бібліотеки:

1. Бібліотека `mss`. Ця бібліотека використовується для трансляції зображення з екрану в інтерфейс програми. Вона дозволяє отримувати знімки екрана з великою швидкістю, що дозволяє ефективно використовувати їх для подальшої обробки.

2. Бібліотека `OpenCV`. `OpenCV` є потужним інструментом для обробки зображень і відео. Використовуючи метод `Haar Cascade`, вона дозволяє виявляти обличчя на зображеннях та відео. `Haar Cascade` - це метод машинного навчання, який використовується для виявлення об'єктів на зображеннях шляхом порівняння їх з шаблонами.

3. Бібліотеки `TensorFlow` та `Keras`. `TensorFlow` і `Keras` використовуються для побудови згорткових нейронних мереж. Вони надають інструменти для створення, навчання та використання нейронних мереж для різних завдань машинного навчання, включаючи виявлення обличчя[18].

4. Інструментарій `QTDesigner`. `QTDesigner` використовується для створення інтерфейсу користувача програми. Він надає зручний інтерфейс для розміщення елементів у програмі та налаштування їх вигляду та поведінки, що дозволяє легко створювати зручний та привабливий інтерфейс для користувача [19].

## 2 ФУНКЦІОНАЛЬНЕ МОДЕЛЮВАННЯ ІНФОРМАЦІЙОЇ ТЕХНОЛОГІЇ РОЗПІЗНАВАННЯ ЕМОЦІЙ

Згідно [20] «інформаційна технологія» є процесом, що містить в собі етапи збору, обробки, збереження та розповсюдження інформації, взаємопов'язані між собою. Для ілюстрації ключових аспектів процесу розпізнавання емоцій, який може розглядатися як складова частина інформаційної технології, застосування методів функціонального моделювання видається доцільним. Ці методи дозволяють структурувати та графічно відобразити послідовність операцій, необхідних для ефективного розпізнавання емоцій, включаючи збір і аналіз вхідних даних, їх обробку за допомогою відповідних алгоритмів, збереження результатів та відображення отриманих емоційних станів.

Серед відомих методів функціонального моделювання варто відзначити IDEF0. Це скорочення означає "Integration Definition for Function Modeling" і використовується для детального опису функціональної роботи інформаційної технології [21].

Рисунки 2.1 та 2.2 надають відображення контекстної діаграми процесу розпізнавання емоцій та її декомпозиції, що описує основні кроки в інформаційній технології:

1. **Збір даних.** Отримання вхідних даних з екрану користувача, які представлені послідовністю транслюючихся зображен.

2. **Попередня обробка даних.** Перетворення отриманих даних у формат, придатний для використання моделлю (чорно-білі зображення розміром 48x48), щоб мати змогу обробляти дані за допомогою попередньо підготовленої моделі.

3. **Екстракція ознак емоцій.** З отриманих даних, зібраних на попередньому етапі, виділяються ключові характеристики, що визначають емоційний стан. Згорткові шари застосовують локальні фільтри для вивчення

місцевих шаблонів та особливостей вхідних зображень. Ці фільтри, рухаючись по всьому вхідному зображенню, дозволяють моделі виявляти місцеві шаблони, такі як різкі переходи кольорів, контури або текстурні особливості.

У згортковій операції локальний фільтр застосовується для точкового множення з областю вхідного зображення, результатом якого є нове значення в вихідному зображенні, залежне від ваг фільтра. Згорткові шари можуть містити кілька таких фільтрів, кожен з яких спрямований на вивчення різних характеристик. Кожен фільтр породжує окремий канал (або карту ознак) в вихідному зображенні, що відображає присутність конкретних локальних зразків або ознак. Під час навчання згорткові шари автоматично адаптують свої ваги для розпізнавання різних глобальних особливостей.

**4. Класифікація емоцій.** Відбувається оцінка та подальша категоризація вилучених характеристик емоцій, отриманих за допомогою навченої згорткової нейронної мережі.

**5. Візуалізація результатів.** Результати розпізнавання емоцій відображаються у текстовому форматі на екрані додатку.

Необхідно підкреслити, що всі кроки у процесі застосування інформаційної технології для розпізнавання емоцій взаємозв'язані і виконуються у певній послідовності. Результативність кожного з них впливає на загальну точність та надійність розпізнавання емоцій.



Рисунок 2.1 – Контекстна діаграма

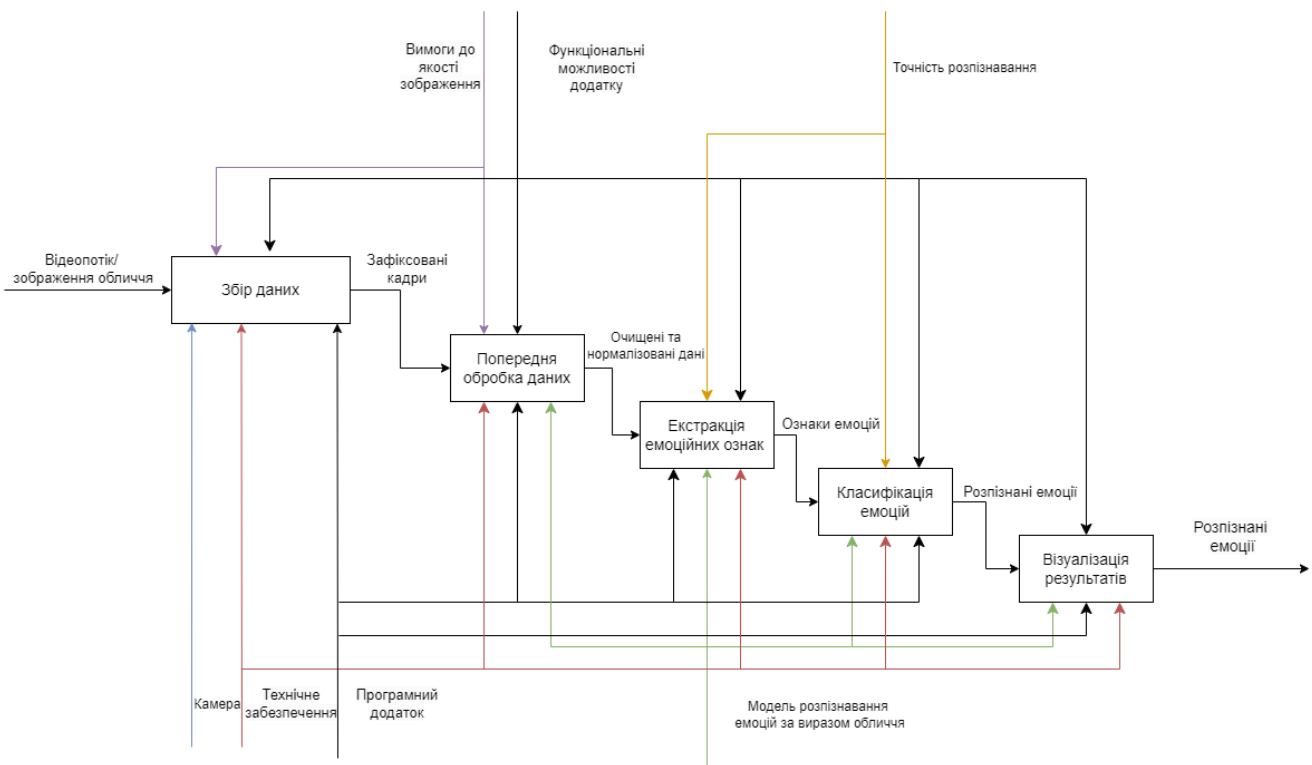


Рисунок 2.2 – Діаграма декомпозиції процесу розпізнавання емоцій

Діаграма варіантів використання зображена на рисунку 2.3.

Рисунок 2.4 демонструє діаграму послідовності UML, яка детально описує функціонування системи розпізнавання емоцій. Ця діаграма доповнює попередню, надаючи чітке уявлення про взаємодію між ключовими компонентами та даними протягом процесу розпізнавання емоцій.

Під час процесу розпізнавання емоцій спочатку завантажується зображення користувача, яке система зберігає у сховищі для подальшого використання, після чого воно проходить підготовку, конвертацію та оптимізацію. Далі, модель розпізнавання емоцій завантажується, стаючи доступною для системи. Система сканує зображення для виявлення та виділення облич, які потім масштабуються до одного розміру. За допомогою згорткових та повністю зв'язаних нейронних мереж аналізуються обличчя, визначаючи емоції, які вони виражають. Результати розпізнавання відображаються на екрані разом із відповідними зображеннями.

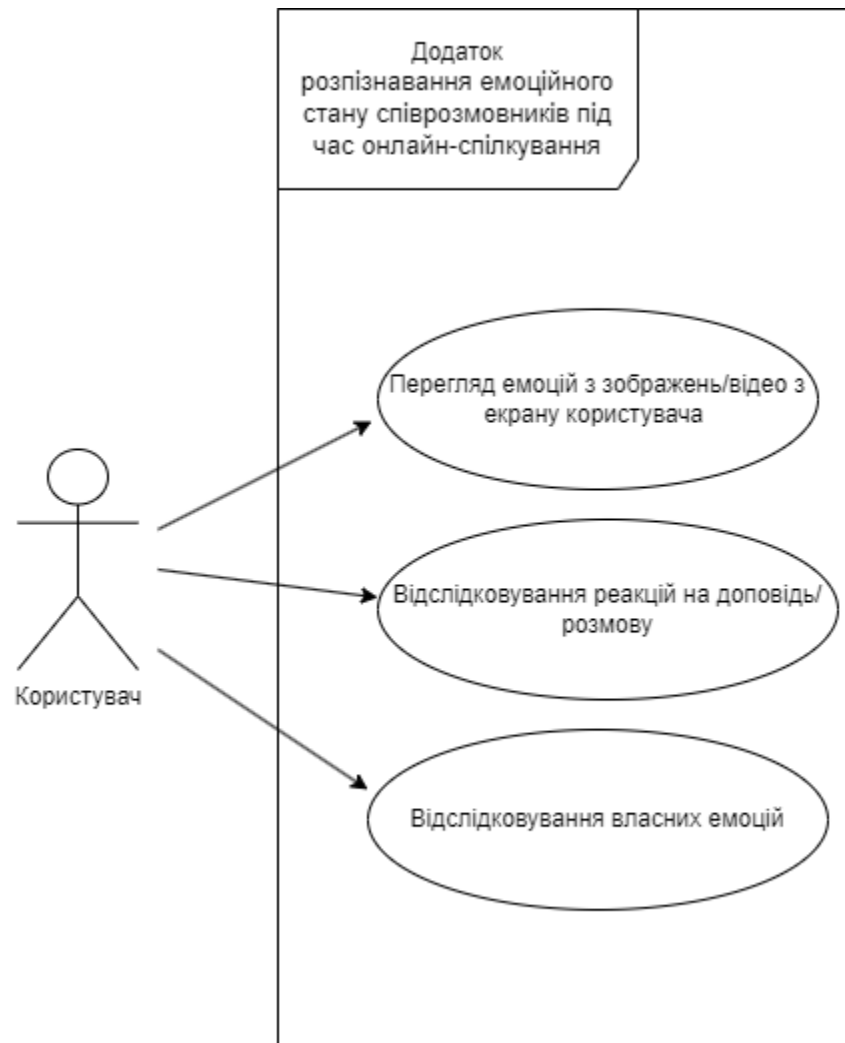


Рисунок 2.3 – Діаграма варіантів використання



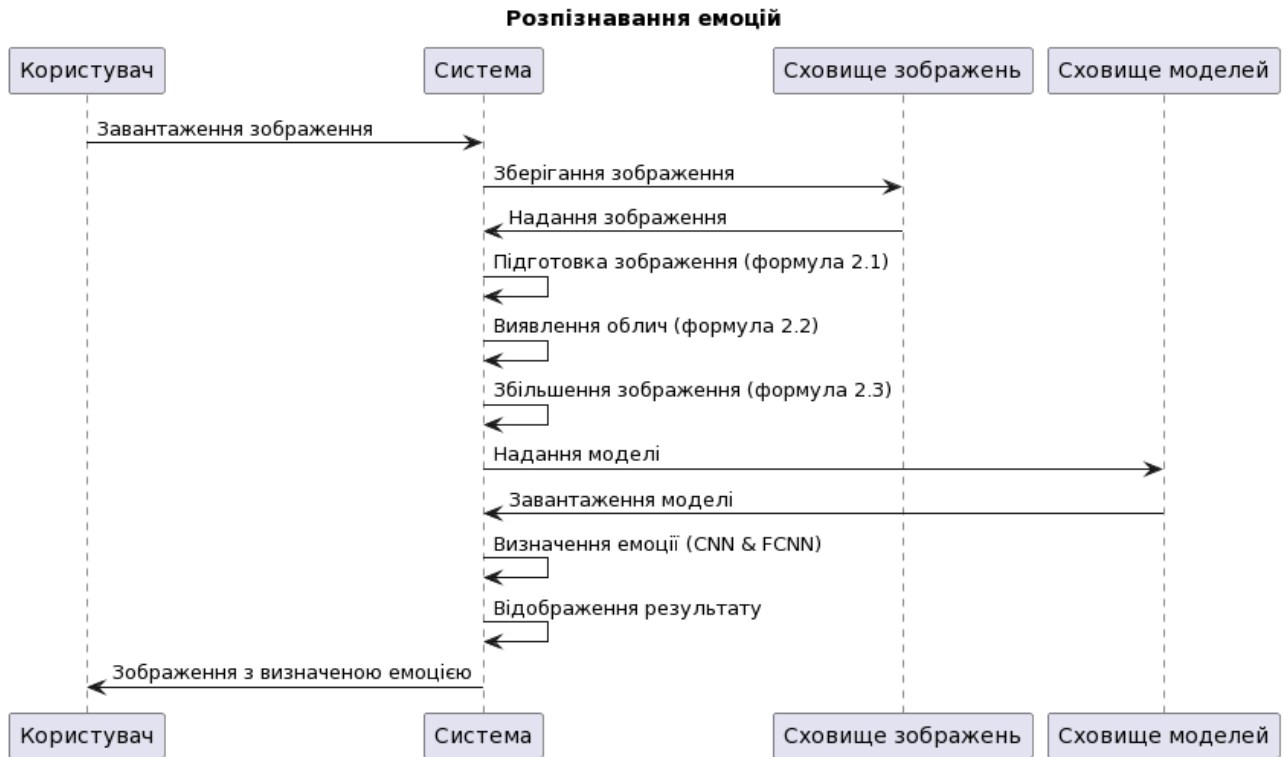


Рисунок 2.4 – Діаграма послідовності розпізнавання емоцій

### 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

Розпочнемо розробку інформаційної системи розпізнавання емоцій із створення згорткової нейронної мережі

```
emotion_model = Sequential()

emotion_model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(units=1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(units=7, activation='softmax'))
emotion_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Рисунок 3.1 – Структура згорткової нейронної мережі

Розберемо кожен шар згорткової мережі:

1. Перший шар – згортковий шар Conv2D, він аналізує зображення, проводить його через фільтри та запам'ятовує унікальні риси зображень з набору даних, за якими в подальшому буде відбуватися класифікація.

2. Другий шар – pooling layer (MaxPool2D()), він зменшує кількість параметрів, які були отримані згортковими шарами Conv2D, прискорюючи роботу нейронної мережі без від'ємного впливу на точність.

3. Dropout layer – додають до згорткової мережі для запобігання перенавчання (overfitting), він випадковим чином позбувається декількох нейронів з минулого шару в відсотках, заданих користувачем.

4. Flatten layer – зводить зображення з двовимірного розміру в одновимірний.

5. Dense layer – звичайний шар нейронної мережі, який аналізує зображення.

Після формування будови нейронної мережі, задаємо параметри компіляції, які підходять до нашого набору даних для навчання.

Вказуємо кількість циклів, крок між кожним зображенням

```
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.01,
    decay_steps=10000,
    decay_rate=0.9)
optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule)

# Train the neural network/model
emotion_model_info = emotion_model.fit_generator(
    train_generator,
    steps_per_epoch=28709 // 64,
    epochs=35,
    validation_data=validation_generator,
    validation_steps=7178 // 64)
```

Рисунок 3.2 – Параметри компіляції нейронної мережі

Процес навчання згортової нейронної мережі зображено на рисунку 3.3:

```
Epoch 27/35
448/448 [=====] - 311s 694ms/step - loss: 0.5680 - accuracy: 0.7909 - val_loss: 1.1362 - val_accuracy: 0.6122
Epoch 30/35
448/448 [=====] - 310s 691ms/step - loss: 0.5531 - accuracy: 0.7938 - val_loss: 1.1607 - val_accuracy: 0.6169
Epoch 31/35
448/448 [=====] - 312s 695ms/step - loss: 0.5442 - accuracy: 0.8009 - val_loss: 1.1546 - val_accuracy: 0.6210
Epoch 32/35
448/448 [=====] - 316s 705ms/step - loss: 0.5392 - accuracy: 0.8003 - val_loss: 1.1313 - val_accuracy: 0.6157
Epoch 33/35
448/448 [=====] - 310s 693ms/step - loss: 0.5273 - accuracy: 0.8078 - val_loss: 1.1870 - val_accuracy: 0.6225
Epoch 34/35
448/448 [=====] - 310s 693ms/step - loss: 0.5177 - accuracy: 0.8104 - val_loss: 1.1856 - val_accuracy: 0.6169
Epoch 35/35
448/448 [=====] - 310s 692ms/step - loss: 0.5125 - accuracy: 0.8116 - val_loss: 1.1687 - val_accuracy: 0.6251

Process finished with exit code 0
```

Рисунок 3.3 – Успішне завершення навчання нейронної мережі

Для тренування моделі було використано 28,709 зображень, що відображають різні людські емоції, розділені на 7 класів універсальних емоцій.

Детальні дані по кожному класу емоцій наведено в таблиці 3.1. Для оцінки ефективності навчання згорткової моделі використовувалися тестові дані набору FER2013, що включають у себе 7180 зображень. Подробиці щодо кожного класу емоцій також представлені у таблиці 3.1.

Таблиця 3.1 – Навчальні та тестові дані для моделі розпізнавання емоцій

Емоція	Кількість зображень у навчальному наборі	Кількість зображень у тестовому наборі
Злість	3995	958
Сум	4830	1247
Нейтральний	4965	1233
Здивування	3171	831
Страх	4097	1024
Огида	436	111
Щастя	7215	1774

Результати ефективності моделі після 25 циклів тренування наступні:

- Середня абсолютна похибка (MAE): 0.9703
- Точність (Accuracy): 0.6208

На рисунку 3.4 на графіку показано траєкторію зміни точності, під час тренування нейронної моделі на навчальному наборі даних, та зміну під час перевірки на тестовому наборі даних. що демонструє, як змінювалася точність моделі під час тренування на навчальному наборі даних. Під час первинного навчання моделі, точність постійно збільшувалась, але під час використання тренувального набору даних, в якому різноманітність зображень є більшою, точність поступово знизилася до значення 0.62, це означає, що модель вірно класифікує 62% прикладів з навчального набору даних.

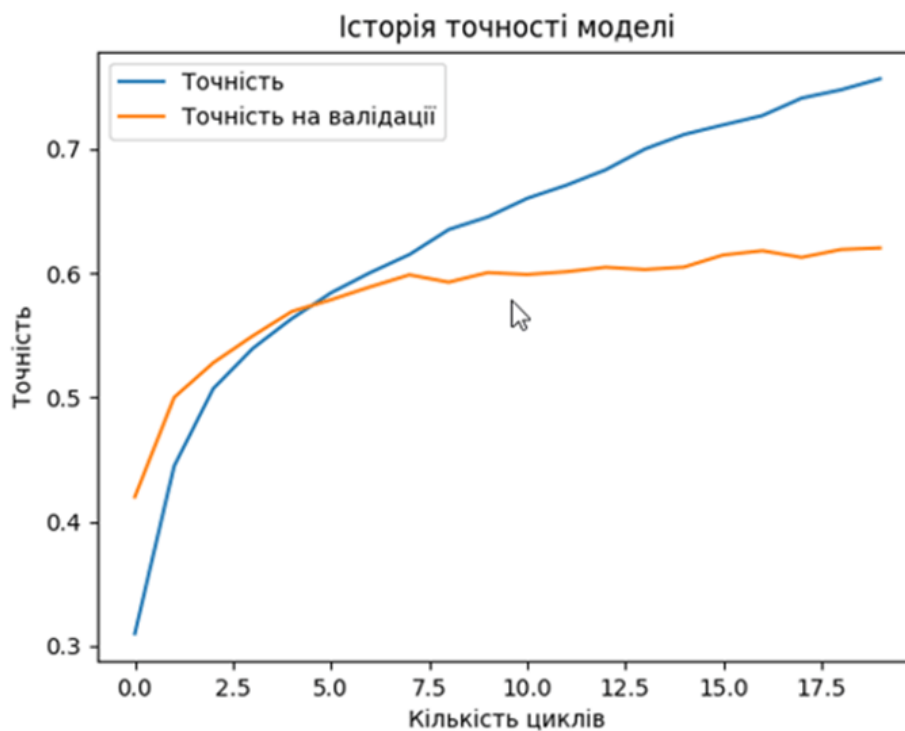


Рисунок 3.4 – Точність навчання згорткової нейронної мережі (25)

На рисунку 3.5 наведено графік точності класифікації даних протягом 50 циклів навчання. Точність навченої моделі не зазнала суттєвих змін.

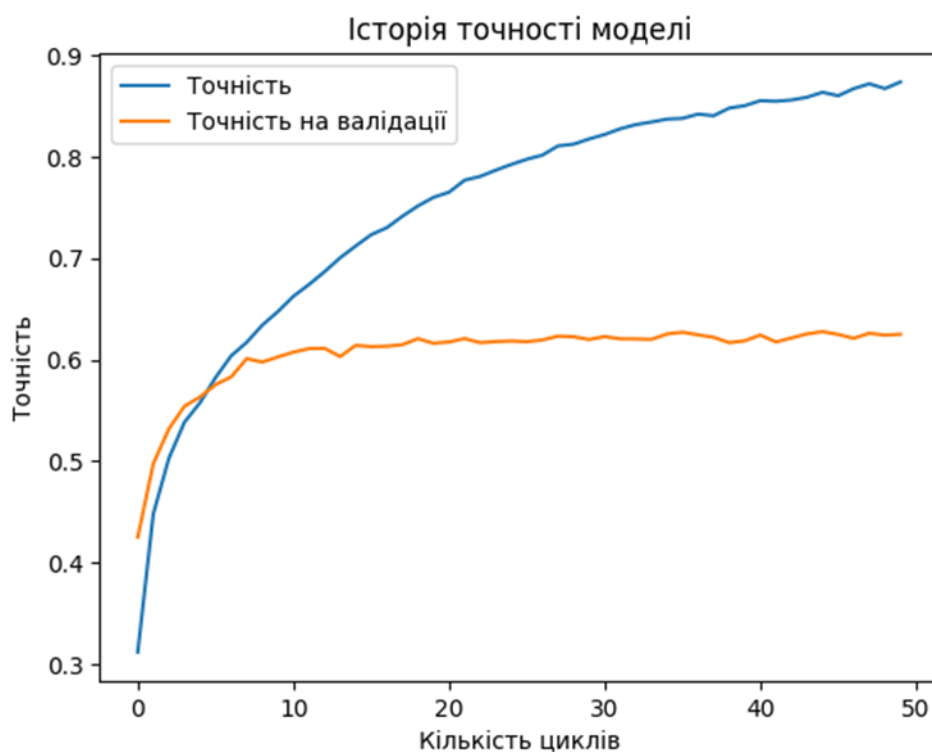


Рисунок 3.5 – Точність навчання згорткової нейронної мережі (50)

Загальна схема навченої згорткової нейромережі представлена на рисунку 3.6:

```
emotion_model = Sequential()

emotion_model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48, 1)))
emotion_model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(units=1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(units=7, activation='softmax'))
emotion_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Рисунок 3.6 – Будова згорткової нейромережі

Також виявлено декілька закономірностей: точність розпізнавання емоції «Огида» є низькою через обмежену кількість зображень, і їх часто помилково класифікують як інші емоції, тому рекомендується розширити набір даних; каскади Хаара можуть помилково ідентифікувати обличчя, що можна виправити за допомогою бібліотеки dlib, яка має високу точність; зниження яскравості зображень призводило до класифікації їх як «Страх», а збільшення контрастності - як «Здивування» або «Радість»; зменшення якості зображень не мало однозначного впливу на класифікацію, що вказує на здатність нейронної мережі адаптуватися до низької якості зображень.

Побудувавши нейромережу, почнемо розроблювати функцію трансляції екрану використовуючи бібліотеку monitorscreensaver(mss).

Загальну функцію трансляції екрану показано на рисунку 3.7:

```
def capture_screen(self):
    if self.current_mode == "screen":
        with mss.mss() as sct:
            monitor = {"top": 0, "left": 0, "width": 800, "height": 600}
            img = np.array(sct.grab(monitor))
```

Рисунок 3.7 – Функція захоплення екрану

Функція, що виконує розпізнавання обличчя на екрані зображена на рисунку 3.8:

```
def process_frame(self, frame):
    # Конвертація зображення в сірий колір для знаходження облич
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Знаходження облич
    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

    # Обробка кожного знайденого обличчя
    for (x, y, w, h) in faces:
        face_gray = gray_frame[y:y + h, x:x + w] # Відтинки сірого для обробки

        emotion_labels = ["Angry", "Disgusted", "Fearful", "Happy", "Neutral", "Sad", "Surprised"]
        face_gray_resized = cv2.resize(face_gray, dsize=(48, 48))
        face_gray_normalized = face_gray_resized / 255.0
        face_gray_reshaped = np.reshape(face_gray_normalized, newshape=(1, 48, 48, 1))
        predicted_emotion = emotion_model.predict(face_gray_reshaped)
        predicted_emotion_label = emotion_labels[np.argmax(predicted_emotion)]

        # Відображення результатів
        cv2.putText(frame, predicted_emotion_label, org=(x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.9, color=(36, 255, 12), thickness=2)
        self.emotions_screen_reg.append(predicted_emotion_label)

    # Оновлення QLabel з передбаченою емоцією
    self.update_image(frame)
```

Рисунок 3.8– Функція process\_frame

Демонстрацію роботи навченої моделі на даних, отриманих за допомогою трансляції екрану зображено на рисунку 3.9:

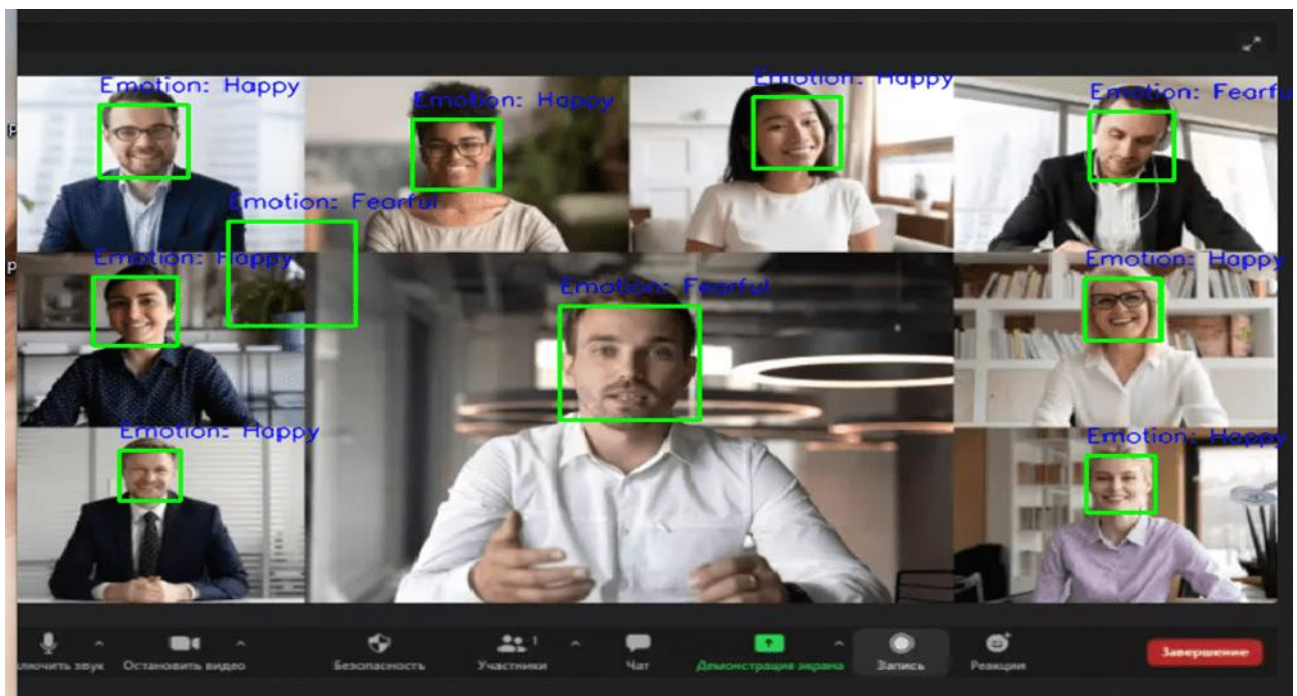


Рисунок 3.9 – Трансляція екрану з розпізнаванням облич з мультимедіа(фото)

Далі приступимо до розробки інтерфейсу додатку, в програмному середовищі Qt Designer. Демонстрація створеного макету додатку зображено на рисунку 3.10:

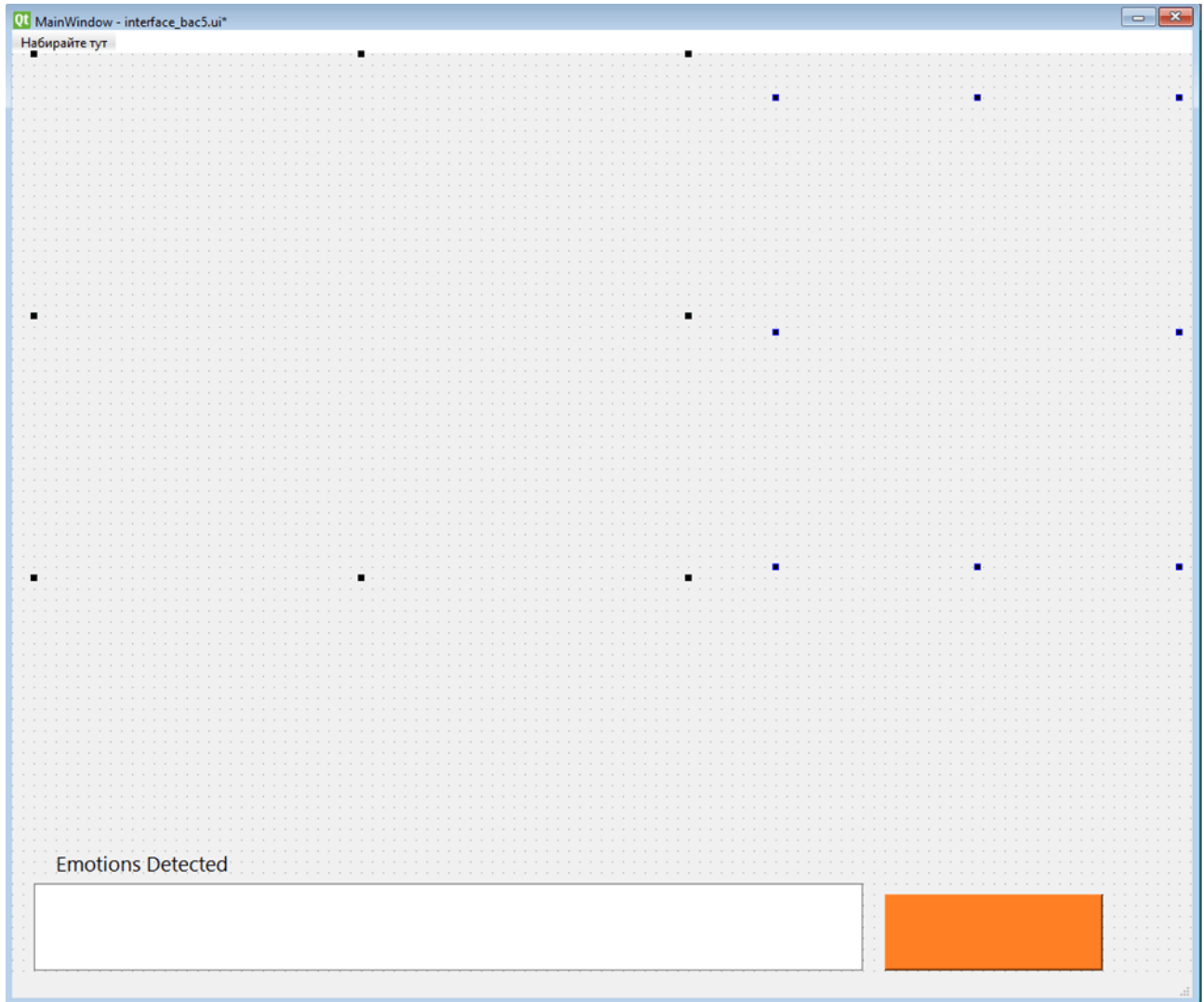


Рисунок 3.10 – Макет програмного додатку, створений в Qt Designer

Інтерфейс додатку складається з таких елементів:

- QLabel screen\_thread – Поле для виведення трансляції екрану.
- QLabel camera\_thread – Поле для виведення зображення з веб-камери користувача.
- QTextEdit emotion\_screen\_reg – текстове поле для реєстрації зафіксованих емоцій.



– QPushButton load\_screen\_button – кнопка для вимкнення/увімкнення камери.

Далі конвертуємо файл інтерфейсу .ui в файл коду .py.

Перелік параметрів головного класу інтерфейсу Ui\_EmotionRecognition зображено на рисунку 3.11:

```
class Ui_EmotionRecognition(object):
    def setupUi(self, EmotionRecognition):
        EmotionRecognition.setObjectName("EmotionRecognition")
        EmotionRecognition.resize(1082, 887)
        EmotionRecognition.setStyleSheet("#tabWidget{\n"
"background-color: #FFFFFF\n"
"}\n"
"\n"
"QPushButton {\n"
"    background-color: #ff7f24; /* Оранжевий колір фону кнопки */\n"
"    color: #000000; /* Сірий колір тексту кнопки */\n"
"}\n"
"\n"
"QLineEdit {\n"
"    background-color: #D3D3D3; /* Світло-сірий колір фону поля введення */\n"
"    color: #FFA500; /* Оранжевий колір тексту поля введення */\n"
"}")

        self.centralwidget = QtWidgets.QWidget(parent=EmotionRecognition)
        self.centralwidget.setObjectName("centralwidget")
        self.screen_thread = QtWidgets.QLabel(parent=self.centralwidget)
        self.screen_thread.setGeometry(QtCore.QRect(20, 0, 600, 480))
        self.screen_thread.setMinimumSize(QtCore.QSize(600, 400))
        self.screen_thread.setMaximumSize(QtCore.QSize(640, 480))
        self.screen_thread.setFrameShape(QtWidgets.QFrame.Shape.NoFrame)
        self.screen_thread.setText("")
```

Рисунок 3.11 – Макет програмного додатку, створений в Qt Designer

Далі потрібно інтегрувати модулі розпізнавання емоцій та трансляції зображень.

Результат роботи наведено на рисунку 3.12

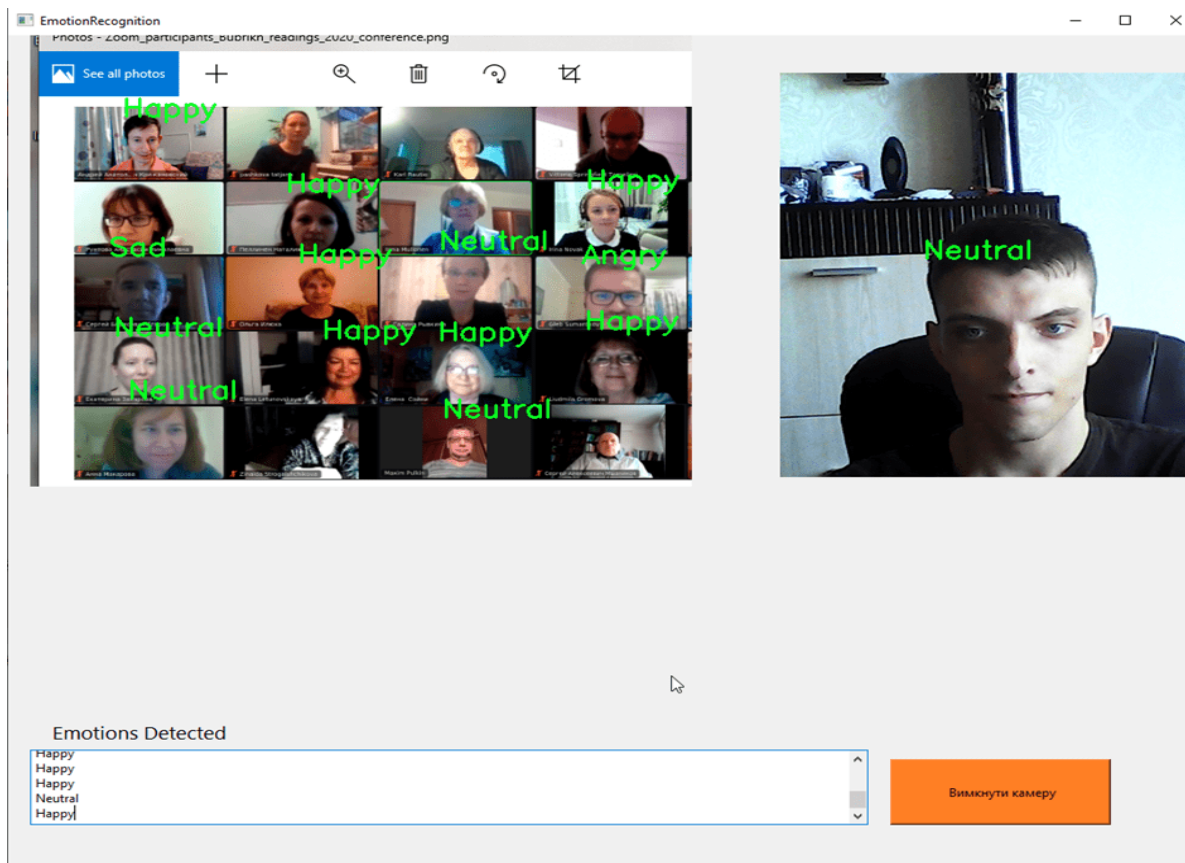


Рисунок 3.12 – Функціональне вікно додатку

Розробивши код додатку, потрібно запакувати його в виконуваний .exe файл, для звичайного користувача, для цього використаємо бібліотеку Pyinstaller[22].

Після успішної компіляції отримуємо готовий .exe файл нашого додатку, список створених файлів додатку зображено на рисунку 3.13:

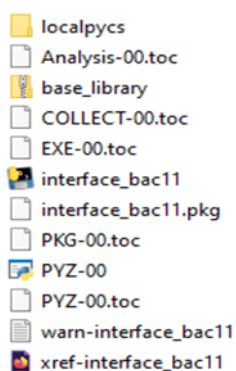


Рисунок 3.13 – Скомпільовані файли проекту в додаток формату .exe

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було проведено детальний аналіз предметної області машинного навчання з розпізнавання емоцій людей. Досліджено існуючі підходи та методи, що використовуються для автоматичного розпізнавання емоцій в онлайн-спілкуванні. Зокрема, розглянуто технології, що базуються на аналізі виразу обличчя. Було виявлено, що останні досягнення у сфері згорткових нейронних мереж значно підвищили точність і надійність таких систем.

Досліджено вплив різних параметрів на точність розпізнавання емоцій. Зокрема, аналізувались фактори, такі як якість зображення, різноманітність емоційних виразів у тренувальних даних, а також складність та глибина нейронних мереж.

Розроблена інформаційна технологія розпізнавання емоцій має потенціал в значній мірі покращити онлайн-спілкування, роблячи його більш індивідуалізованим, взаємодійним та продуктивним. Така технологія здатна надавати миттєвий зворотний зв'язок про емоційний стан співрозмовників, що дозволяє адаптувати тон і зміст повідомлень у реальному часі.

Проте для успішного впровадження таких технологій необхідно дотримуватись чітких правил та рекомендацій, які гарантують етичне збирання та використання даних користувачів. Особливу увагу слід приділяти питанням конфіденційності та захисту персональної інформації. Використання систем розпізнавання емоцій повинно супроводжуватися інформуванням користувачів про збір та обробку їхніх даних, а також забезпеченням їхньої згоди на такі дії.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. M. Imani and G. A. Montazer, “A survey of emotion recognition methods with emphasis on E-Learning environments,” *Journal of Network and Computer Applications*, vol. 147. 2019. doi: 10.1016/j.jnca.2019.102423.
2. R. Karani and S. Desai, “Review on Multimodal Fusion Techniques for Human Emotion Recognition,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022, doi: 10.14569/IJACSA.2022.0131035.
3. D. Dresvyanskiy, E. Ryumina, H. Kaya, M. Markitantov, A. Karpov, and W. Minker, “End-to-End Modeling and Transfer Learning for Audiovisual Emotion Recognition in-the-Wild,” *Multimodal Technologies and Interaction*, vol. 6, no. 2, 2022, doi: 10.3390/mti6020011.
4. A. Handa, R. Agarwal, and N. Kohli, “Audio-Visual Emotion Recognition System Using Multi-Modal Features,” *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 15, no. 4, 2021, doi: 10.4018/IJCINI.20211001.0a34.
5. Facial expression recognition software | FaceReader. Facial expression recognition software | FaceReader. URL: <https://www.noldus.com/facereader> (date of access: 30.04.2024).
6. Happy? Sad? Angry? This Microsoft tool recognizes emotions in pictures - The AI Blog. The AI Blog. URL: <https://blogs.microsoft.com/ai/happy-sad-angry-this-microsoft-tool-recognizes-emotions-in-pictures/> (date of access: 30.04.2024).
7. Face Analysis: Age, Gender & Emotion Recognition - Visage Technologies. Visage Technologies. URL: <https://visagetech.com/face-analysis/> (date of access: 30.04.2024).
8. Що таке функціональні вимоги: приклади, визначення, повний посібник - Рішення Visure. *Visure Solutions*. URL: <https://visuresolutions.com/uk/блог/функціональні-вимоги/> (date of access: 05.05.2024).

9. M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electronics (Switzerland)*, vol. 10, no. 9, 2021, doi: 10.3390/electronics10091036.
10. M. Sahu and R. Dash, "A survey on deep learning: Convolution neural network (cnn)," in *Smart Innovation, Systems and Technologies*, 2021. doi: 10.1007/978-981-15-6202-0\_32
11. M. Z. Alom et al., "A state-of-the-art survey on deep learning theory and architectures," *Electronics (Switzerland)*, vol. 8, no. 3, 2019. doi: 10.3390/electronics8030292.
12. "Building a Convolutional Neural Network (CNN) Model for Image classification. URL: <https://becominghuman.ai/building-a-convolutional-neural-network-cnn-model-for-image-classification-116f77a7a236>."
13. P.-L. Carrier and A. Courville, "The Facial Expression Recognition 2013 (FER-2013) Dataset," *Wolfram Data Repository*.
14. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 2010. doi: 10.1109/CVPRW.2010.5543262.
15. S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, and J. F. Cohn, "DISFA: A spontaneous facial action intensity database," *IEEE Trans Affect Comput*, vol. 4, no. 2, 2013, doi: 10.1109/T-AFFC.2013.4.
16. P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *Int J Comput Vis*, vol. 57, no. 2, 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.
17. I. В. Іванішин, "Використання методів Віоли-Джонса і к найближчих сусідів для детекції та розпізнавання обличч," *Міжнародний науковий журнал "Інтернаука"*, vol. 11, 2022.
18. Guide | TensorFlow Core. *TensorFlow*. URL: <https://www.tensorflow.org/guide?hl=en> (date of access: 05.05.2024).

19. Qt Designer Manual. *Qt Documentation / Home*. URL: <https://doc.qt.io/qt-6/qtdesigner-manual.html> (date of access: 05.05.2024).
20. К. В. Юдкова, “Особливості визначення поняття ‘інформаційні технології,’” *Інформація і право*, vol. 1, no. 13, pp. 63–67, 2015.
21. Draft Federal Information and Processing Standards Publication 183, Announcing the Standard for INTEGRATION DEFINITION FOR FUNCTION MODELING (IDEF0), no. December 21. 1993.
22. Fitzpatrick M. Packaging PyQt6 applications for Windows, with PyInstaller & InstallForge. *Python GUIs*. URL: <https://www.pythonguis.com/tutorials/packaging-pyqt6-applications-windows-pyinstaller/> (date of access: 26.05.2024).
23. “GDPR,” <https://gdpr.eu/article-83-conditions-for-imposing-administrative-fines>

## ДОДАТОК А

**ТЕХНІЧНЕ ЗАВДАННЯ**  
**на створення додатку**  
**«Інформаційна технологія розпізнавання**  
**емоційного стану співрозмовників**  
**під час онлайн-спілкування»**

**ПОГОДЖЕНО:**

Старший викладач  
кафедри інформаційних технологій

\_\_\_\_\_ Бойко О.В.

Студент групи ІТ-03\_\_\_\_\_

\_\_\_\_\_ Суровцев А.В.

**Суми 2024**

## **1 Призначення й мета створення додатку розпізнавання емоційного стану співрозмовників під час онлайн-спілкування**

### **1.1 Призначення додатку**

Розроблюваний додаток призначений для кращого розуміння залученості та емоційного стану співрозмовників під час онлайн спілкування, це може допомогти знайти найкращі підходи до подачі матеріалу та засвоєння його студентами.

### **1.2 Мета створення додатку розпізнавання емоційного стану співрозмовників під час онлайн-спілкування**

Метою даної роботи є покращення можливості аналізу емоційного стану людей під час онлайн-спілкування.

### **1.3 Цільова аудиторія**

Цільовою аудиторією мого додатку є кожна людина, яка цікавиться своїми онлайн-комунікаціями або бажає покращити їх.

## **2 Вимоги до додатку**

### **2.1 Вимоги до інформаційної технології в цілому**

### **2.2 Вимоги до структури й функціонування додатку**

Додаток має бути доступним широкому колу пристроїв та бути не вибагливим до потужності комп'ютера. Мова програмування Python дозволяє скомпілювати код під всі популярні операційні системи такі як Windows, MacOS та Linux.

### **2.3 Вимоги до програмного та апаратного забезпечення**

Додаток повинен використовувати останні версії бібліотек для машинного навчання та ефективно використовувати пам'ять пристрою користувачів.

### **2.4 Вимоги до збереження інформації**

Оскільки технологія розпізнавання емоцій на обличчях включає обробку біометричних даних, що класифікуються як дані спеціальної категорії згідно з Загальним регламентом захисту даних Європейського Союзу (GDPR) [23]. Згідно GDPR, обробка даних спеціальної категорії забороняється, якщо не виконуються



певні умови. Тому використання такої технології в електронному навчанні повинно відповідати цим правилам, включаючи згоду від осіб, чії дані обробляються, та захист цих даних. Уся інформація не зберігається після використання.

## **2 Структура додатку розпізнавання емоційного стану співрозмовників під час онлайн-спілкування**

Структура додатку являє собою вікно інтерфейсу з двома вкладками:

Трансляція екрану – у вікні QLabel відбувається демонстрація екрану користувача та розпізнавання емоцій при знаходженні обличчя.

Веб-камера – у вікно QLabel виводиться зображення з веб-камери та розпізнається емоція користувача

### **3.1 Дизайн та структура додатку розпізнавання емоційного стану співрозмовників під час онлайн-спілкування**

Оформлення та структура додатку повинна бути інтуїтивно зрозумілою користувачу та видавати точні результати значень емоцій.

Розташування елементів на головній сторінці додатку зображено на рисунку А.1



Рисунок А.1 – Схема головної сторінки

Карта додатку зображена на рисунку А.2.

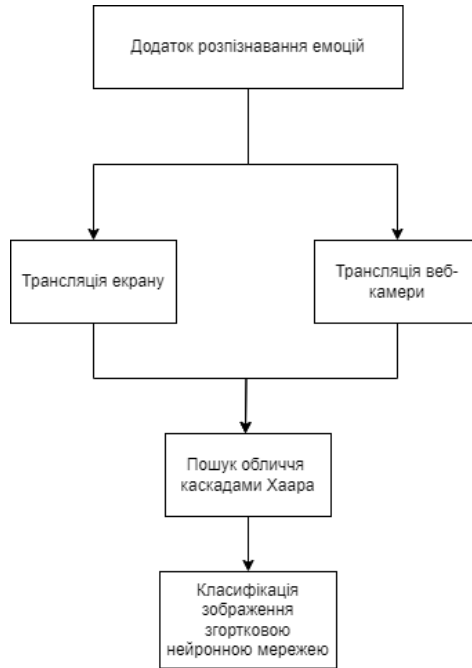


Рисунок А.2 – Карта додатку

На зображення екрану додається текст з назвою емоції. Зображення екрану з визначеною емоцією виводиться на екран.

Список емоцій:

*emotionlabels* =

*["Angry", "Disgusted", "Fearful", "Happy", "Neutral", "Sad", "Surprised"]*

Блок-схема алгоритму розпізнавання емоцій на обличчі наведена на рисунку А.3



Рисунок А.3 – блок-схема алгоритму розпізнавання емоцій на обличчі

**4. Склад і зміст робіт зі створення додатку розпізнавання емоційного стану співрозмовників під час онлайн-спілкування** Докладний опис етапів роботи зі створення додатку розпізнавання емоцій таблиці 1.

Таблиця 1 – Етапи створення додатку розпізнавання емоцій

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Розробка нейронної мережі	4 дні
2	Розробка модулю трансляції екрану	19 днів
3	Розробка інтерфейсу програми	8 днів
4	Підключення нейронної мережі до інтерфейсу та його модулів	7 днів
5	Тестування	7 днів
6	Оформлення зовнішнього вигляду інтерфейсу	2 дні
7	Компіляція додатку	1 день
8	Тестування	7 днів
9	Написання супровідної документації	8 днів
10	Реліз додатку	1 день
	Загальна тривалість робіт	64 дні

## **ДОДАТОК Б ПЛАНУВАННЯ РОБІТ**

Метою роботи є розробка та впровадження інформаційної технології розпізнавання емоційного стану співрозмовників під час онлайн-спілкування.

Для досягнення мети проєкту необхідно виконати наступні задачі:

- провести порівняльний аналіз існуючих підходів до автоматичного розпізнавання емоцій з метою визначення їхніх переваг та обмежень у контексті застосування у системах навчання;
- розробити алгоритм та модель автоматичного розпізнавання емоцій, що базуються на виразі обличчя, враховуючи сучасні досягнення в галузі машинного навчання;
- запропонувати інформаційну технологію розпізнавання емоційного стану співрозмовників під час онлайн-спілкування;

ідентифікувати можливі занепокоєння користувачів щодо конфіденційності та захисту їх персональних даних під час використання інформаційної технології.

### **Б.1 Деталізація мети проєкту методом SMART**

Метою даної роботи є покращення можливості аналізу залученості співрозмовників під час онлайн-спілкування.

Отже, можемо сформулювати мету нашого проєкту за цими п'ятьма факторами. Результати наведені у таблиці Б.1.

### **Б.2 Планування змісту робіт та структури виконавців**

Планування змісту робіт (WBS) – являє собою графічний вигляд елементів проєкту розміщених в ієрархічному порядку. WBS є дуже гарним способом організувати командну роботу та легко оцінити кількість часу необхідного як для виконання певного етапу розробки, так і проєкту в цілому.

На найвищому (першому) рівні розміщений продукт проєкту. Основні дії та заходи, що забезпечують досягнення мети проєкту, зафіксовані на другому рівні декомпозиції. Декомпозиція робіт виконується до тих пір, поки вони не стануть

елементарними (простими). Елементарні роботи – це дії, які мають однозначний чіткий результат, на які призначена відповідальному одна конкретна особа, для якої можна обчислити витрати праці і тривалість виконання

На рисунку Б.1 представлено WBS з розробки web-додатку організації діяльності розробки додатку розпізнавання емоцій.

На рисунку Б.2 представлено організаційну структуру планування проєкту. Список виконавців, що функціонують в проєкті, описано в таблиці Б.2.

Таблиця Б.1 – Формалізація мети за технологією SMART

Specific	Розробка та впровадження інформаційної технології розпізнавання емоцій для оцінки залучення та емоційного стану співрозмовників під час онлайн-спілкування
Measurable	Підвищення точності розпізнавання емоцій співрозмовників під час онлайн-комунікації та їх залученості в навчальний процес
Achievable	Наявне затвержене технічне завдання від замовника та команда програмістів і тестувальників, які будуть виконувати поставлене завдання, 3 комплекти комп'ютерної техніки, безкоштовне ПЗ для розробки PyCharm, менеджер, бібліотеки штучного інтелекту TensorFlow, OpenCV.
Relevant	Підвищення рівня комунікації між учасниками онлайн-заходів
Time-bound	Наявний конкретний термін – до кінця 4 курсу (06 червня 2024 р.).

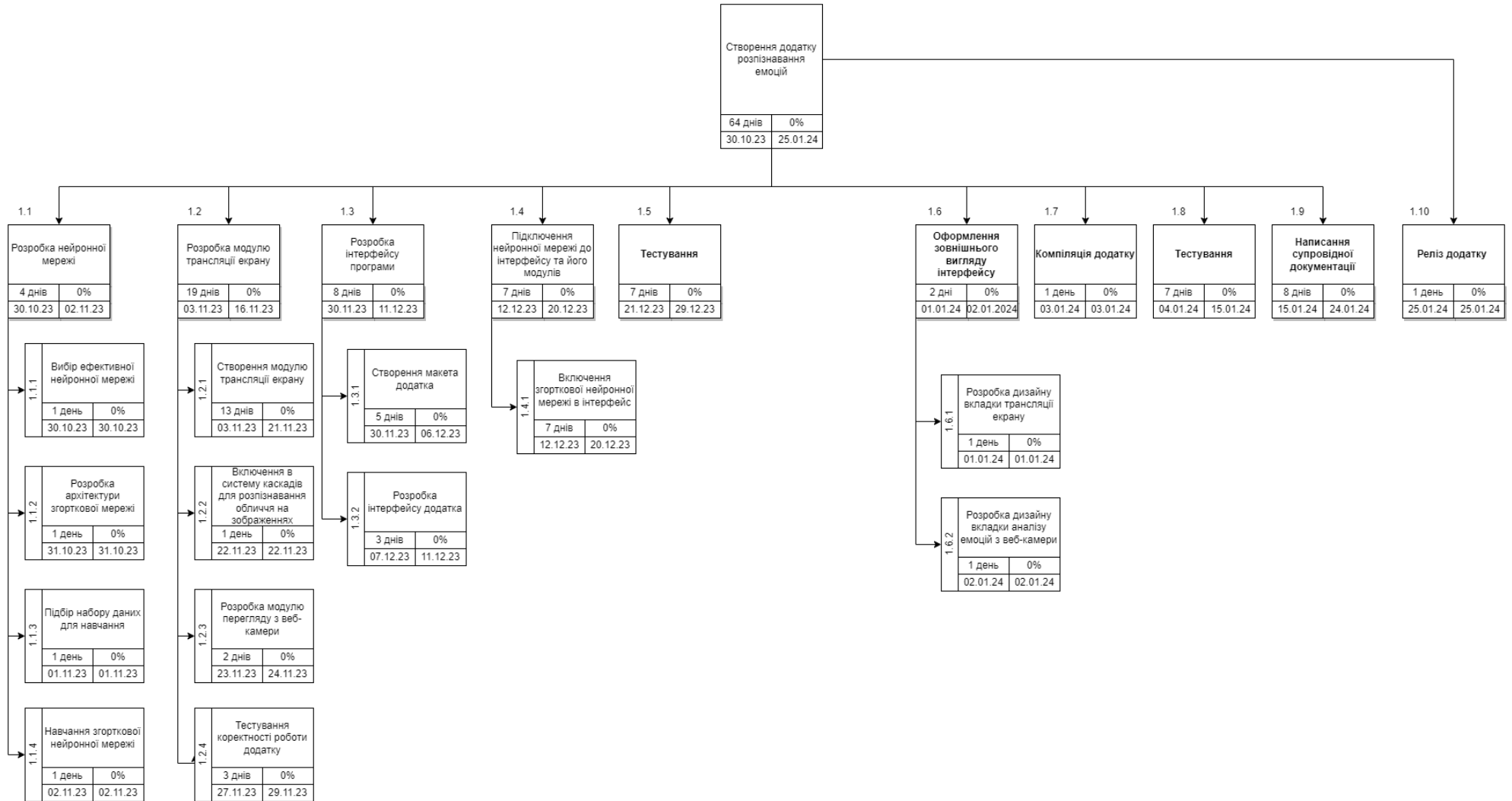


Рисунок Б.1 – WBS-структура робіт проєкту

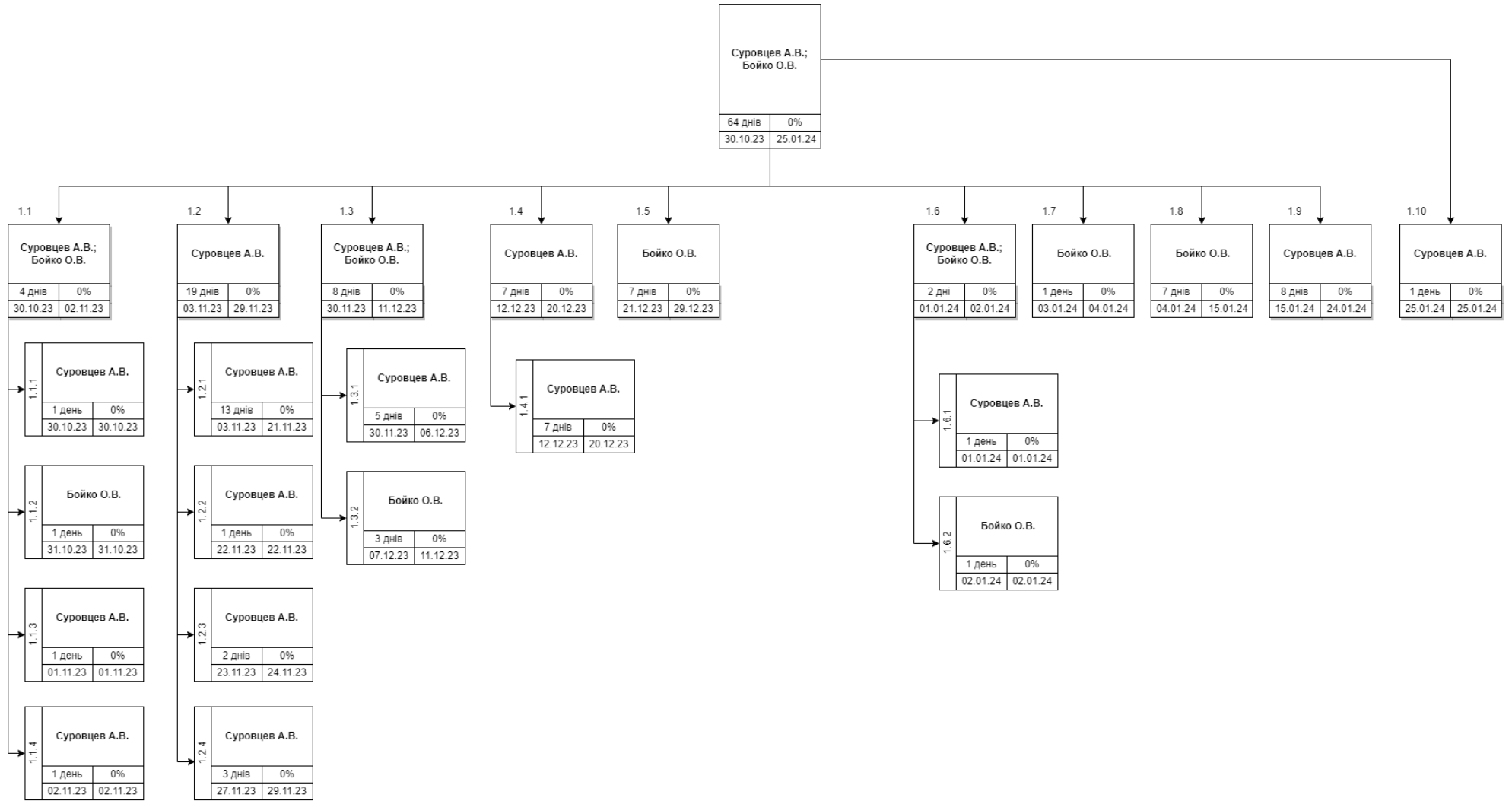


Рисунок Б.2 – OBS-структура робіт проект



Таблиця Б.2 – Виконавці проєкту

Роль	ПІБ	Проектна роль
Розробник	Суровцев А.В	Виконує front-end та back-end розробку
Проектувальник	Суровцев А.В	Виконує проектування бази даних та розробляє структуру web-додатку
Тестувальник	Суровцев А.В	Відповідає за тестування функціоналу та дизайну web-додатку.
Керівник проєкту	Бойко О.В.	Формує завдання на розробку проєкту.

### Б.3 Діаграма Ганта

Діаграма Ганта. Побудова календарного графіку (діаграми Ганта) є одним з важливих етапів планування проєкту, що виглядає як розклад виконання робіт з реальним розподілом дат. Завдяки йому можна отримати достовірне уявлення про тривалість процесів з обмеженнями у ресурсах, урахуванням вихідних днів та свят. Календарний графік проєкту представлено на рисунку Б.3.



#### Б.4 Управління ризиками проєкту

Під час виконання якісної оцінки ризиків треба визначити ризики, які мають бути усунені якнайшвидше. В залежності від ступеня важливості ризику – реагування буде відповідне. Наступним етапом є виконання кількісного оцінювання ризиків. Кількісне та якісне оцінювання можуть виконувати одночасно або окремо, що залежить від ступеня забезпечення проєкту. У таблиці Б.3 надано перелік ризиків даного проєкту. Результати оцінки ризиків надано у таблиці Б.4. Таблиця Б.5 представляє шкалу для класифікації ризиків за величиною впливу на проєкт та ймовірністю їх виникнення.

Таблиця Б.3 – Ризики проєкту

№	Назва (опис) ризику
1	Непорозуміння між розробником та замовником
2	Нечітке завдання на розробку
3	Відключення хостингу
4	Закриття магазину
5	Розробники захворіли
6	Низька кваліфікація розробників
7	Неоптимальний розподіл часу
8	Часте внесення змін у ТЗ

Таблиця Б.4 – Результати визначення ймовірності, впливу та рангу ризиків проекту

№	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Непорозуміння між розробником та замовником	0,1	0,2	0,02
2	Нечітке завдання на розробку	0,5	0,4	0,2
3	Поганий набір даних для навчання	0,1	0,2	0,02
4	Неефективна архітектура згорткової нейронної мережі	0,1	0,8	0,08
5	Розробники захворіли	0,2	0,7	0,14
6	Низька кваліфікація розробників	0,5	0,5	0,25
7	Неоптимальний розподіл часу	0,3	0,5	0,15
8	Часте внесення змін у ТЗ	0,45	0,5	0,25

Таблиця Б.5 – Шкала оцінювання ризиків за ймовірністю виникнення та величиною впливу

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Для того, щоб знизити негативний вплив ризиків на проект треба виконати планування реагування на них. До нього входить оцінка наслідків впливу на проект і розробка відповідних заходів. Аналіз виконується за показниками, які описані в

таблиці Б.4. У результаті планування заходів реагування на ризики проєкту було отримано матрицю ймовірності виникнення та впливу ризиків (таблиця Б.6). Зеленим кольором на матриці позначають прийнятні ризики, жовтим – виправдані, а червоним – недопустимі.

Таблиця Б.6 – Матриця ймовірності та впливу згідно проєкту

Ймовірність ризику(Й)	Вплив загрози (ризик)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9					
0,7					
0,5				R2(0,20)	R6(0,25)
0,3					R7(0,15)
0,1			R1(0,02), R3(0,02)		R4(0,08), R5(0,14)

Класифікація ризиків проєкту за рівнем, відповідно до отриманого значення індексу, представлена у таблиці Б.7. У таблиці Б.8 описано ризики та стратегії реагування на кожен із них.

Таблиця Б.7 - Шкала оцінювання ризику за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$0,005 < R < 0,05$	1,3
2	Виправдані	$0,05 < R < 0,14$	4,5,6
3	Недопустимі	$0,14 < R \leq 0,72$	2,7,8

Таблиця Б.8 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Open	Непорозуміння між розробником та замовником	Низька	Середній	3	<ol style="list-style-type: none"> <li>1. Налагодити гарні відносини між розробником та керівником.</li> <li>2. Дотримуватися ділового етикету спілкування.</li> <li>3. Створити комфортні умови для співпраці</li> </ol>	Попередження	При виявленні непорозуміння потрібно з'ясувати, що саме стало причиною непорозуміння обговорити її та створити здорову атмосферу в колективі
RS_2	Open	Нечітке завдання на розробку	Середня	Високий	6	<ol style="list-style-type: none"> <li>1. Конкретно і однозначно обговорити із замовником усі вимоги.</li> <li>2. Скласти глосарій для запобігання розбіжностей у розумінні слів та термінів</li> <li>3. Періодичний контроль замовником етапів роботи.</li> </ol>	Попередження	При виявленні невідповідностей деяких характеристик продукту заявленим вимогам потрібно уважно та чітко окреслити те, що було виконано невірно та зробити правки

Продовження таблиці Б.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_3	Open	Поганий набір даних для навчання	Низька	Середній	3	1. Перед вибором набору даних проаналізувати літературу з предметної області, подивитись на різні датасети та точність навчених на їх основі моделей.	Попередження	Замінити набір даних на інший.
RS_4	Open	Неефективна архітектура згорткової нейронної мережі	Низька	Високий	3	1.Проводити тести класифікації при побудові нейронної мережі, порівнюючи з найкращими та середніми результатами точності в даній області.	Попередження	При виявленні неефективної класифікації проаналізувати будову нейронної мережі та зібрати її з іншою конфігурацією фільтрів та шарів.

Продовження таблиці Б.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_5	Open	Розробники захворіли	Низька	Високий	3		Прийняття	
RS_6	Open	Низька кваліфікація розробників	Середня	Високий	6	1. Підбирати розробників через посередників або за рекомендацією знайомих-розробників, тих хто вже мав подібний проект. 2. Найняти кваліфіковану людину (HR), яка зможе гарно оцінити навички розробників та їх компетентність.	Попередження	При виявленні неякісно виконаної роботи, звільнити розробника і пошукати іншого, можливо на інших площадках, заплатити HR для оцінки компетентності нового кандидата.



Продовження таблиці Б.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_7	Open	Неоптимальний розподіл часу	Низька	Високий	3	<p>1.Сформувати дорожню карту розробки проекту з зазначенням конкретних дат, та враховуючи можливі затримки.</p> <p>2.Проводити щотижневі зустрічі з розробниками на яких обговорити спільне бачення та дати завершення критичних етапів проекту.</p>	Попередження	Замовник та розробники повинні зібратись та послухати ідеї один одного, знайти консенсус, після цього змінити строки розробки та/або підхід до неї/кількість бажаних функцій тощо.

Продовження таблиці Б.8

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_8	Open	Часте внесення змін у ТЗ	Середня	Високий	6	<p>1. Створити чіткий план розробки, з урахуванням потреб користувачів та замовника.</p> <p>2. Обговорити з розробниками їх ідеї та бачення кінцевого продукту.</p> <p>3. Сформувати чітке бачення кінцевого продукту та зрозуміло доносити його до команди розробників</p>	Попередження	Замовник та розробники повинні зібратись та послухати ідеї один одного, знайти консенсус, після цього змінити строки розробки та/або підхід до неї/кількість бажаних функцій тощо.

# ДОДАТОК Г

## Лістинг програми

### Interface\_bac

```
import sys

from PyQt6.QtGui import QImage, QPixmap

from PyQt6.QtWidgets import QMainWindow

from PyQt6 import QtCore, QtGui, QtWidgets

import cv2

import mss

import numpy as np

from tf_keras.models import model_from_json

json_file = open(r'C:\Users\Artem\Desktop\pythonProject\emotion_model1505.json')

loaded_model_json = json_file.read()

json_file.close()

emotion_model = model_from_json(loaded_model_json)

# Завантаження ваг з H5-файлу

emotion_model.load_weights(r"C:\Users\Artem\Desktop\pythonProject\emotion_model1505.h5")

print("Loaded model architecture and weights from disk")

# Load Haar cascade for face detection

face_cascade = cv2.CascadeClassifier(

r'C:\Users\Artem\Desktop\pythonProject\haarcascades/haarcascade_frontalface_default.xml')

class Ui_EmotionRecognition(object):

    def setupUi(self, EmotionRecognition):

        EmotionRecognition.setObjectName("EmotionRecognition")
```

```

EmotionRecognition.resize(1082, 887)

EmotionRecognition.setStyleSheet("#tabWidget{\n"
    "background-color: #FFFFFF\n"
    "}\n"
    "\n"
    "QPushButton {\n"
    "    background-color: #ff7f24; /* Оранжевий колір
фону кнопки */\n"
    "    color: #000000; /* Сірий колір тексту кнопки
*/\n"
    "}\n"
    "\n"
    "QLineEdit {\n"
    "    background-color: #D3D3D3; /* Світло-сірий
колір фону поля введення */\n"
    "    color: #FFA500; /* Оранжевий колір тексту поля
введення */\n"
    "}")

self.centralwidget = QtWidgets.QWidget(parent=EmotionRecognition)
self.centralwidget.setObjectName("centralwidget")
self.screen_thread = QtWidgets.QLabel(parent=self.centralwidget)
self.screen_thread.setGeometry(QtCore.QRect(20, 0, 600, 480))
self.screen_thread.setMinimumSize(QtCore.QSize(600, 400))
self.screen_thread.setMaximumSize(QtCore.QSize(640, 480))
self.screen_thread.setFrameShape(QtWidgets.QFrame.Shape.NoFrame)
self.screen_thread.setText("")
self.screen_thread.setObjectName("screen_thread")
self.emotions_screen_reg = QtWidgets.QTextEdit(parent=self.centralwidget)
self.emotions_screen_reg.setGeometry(QtCore.QRect(20, 760, 760, 80))
self.emotions_screen_reg.setMinimumSize(QtCore.QSize(760, 20))
self.emotions_screen_reg.setMaximumSize(QtCore.QSize(211, 311))
self.emotions_screen_reg.setObjectName("emotions_screen_reg")
self.labelscreen1 = QtWidgets.QLabel(parent=self.centralwidget)
self.labelscreen1.setGeometry(QtCore.QRect(40, 730, 171, 21))

```

```
font = QtGui.QFont()
font.setPointSize(14)
self.labelscreen1.setFont(font)
self.labelscreen1.setObjectName("labelscreen1")
self.load_screen_button = QtWidgets.QPushButton(parent=self.centralwidget)
self.load_screen_button.setGeometry(QtCore.QRect(800, 770, 200, 70))
self.load_screen_button.setObjectName("load_screen_button")
self.camera_thread = QtWidgets.QLabel(parent=self.centralwidget)
self.camera_thread.setGeometry(QtCore.QRect(700, 40, 370, 430))
self.camera_thread.setMinimumSize(QtCore.QSize(100, 100))
self.camera_thread.setMaximumSize(QtCore.QSize(640, 480))
self.camera_thread.setFrameShape(QtWidgets.QFrame.Shape.NoFrame)
self.camera_thread.setText("")
self.camera_thread.setObjectName("camera_thread")
EmotionRecognition.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(parent=EmotionRecognition)
self.menubar.setGeometry(QtCore.QRect(0, 0, 1082, 22))
self.menubar.setObjectName("menubar")
EmotionRecognition.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(parent=EmotionRecognition)
self.statusbar.setObjectName("statusbar")
EmotionRecognition.setStatusBar(self.statusbar)

self.retranslateUi(EmotionRecognition)
QtCore.QMetaObject.connectSlotsByName(EmotionRecognition)

self.timer = QtCore.QTimer()
self.camera_timer = QtCore.QTimer()
self.cap = cv2.VideoCapture(0)

self.timer.timeout.connect(self.capture_screen)
self.timer.start(24) # Захват кожні 24 мс
self.camera_timer.timeout.connect(self.update_camera_image)
```

```

self.camera_timer.start(24) # 24 мс

self.load_screen_button.clicked.connect(self.toggle_camera)

self.current_mode = "screen" # Можливі значення: "screen", "camera"

def toggle_camera(self):
    if self.cap.isOpened():
        self.current_mode = "screen"
        self.load_screen_button.setText("Увімкнути камеру")
        self.cap.release() # Вимкнення камери
        self.camera_thread.clear() # Очищення потоку камери
    else:
        # Перемикання на режим камери
        self.current_mode = "camera"
        self.camera_timer.start(24) # Запуск таймера камери
        self.load_screen_button.setText("Вимкнути камеру")
        if not self.cap.isOpened():
            self.cap = cv2.VideoCapture(0) # Вмикання камери, якщо вона ще не активна
            self.timer.start()
            self.current_mode = "screen"

def retranslateUi(self, EmotionRecognition):
    _translate = QtCore.QCoreApplication.translate
    EmotionRecognition.setWindowTitle(_translate("EmotionRecognition",
"EmotionRecognition"))
    self.labelscreen1.setText(_translate("EmotionRecognition", "Emotions Detected"))
    self.load_screen_button.setText(_translate("EmotionRecognition", " Вимкнути
камеру"))

def capture_screen(self):
    if self.current_mode == "screen":
        with mss.mss() as sct:

```

```

        monitor = {"top": 0, "left": 0, "width": 800, "height": 600}

        img = np.array(sct.grab(monitor))

        self.process_frame(img)

        self.update_image(img)

def update_image(self, img):

    # Convert the image format to RGB for QPixmap

    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    qimg = QImage(img_rgb.data, img_rgb.shape[1], img_rgb.shape[0],
QImage.Format.Format_RGB888)

    pixmap = QPixmap.fromImage(qimg)

    self.screen_thread.setPixmap(pixmap)

def process_frame(self, frame):

    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

    # Обробка кожного знайденого обличчя

    for (x, y, w, h) in faces:

        face_gray = gray_frame[y:y + h, x:x + w] # Відтинки сірого для обробки

        emotion_labels = ["Angry", "Disgusted", "Fearful", "Happy", "Neutral", "Sad",
"Surprised"]

        face_gray_resized = cv2.resize(face_gray, (48, 48))

        face_gray_normalized = face_gray_resized / 255.0

        face_gray_reshaped = np.reshape(face_gray_normalized, (1, 48, 48, 1))

        predicted_emotion = emotion_model.predict(face_gray_reshaped)

        predicted_emotion_label = emotion_labels[np.argmax(predicted_emotion)]

    # Відображення результатів

```

```

        cv2.putText(frame, predicted_emotion_label, (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12), 2)

        self.emotions_screen_reg.append(predicted_emotion_label)

        # Оновлення QLabel з передбаченою емоцією
        self.update_image(frame)

def update_camera_image(self):

    ret, frame = self.cap.read()

    if ret:

        # Конвертація зображення для відображення у QLabel
        gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        faces = face_cascade.detectMultiScale(gray_frame, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

        for (x, y, w, h) in faces:

            face_gray = gray_frame[y:y + h, x:x + w]

            emotion_labels = ["Angry", "Disgusted", "Fearful", "Happy", "Neutral",
"Sad", "Surprised"]

            face_gray_resized = cv2.resize(face_gray, (48, 48))
            face_gray_normalized = face_gray_resized / 255.0
            face_gray_reshaped = np.reshape(face_gray_normalized, (1, 48, 48, 1))
            predicted_emotion = emotion_model.predict(face_gray_reshaped)
            predicted_emotion_label = emotion_labels[np.argmax(predicted_emotion)]

            # Отображение результатов на face_color

            cv2.putText(frame, predicted_emotion_label, (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (36, 255, 12),

                2)

            # Конвертація зображення для відображення у QLabel
            rgb_image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

            qimg = QtGui.QImage(rgb_image.data, rgb_image.shape[1], rgb_image.shape[0],
QImage.Format.Format_RGB888)

            pixmap = QtGui.QPixmap.fromImage(qimg)

            self.camera_thread.setPixmap(pixmap)

```



```
if __name__ == "__main__":  
    app = QtWidgets.QApplication(sys.argv)  
    EmotionRecognition = QMainWindow()  
    ui = Ui_EmotionRecognition()  
    ui.setupUi(EmotionRecognition)  
    EmotionRecognition.show()  
    sys.exit(app.exec())
```

## **TrainEmotionDetected**

```
from keras.models import Sequential  
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten  
from keras.preprocessing.image import ImageDataGenerator  
import tensorflow as tf  
import json  
import matplotlib.pyplot as plt  
  
train_data_gen = ImageDataGenerator(rescale=1./255)  
validation_data_gen = ImageDataGenerator(rescale=1./255)  
  
# Preprocess all test images  
train_generator = train_data_gen.flow_from_directory(  
    'C:/Users/Artem/Desktop/data/train',  
    target_size=(48, 48),  
    batch_size=64,  
    color_mode="grayscale",  
    class_mode='categorical')  
  
# Preprocess all train images  
validation_generator = validation_data_gen.flow_from_directory(  
    'C:/Users/Artem/Desktop/data/train',  
    target_size=(48, 48),  
    batch_size=64,  
    color_mode="grayscale",  
    class_mode='categorical')
```

```

'C:/Users/Artem/Desktop/data/test',
target_size=(48, 48),
batch_size=64,
color_mode="grayscale",
class_mode='categorical')

emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48, 48,
1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))
emotion_model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.01,
    decay_steps=10000,
    decay_rate=0.9)
optimizer = tf.keras.optimizers.Adam(learning_rate=lr_schedule)

```

```
# Train the neural network/model
emotion_model_info = emotion_model.fit(
    train_generator,
    steps_per_epoch=28709 // 64,
    epochs=20,
    validation_data=validation_generator,
    validation_steps=7178 // 64)

model_json = emotion_model.to_json()
with open("emotion_model1905.json", "w") as json_file:
    json_file.write(model_json)

emotion_model.save_weights('emotion_model1905.h5')

plt.plot(emotion_model_info.history['accuracy'], label='Точність')
plt.plot(emotion_model_info.history['val_accuracy'], label='Точність на валідації')
plt.title('Історія точності моделі')
plt.ylabel('Точність')
plt.xlabel('Кількість циклів')
plt.legend()
plt.show()

with open('emotion_model_history.json', 'w') as f:
    json.dump(emotion_model_info.history, f)

with open('emotion_model_history.json', 'r') as f:
    history = json.load(f)
```