

Пояснювальна записка до курсової роботи

на тему: Програмне забезпечення виявлення мімічних ознак брехні та виразів
небезпечної поведінки людини

КП.ІП-1328.045490.01.81

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП.....	5
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	6
1.1 Загальні положення.....	6
1.2 Змістовний опис і аналіз предметної області	9
1.3 Аналіз існуючих технологій та успішних ІТ-проектів	9
1.3.1 Аналіз відомих алгоритмічних та технічних рішень.....	12
1.3.2 Аналіз допоміжних програмних засобів та засобів розробки	14
1.3.3 Аналіз відомих програмних продуктів	15
1.4 Аналіз вимог до програмного забезпечення.....	16
1.4.1 Розроблення функціональних вимог	21
1.4.2 Розроблення нефункціональних вимог	24
1.5 Постановка задачі.....	26
Висновки до розділу.....	27
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	28
2.1 Моделювання та аналіз програмного забезпечення	28
2.2 Архітектура програмного забезпечення	30
2.3 Конструювання програмного забезпечення	32
2.4 Аналіз безпеки даних.....	36
Висновки до розділу.....	36
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	37
3.1 Аналіз якості ПЗ.....	37
3.2 Опис процесів тестування	38
3.3 Опис контрольного прикладу	52
Висновки до розділу.....	58
4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.	62
4.1 Розгортання програмного забезпечення	62
4.2 Підтримка програмного забезпечення	62

Висновки до розділу	63
ВИСНОВКИ	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IDE	– Integrated Development Environment – інтегроване середовище розробки.
AI	– Artificial Intelligence – Штучний Інтелект
LSTM	– Long Short-Term Memory – довга короткострокова пам'ять
IT	– Інформаційні технології
CNN	– Convolutional Neural Network – згорткова нейронна мережа
OS	– Операційна система.
GBM	– Gradient Boosting Machine – метод градієнтного бустингу

ВСТУП

Термін "штучний інтелект" охоплює як "технологію, що імітує людський інтелект", так і ширшу сферу досліджень, присвячених його розвитку. Від роботів до мережевого розподіленого програмного забезпечення, штучний інтелект став повсюдним терміном, який використовується в різних галузях і для вирішення різноманітних проблем. Він імітує складну людську поведінку або спеціалізовані інтелектуальні функції. Примітно, що ШІ може навчатися автономно, без чітких інструкцій.

Останнім часом з'явилося багато досліджень, присвячених можливості використання штучного інтелекту в сфері психології. Для психологів, терапевтів і консультантів стає все більш важливим розуміння його нинішніх можливостей і майбутнього потенціалу, особливо в умовах трансформації системи охорони психічного здоров'я.

Проте аналіз емоцій виходить далеко за рамки поняття медицини, саме це твердження підкріплює дана курсова робота. Технології розпізнавання щирих і підроблених емоцій наочно описані у книзі американського професора і психолога Пола Екмана та були використані для втілення представленої роботи. Мімічні ознаки брехні виявляються на обличчі, коли людина намагається надати неправдиву відповідь на поставлене запитання або приховати інформацію. Ознаки виразу фальшивого суму можуть свідчити про навмисне прийняття позиції жертви, те саме відображає підроблена емоція страху, або ненавмисна щира усмішка при непідходящій для цього ситуації.

Дана курсова робота є прикладом реалізації та дослідженням використання принципів психології в симбіозі з нейронними мережами. Сфера використання може варіюватися від навчання та тренування людей з алекситимією, підкріплення результатів поліграфу для уникнення прийому на відповідальні посади сумнівних особистостей до полегшення задач фахівців сфери психології та запобігання випадків помилкового звинувачення у правовій системі.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Загальні положення

Основна інформація зі сфери психології яка необхідна для коректного втілення даної роботи зображена у науковій книзі «Теорія брехні»[1] Пола Екмана та базується на концепції, що емоції можна впізнавати за експресією обличчя. Там висвітлено поняття "мікроекспресій" – коротких виразів обличчя, які можуть вказувати на справжні або фальшиві емоції, що контрастують із зовнішнім вираженням. Це корисно у розкритті непрямих сигналів, які можуть вказувати також на приховані емоції або намагання приховати правду.

Згадана робота американського психолога також знаходить своє застосування в практиці, зокрема у сфері правосуддя. Експерти використовують її для аналізу виразів обличчя, що може допомагати в розслідуванні злочинів, виявленні шахрайств, ідентифікації можливих загроз тощо. Проте така навичка потребує довгого часу навчання, крім цього включає «людський фактор» з чим би допомогло впоратись застосування штучного інтелекту. Загалом, глибше і точніше розуміння взаємозв'язку між емоціями та невербальним вираженням обличчя може відкрити нові можливості для досліджень у галузі психології та соціальних наук. Пропозиція моєї роботи це також застосування для допомоги людям з алекситимією, підкріплення роботи поліграфів для точнішого результату, а також для навчання методам представленим у книзі наочним способом.

Екман визначив шість основних емоцій, які виражаються у міміці обличчя: щастя, смуток, гнів, страх, здивування та огида. Ця теорія допомагає в розумінні того, як люди виявляють свої емоції через фізіологічні вирази обличчя та саме вони будуть визначатися у даній роботі. Три з основних емоцій найчастіше підпадають під фальсифікацію, а саме сум щастя та радість, що відображається посмішкою (рисунок 1.1, 1.2).

Фальшива емоція	Поведінкова ознака
Жах	Відсутність надійних ознак на чолі
Сум	Відсутність надійних ознак на чолі
Радість	Не беруть участь м'язи очей

Рисунок 1.1 Ознаки фальшивих емоцій[1]

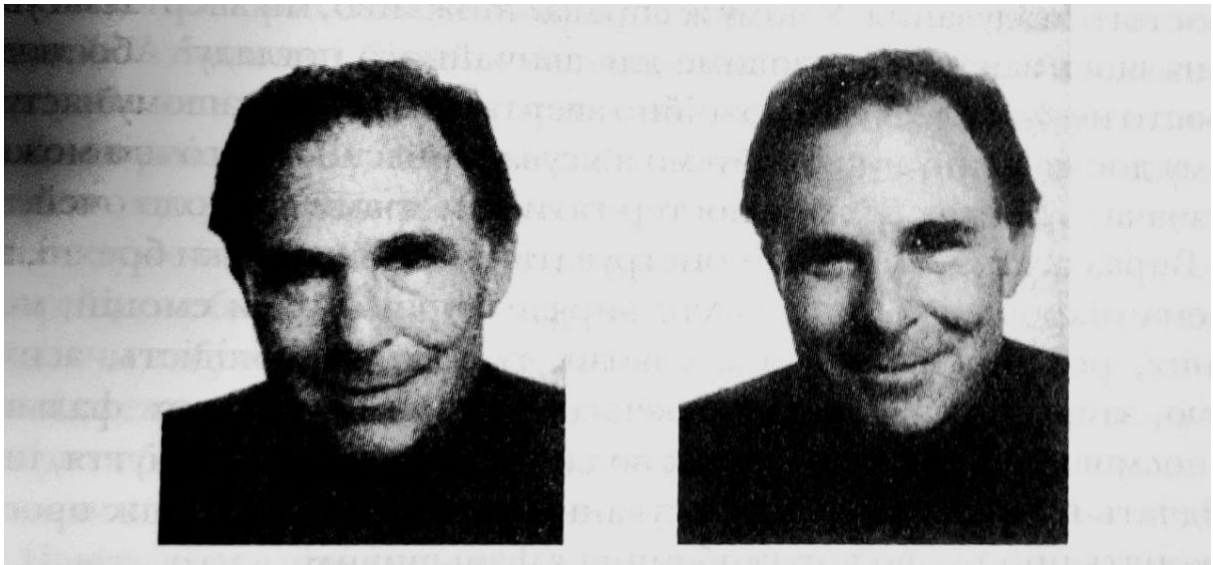


Рисунок 1.2 Справа щира усмішка, зліва фальшива усмішка[1]

Також важливість сприйняття поведінки та людських емоцій через вираз обличчя висвітлює стаття кафедри комп'ютерних наук та інженерії Національний технологічний інституту Дургапур[2]. Вона відзначає, що щира посмішка відображає впевненість та сприймається позитивно у взаємодії та комунікації, а фальшива може вказувати на невпевненість особи. Також в тексті висвітлюється те, що розпізнавання справжнього та фальшивого виразу обличчя є складним завданням для людського мозку, що підтверджує користь даної роботи у напрямку застосування AI для розвитку сфери психології та соціальної обізнаності людини, тощо.

Автор висвітлив питання того, як комп'ютерні системи можуть розрізняти справжні емоції від фальшивих. Зазначається, що на сьогоднішній день немає адекватної відповіді на це питання, але в тексті вказано, що було представлено різні обчислювальні методи для розв'язання цієї проблеми. Зокрема, зазначається робота французького лікаря Гійома Дюшенна, який у 19 столітті досліджував

м'язи, що беруть участь у формуванні міміки, з метою відрізнєння справжньої посмішки від фальшивої. Також згадується використання очей як елемєнту для виявлення справжньої та фальшивої посмішки[2].

Отже дана курсова робота передбачає створєння застосунку який буде використовувати CNN моделі для розпізнавання емоції та якщо вона підпадає під найчастіше використовувані(рис 1.1.1) для приховування інформації та означєння брєхні - визначєння її правдивості на файлі який користувач завантажив для описаної обробки. Передбачається що користувач буде завантажувати короткі уривки інтерв'ю або зображення обличчя, та визначати результат (таблиця 1.1).

Таблиця 1.1 Можливе інтерпрєтування результату

Емоція	Можливе визначєння залежне від контексту
Справжній сум	Людину засмутило питання, згадка ситуації, приховування душевного болю, приховування не знання ситуації, приховування провини
Несправжній сум	Навмисна «позиція жертви», вдавання суму задля приховання провини
Справжня посмішка	Щира радість до створєної ситуації, гордість вчинком про який було згадано(може бути виявом брєхні у ситуацію коли людина заперєчує провину, проте виникає дана експресія)

Продовження таблиці 1.1

Несправжня посмішка	Удавання невразливості, невпевненість, зніяковілість, використовується для того щоб впевнити співрозмовника у своїй впевненості, при цьому сама особа в це не вірить, тобто ознака брехні)
Справжній страх	Реакція на жахливу ситуацію, страх бути викритим, що розкриється брехня, також вияв признання провини
Несправжній страх	Навмисна «позиція жертви», вдавання страху задля певної вигоди(наприклад звинувачення іншої особи)
Огида	Признак попередження небезпечної поведінки, визначення огиди до співрозмовника або людини в полі зору, презирство інших, що також може бути ознакою виявлення агресії та навіть попереджати напад
Гнів	Агресія, признак попередження небезпечної поведінки та нападу
Здивування	Визначення справжнього не знання про ситуацію, також допомагає визначити правдивість провини

1.2 Змістовний опис і аналіз предметної області

Додатковий розвиток та імплементація систем розпізнавання зображень в сфері AI надає широкий спектр можливостей в різних галузях, зокрема таких як

медицина та психологія. Проте, навіть при високій продуктивності та точності сучасних систем розпізнавання, існують питання, які потребують уваги.

Одним із недоліків поточного стану речей у сфері розпізнавання зображень є нестабільність результатів у складних умовах, таких як змінність освітлення, прикладом може бути алгоритм Віоли Джонса[3][4], який до того ж чутливий до градусу нахилу обличчя на зображенні. Також важливо враховувати аспекти ефективності та швидкодії при роботі з великими обсягами даних.

Для покращення ситуації можна розглядати шляхи оптимізації алгоритмів розпізнавання та вдосконалення методів навчання нейронних мереж. Крім того важливо розглядати варіанти використання передових технологій у сфері обчислень, щоб забезпечити оптимальну швидкість систем. Знову спираючись на алгоритм Віоли Джонса[3][4], замість якого у сучасних програмах краще використовувати MT-CNN[5].

У контексті курсової роботи вибраний шлях дослідження та оптимізації алгоритмів розпізнавання зображень з використанням згорткових нейронних мереж CNN. Основний акцент робиться на аналізі та вирішенні конкретних викликів, що виникають при впровадженні системи розпізнавання.

Під час дослідження будуть враховані підходи до покращення ефективності за допомогою використання новіших активаційних функцій, таких як mish[7] та точності розпізнавання, зокрема, застосування технік передньої обробки зображень, оптимізація та розробка архітектури мережі під конкретне завдання для уникнення перевантаження. Основна мета – досягти оптимального балансу між точністю розпізнавання та швидкістю обробки, що є критичним для практичного застосування нейронних мереж.

1.3 Аналіз існуючих технологій та успішних ІТ-проектів

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації програмного забезпечення виявлення мімчних ознак брехні та небезпечної поведінки людини.

Далі будуть розглянуті допоміжні програмні засоби, засоби розробки та готові програмні рішення.

На сьогоднішній день розрізнення між спонтанними та фальсифікованими емоціями шляхом аналізу обличчя є досить важкою задачею в галузі розпізнавання емоцій за допомогою комп'ютерних систем. У цьому розділі проведемо аналіз застосунку представленого у статті «Discrimination between genuine versus fake emotion using long-short term memory with parametric bias and facial landmarks»[6] де використовується LSTM з параметричним зміщенням та використанням лицьових точкових ознак.

Загалом для розрізнення між справжніми та вигаданими емоціями існують різні підходи. Більшість існуючих систем зосереджуються на класифікації емоцій за виразом обличчя, але розрізнення їх справжності з точки зору науки психології не завжди точне (рисунок 1.3).



Рисунок 1.3 Приклади зображень справжньої та фальшивої міміки, що використовуються в програмному забезпеченні[6]

У наведеному в статті програмному забезпеченні є поєднання моделювання дзеркальних нейронів та глибоких рекурентних мереж з РВ. Основною перевагою цього підходу є можливість виділення ознак у просторово-часовому домені.

Безпосереднє використання бібліотеки DLib[8] для застосування алгоритму обробки зображень є цікавим рішенням.

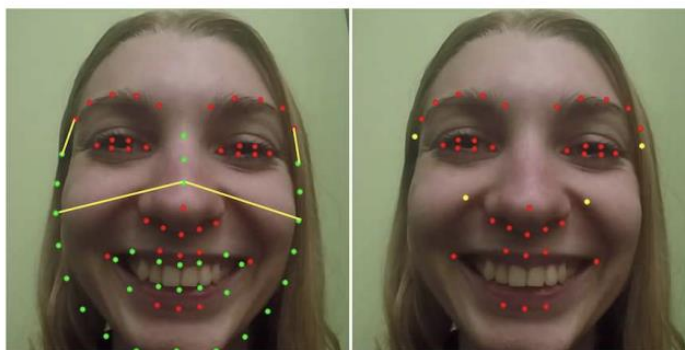


Рисунок 1.4 Ознаки обличчя дослідження[6]

Представлене програмне забезпечення розробляє та навчає модель для відтворення просторово-часових залежностей обличчя та емоцій, також використовує архітектуру 2D Grid[9] для збереження інформації на тривалий термін. Крім того використовується GBM для класифікації РВ векторів на справжні та фальсифіковані емоції та навчається з використанням великої кількості різноманітних даних, включаючи аугментовані зразки. Також тут використовуються техніки регуляризації для уникнення перенавчання та підвищення стабільності моделі. Розроблена система здатна виявляти реальні та вигадані емоції з точністю[10], що дорівнює 66.7% і є доволі непоганим результатом, але не виправдовує складність ПЗ, час та ресурси на виконання. Проте при виконанні курсової роботи варто взяти до уваги представлене в цьому програмному забезпеченні використання бібліотеки Dlib.

1.3.1 Аналіз відомих алгоритмічних та технічних рішень

Розглянемо систему для розпізнавання емоцій на обличчях зі статті «Discriminating Real from Fake Smile Using Convolution Neural Network»[2] вже загаданої раніше, де використовується три ключові алгоритми: алгоритм тренування (Algorithm 1), алгоритм тестування (Algorithm 2) та алгоритм попередньої обробки зображень (Viola-Jones algorithm) (рисунок 1.5).

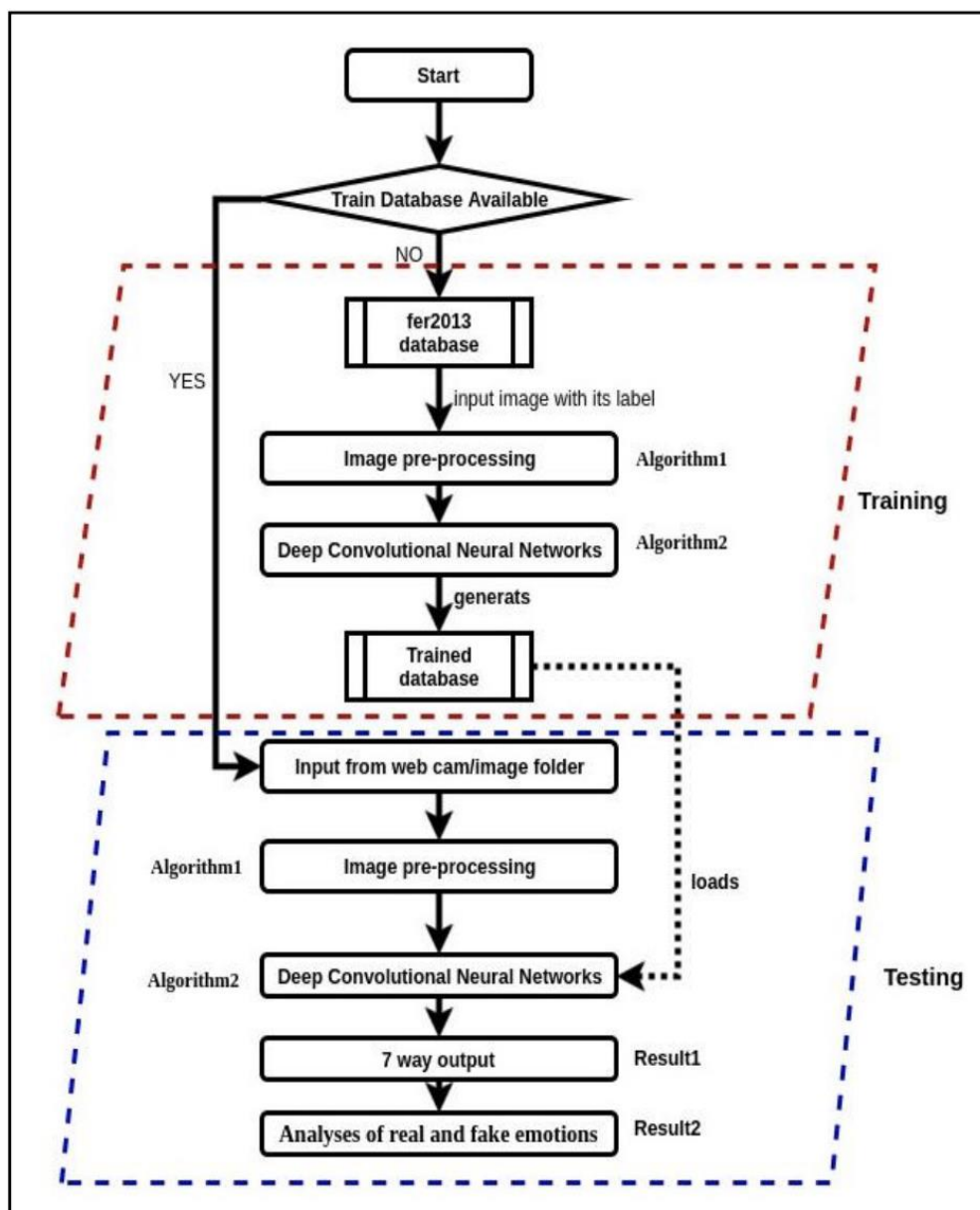


Рисунок 1.5 Схема роботи проекту[2]

Інтегрована архітектура розділена на дві фундаментальні частини: тренування та тестування. Перед переходом до етапу тестування систему необхідно піддати тренуванню для ефективного виявлення емоцій. Важливий аспект - перевірка наявності тренуваних даних. Якщо дані не готові, система автоматично переходить до тренування.

Алгоритм включає кроки виявлення та обробки обличчя, зокрема використання алгоритму Віолі-Джонса для точного визначення обличчя та його атрибутів. Пройшовши через алгоритм, обличчя зберігається для подальшого використання у тестуванні. У випадку вже існуючих тренуваних даних система

переходить до етапу тестування, використовуючи підготовлені дані для емоційного аналізу зображень. Для тренування та тестування використовується база даних FER-2013[11], яка, як відзначено, є ідеальною для цієї задачі завдяки своєму розмаїттю виразів обличчя.

Після підготовки даних докладно описано глибокі згорткові нейронні мережі (CNN)[12], які використовуються для розпізнавання емоцій. Наведено покроковий опис алгоритму CNN (рисунок 1.6), розкриваючи процес встановлення ваг та фільтрів, прямого та оберненого поширення, обчислення помилки та коригування параметрів. Особлива увага приділяється структурі шарів, таких як згорткові та повністю з'єднані, що сприяють високій точності класифікації на зображеннях.

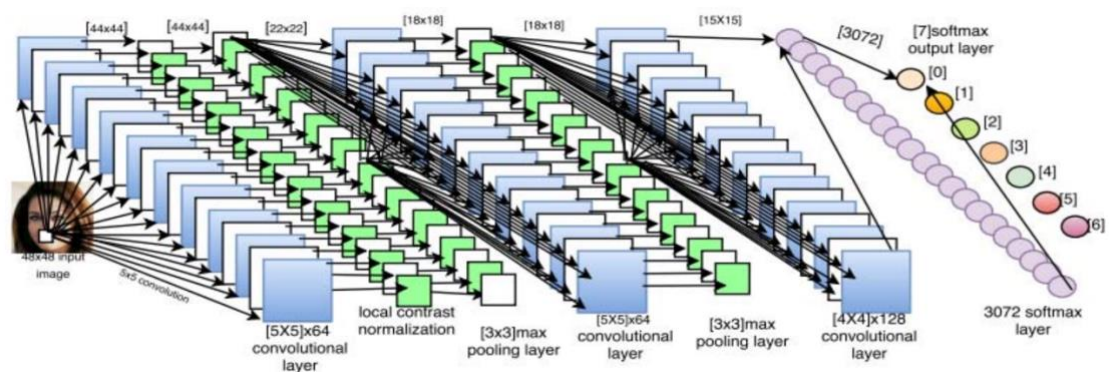


Рисунок 1.6 Архітектура CNN проекту[2]

До курсової роботи буде доцільно використати такі технічні рішення як: використання окремого алгоритму для попередньої обробки вхідних зображень та їх повернення та використання відкритого датасету FER-2013 для тренування та тестування системи, оскільки він підходить для цієї задачі завдяки своєму розмаїттю виразів обличчя.

1.3.2 Аналіз допоміжних програмних засобів та засобів розробки

При розгляді вибору мови програмування та IDE для системи розпізнавання емоцій на обличчях, Python є оптимальним вибором з огляду на його легку інтеграцію з бібліотеками ML[13]. Для реалізації CNN краще буде використовувати TensorFlow та Keras, які надають зручний інтерфейс для

побудови та тренування мереж. У сфері виявлення обличч – MTCNN оптимальний вибір для розпізнавання обличч на зображеннях також зважаючи на чутливість алгоритму Віоли-Джонса до градусу їх повороту та чутливості до якості освітлення краще використовувати MTCNN. Також PİL доповнює функціонал системи, забезпечуючи операції з роботою з зображеннями, такі як зміна розміру та конвертація форматів Важливим також є використання бібліотеки OpenCV[14], оскільки вона хороший інструмент обрізки, розпізнавання контурів та інших операцій, які є важливими на етапі попередньої обробки зображень. Для забезпечення доступу до графічних процесорів під час тренування моделей краще буде користуватись Google Colab[15], безкоштовним середовищем для виконання коду Python у хмарі.

Усі ці інструменти взаємодіють для створення коректної та ефективної роботи програмного забезпечення розпізнавання емоцій на обличчях.

1.3.3 Аналіз відомих програмних продуктів

Для порівняння курсової роботи з аналогом можна скористатись таблицею 1.2

Таблиця 1.2 – Порівняння з аналогом

Функціонал	Курсова робота (PekmanAnnalyzer)	Програмне забезпечення стаття [6]	Microsoft Azure Face API:	Пояснення
Класифікація основних 6 емоцій	+	+	+	Всі програми розпізнають 6 основних людський емоцій на вхідних даних

Продовження таблиці 1.2

Розпізнавання фальшивих і справжніх емоцій	+	+	-	Не всі програми використовують методи розпізнавання фальсифікованих чи справжніх емоцій за виразом обличчя
Наявність інтерфейсу	+	-	+	Не всі програми мають інтерфейс
Можливість зберігати результат	+	-	+	Не всі програми дозволяють зберігати результат класифікації

1.4 Аналіз вимог до програмного забезпечення

Головною функцією програмного забезпечення є PekmanAnalyzer, більше функцій можна побачити на рисунку 1.7.

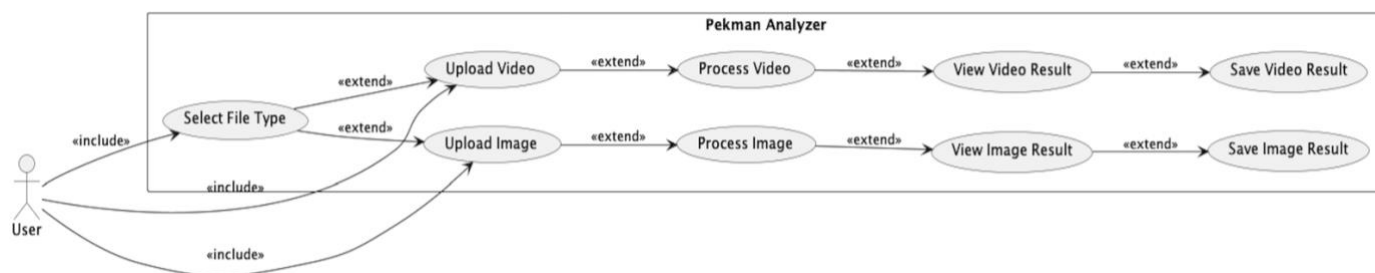


Рисунок 1.7 – Діаграма варіантів використання

В таблицях 1.3 - 1.11 наведені варіанти використання програмного забезпечення.

Таблиця 1.3 - Варіант використання UC-1

Use case name	Вибір типу файла
Use case ID	UC-01
Goals	Обрати тип файлу для завантаження
Actors	User(користувач)
Trigger	Користувач відкриває додаток
Pre-conditions	-
Flow of Events	Користувач відкриває додаток та вибирає тип файлу для завантаження
Extension	Без вибору файлу не доступна кнопка «Next» (Далі)
Post-Condition	Після вибору кнопка «Next» (Далі) стає доступною

Таблиця 1.4 - Варіант використання UC-2

Use case name	Завантаження зображення
Use case ID	UC-02
Goals	Завантаження зображення для обробки
Actors	User(користувач)
Trigger	Користувач натискає на кнопку «Upload image» (Завантажити зображення) та обирає зображення для завантаження
Pre-conditions	Обраний тип файлу «image»
Flow of Events	Користувач завантажує зображення, файл передається на перевірку
Extension	У випадку вибору файлу яке не є зображенням виводиться повідомлення про помилку спроби завантаження хибного файлу, або файлу що перевищив розмір допущеного
Post-Condition	Зображення передається на аналіз при натиску кнопки «Next»(Далі)

Таблиця 1.5 - Варіант використання UC-3

Use case name	Завантаження відео
Use case ID	UC-03
Goals	Завантаження відео для обробки
Actors	User(користувач)
Trigger	Користувач натискає на кнопку «Upload video»(Завантажити відео) та обирає відео для завантаження
Pre-conditions	Обраний тип файлу «video»
Flow of Events	Користувач завантажує відео, файл передається на перевірку
Extension	У випадку вибору файлу яке не є відео виводиться повідомлення про помилку спроби завантаження хибного файлу, або файлу що перевищив розмір допущеного
Post-Condition	Відео передається на аналіз при натиску кнопки «Next»(Далі)

Таблиця 1.6 - Варіант використання UC-4

Use case name	Ааналіз зображення
Use case ID	UC-04
Goals	Розпізнавання емоцій на завантаженому зображенні
Actors	User(користувач)
Trigger	Завантаження зображення
Pre-conditions	Зображення успішно завантажено
Flow of Events	Відбувається аналіз емоцій на зображенні за допомогою відповідних моделей
Extension	У випадку якщо на зображенні не виявиться обличчя, не буде отримано результат обробки
Post-Condition	Результат аналізу повертається до модуля інтерфейсу

Таблиця 1.7 - Варіант використання UC-5

Use case name	Аналіз відео
Use case ID	UC-05
Goals	Розпізнавання емоцій на завантаженому відео

Продовження таблиці 1.7

Actors	User(користувач)
Trigger	Завантаження відео
Pre-conditions	Відео успішно завантажено
Flow of Events	Відбувається аналіз емоцій на фреймах відео за допомогою відповідних моделей
Extension	У випадку якщо на відео не виявиться обличчя, не буде отримано результат обробки
Post-Condition	Результат аналізу повертається до модуля інтерфейсу

Таблиця 1.8 - Варіант використання UC-6

Use case name	Перегляд результатів аналізу зображення
Use case ID	UC-06
Goals	Перегляд результатів аналізу зображення
Actors	User(користувач)
Trigger	Результат аналізу зображення був переданий в модуль інтерфейсу
Pre-conditions	Завершення аналізу зображення
Flow of Events	Користувач натиснув кнопку «View result» (Переглянути результат), система відображає зображення з написом результату аналізу
Extension	
Post-Condition	Оброблене зображення відображене на екрані

Таблиця 1.9 - Варіант використання UC-7

Use case name	Перегляд результатів аналізу відео
Use case ID	UC-06
Goals	Перегляд результатів аналізу відео
Actors	User(користувач)
Trigger	Результат аналізу відео був переданий в модуль інтерфейсу
Pre-conditions	Завершення аналізу відео

Продовження таблиці 1.9

Flow of Events	Користувач натиснув кнопку «View result» (Переглянути результат), система відображає відео з написом результату аналізу на кожному фреймі
Extension	
Post-Condition	Оброблене відео відображене на екрані

Таблиця 1.10 - Варіант використання UC-8

Use case name	Збереження результатів зображення
Use case ID	UC-08
Goals	Зберегти результат обробки зображення
Actors	User(користувач)
Trigger	Результат аналізу зображення був переданий в модуль інтерфейсу
Pre-conditions	Завершення аналізу зображення
Flow of Events	Користувач натиснув кнопку «Save result» (Переглянути результат), система пропонує обрати шлях куди завантажити оброблене зображення
Extension	
Post-Condition	Оброблене зображення збережено

Таблиця 1.11 - Варіант використання UC-9

Use case name	Збереження результатів відео
Use case ID	UC-08
Goals	Зберегти результат обробки відео
Actors	User(користувач)
Trigger	Результат аналізу відео був переданий в модуль інтерфейсу
Pre-conditions	Завершення аналізу відео
Flow of Events	Користувач натиснув кнопку «Save result» (Переглянути результат), система пропонує обрати шлях куди завантажити оброблене відео

Продовження таблиці 1.11

Extension	
Post-Condition	Оброблене відео збережено

1.4.1 Розроблення функціональних вимог

Програмне забезпечення розділене на модулі. Кожен модуль має свій певний набір функцій. На рисунку 1.8 наведено загальну модель вимог, а в таблицях 1.7 – 1.20 наведений опис функціональних вимог до програмного забезпечення. Матрицю трасування вимог можна побачити на рисунку 1.9.

Назва вимоги	Опис вимоги
FR-1: Вибір типу файлу	Користувач відкриває додаток та обирає тип файлу: зображення чи відео. Залежно від вибору, відкривається відповідне вікно.
FR-2: Завантаження зображення	У вікні завантаження файлу виводиться кнопка "Завантажити зображення".
FR-3: Перевірка типу файлу для зображення	Якщо користувач намагався завантажити файл, що не є зображенням, то виводиться відповідне повідомлення про помилку.
FR-4: Перевірка розміру зображення	Якщо користувач намагався завантажити зображення, розміром більше 10МБ, то виводиться відповідне повідомлення про помилку перевищення розміру файлу.
FR-5: Завантаження відео	У вікні завантаження файлу виводиться кнопка "Завантажити відео".
FR-6: Перевірка типу файлу для відео	Якщо користувач намагався завантажити файл, що не є відео, то виводиться відповідне повідомлення про помилку.
FR-7: Перевірка розміру відео	Якщо користувач намагався завантажити відео, розміром більше 100МБ, то виводиться відповідне повідомлення про помилку перевищення розміру файлу.
FR-8: Перехід до наступного вікна	Кнопка «далі» стає активною тільки після завантаження файлу.
FR-9: Аналіз зображення	Відбувається аналіз зображення, відкривається вікно відслідковування прогресу
FR-10: Аналіз відео	Відбувається аналіз відео, відкривається вікно відслідковування прогресу
FR-11: Вибір перегляду результату для зображень	На вікні результату відображаються дві кнопки, користувач натискає «Переглянути результат», відкривається зображення з написом класифікованої емоції.
FR-12: Вибір збереження результату для зображень	На вікні результату відображаються дві кнопки, користувач натискає «Зберегти результат», зображення з написом класифікованої емоції зберігається по вибрану користувачем шляху.
FR-13: Вибір перегляду результату для відео	На вікні результату відображаються дві кнопки, користувач натискає «Переглянути результат».
FR-14: Вибір збереження результату для відео	На вікні результату відображаються дві кнопки, користувач натискає «Переглянути результат».

Рисунок 1.8 – Модель вимог у загальному вигляді

Таблиця 1.7 – Функціональна вимога FR-1

Назва	Вибір типу файлу
Опис	Користувач відкриває додаток та обирає тип файлу: зображення чи відео. Залежно від вибору, відкривається відповідне вікно.

Таблиця 1.8 – Функціональна вимога FR-2

Назва	Завантаження зображення
Опис	У вікні завантаження файлу виводиться кнопка "Завантажити зображення"

Таблиця 1.9 – Функціональна вимога FR-3

Назва	Перевірка типу файлу для зображення
Опис	Якщо користувач намагався завантажити файл, що не є зображенням, то виводиться відповідне повідомлення про помилку

Таблиця 1.10 – Функціональна вимога FR-4

Назва	Перевірка розміру зображення
Опис	Якщо користувач намагався завантажити зображення, розміром більше 10МБ, то виводиться відповідне повідомлення про помилку перевищення розміру файлу

Таблиця 1.11 – Функціональна вимога FR-5

Назва	Завантаження відео
Опис	У вікні завантаження файлу виводиться кнопка "Завантажити відео"

Таблиця 1.12 – Функціональна вимога FR-6

Назва	Перевірка типу файлу для відео
Опис	Якщо користувач намагався завантажити файл, що не є відео, то виводиться відповідне повідомлення про помилку

Таблиця 1.13 – Функціональна вимога FR-7

Назва	Перевірка розміру відео
Опис	Якщо користувач намагався завантажити відео, розміром більше 100МБ, то виводиться відповідне повідомлення про помилку перевищення розміру файлу

Таблиця 1.14 – Функціональна вимога FR-8

Назва	Перехід до наступного вікна
Опис	Кнопка «далі» стає активною тільки після завантаження файлу

Таблиця 1.15 – Функціональна вимога FR-9

Назва	Аналіз зображення
Опис	Відбувається аналіз зображення, відкривається вікно відслідковування прогресу

Таблиця 1.16 – Функціональна вимога FR-10

Назва	Аналіз відео
Опис	Відбувається аналіз відео, відкривається вікно відслідковування прогресу

Таблиця 1.17 – Функціональна вимога FR-11

Назва	Вибір перегляду результату для зображень
Опис	На вікні результату відображаються дві кнопки, користувач натискає «Переглянути результат», відкривається зображення з написом класифікованої емоції

Таблиця 1.18 – Функціональна вимога FR-12

Назва	Вибір збереження результату для зображень
-------	---

Продовження таблиці 1.18

Опис	На вікні результату відображаються дві кнопки, користувач натискає «Зберегти результат», зображення з написом класифікованої емоції зберігається по вибрану користувачем шляху
------	--

Таблиця 1.19 – Функціональна вимога FR-13

Назва	Вибір перегляду результату для відео
Опис	На вікні результату відображаються дві кнопки, користувач натискає «Переглянути результат»

Таблиця 1.20 – Функціональна вимога FR-14

Назва	Вибір збереження результату для відео
Опис	На вікні результату відображаються дві кнопки, користувач натискає «Зберегти результат», відео з відображенням рамки навколо виявленого обличчя і класифікації емоції на кожному кадрі зберігається по вибраному користувачем шляху

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8	FR-9	FR-10	FR-11	FR-12	FR-13	FR-14
UC-01	+													
UC-02		+	+	+				+						
UC-03					+	+	+	+						
UC-04									+					
UC-05										+				
UC-06											+			
UC-07													+	
UC-08												+		
UC-09														+

Рисунок 1.9 – Матриця трасування вимог

1.4.2 Розроблення нефункціональних вимог

Вид обслуговування:

Вимоги до виду обслуговування не висуваються.

Обслуговуючий персонал:

Вимоги до обслуговуючого персоналу не висуваються.

Вимоги до складу і параметрів технічних засобів:

Програмне забезпечення повинно функціонувати на IBM-сумісних персональних комп'ютерах.

Мінімальна технічна конфігурація:

- Тип процесору: Intel Core i7

- Об'єм ОЗП: 8 Гб

Рекомендована технічна конфігурація:

- Тип процесору: Intel Core i7

- Об'єм ОЗП: 16 Гб

Вимоги до інформаційної та програмної сумісності:

Програмне забезпечення повинно бути сумісним з операційною системою macOS версії 13.3 (MacOS Mojave).

Вимоги до вхідних даних:

- Зображення: формат JPG, IMEG, PNG розмір до 10 МБ

- Відео: формат MP4, AVI, MOV розмір до 100 МБ

Вимоги до вихідних даних:

- Зображення: формат JPG

- Відео: формат MP4

Вимоги до мови розробки:

Розробку виконати на мові програмування Python.

Вимоги до середовища розробки:

Розробку виконати на платформі PyCharm, тренування моделей у Google Colaboratory

Вимоги до представлення вихідних кодів

Вихідний код програми має бути представлений у вигляді архіву з .app файлом

Вимоги до маркування та пакування:

Вимоги до маркування та пакування не висуваються.

Вимоги до транспортування та зберігання:

Вимоги до транспортування та зберігання не висуваються.

Спеціальні вимоги:

Спеціальні вимоги не висуваються

1.5 Постановка задачі

Розробка даного програмного забезпечення має на меті отримання класифікації емоцій та експресій на відео та зображеннях. Основна ідея полягає в тому, щоб визначити «реальність» емоцій, які часто використовуються для замаскування чи спотворення інформації(сум, страх, посмішка), а також можуть служити ознаками брехні і небезпечної поведінки(гнів, огида) відповідно до теорії визнаного професора психології Пола Екмана[1].

Основне призначення цієї роботи може мати різноманітні напрями, починаючи від тренування людей з алексітимією і закінчуючи використанням отриманих результатів для підтримки рішень поліграфічних експертів, щоб уникнути найму осіб з сумнівними характеристиками на відповідальні посади. Крім того важливо також зазначити, що дана технологія може стати корисною для фахівців у галузі психології, надаючи їм додаткові інструменти для вивчення та розуміння людської поведінки. Також, вона може сприяти запобіганню неправомірних звинувачень у юридичній системі та полегшенню завдань

судових експертів. Такий комплексний підхід дозволяє максимально використовувати потенціал технології для різних сфер суспільства.

Висновки до розділу

У цьому розділі було проведено аналіз вимог до програмного забезпечення, що є ключовим етапом у процесі розробки будь-якого ІТ-проекту.

Також було визначено загальний контекст і завдання. Визначення опису та аналізу предметної області дав можливість зрозуміти специфіку середовища, в якому буде функціонувати програмне забезпечення даної курсової роботи. Аналіз існуючих технологій та успішних ІТ-проектів вказав на можливості використання сучасних рішень та підходів AI, що сприятимуть ефективній розробці та впровадженню системи.

Було розглянуто відомі алгоритмічні та технічні рішення вже існуючих програмних продуктів зі схожою задачею. Також порівняння програмних засобів дало змогу визначити те, що найбільше підходить для впровадження описаного проекту. Був розширений аналіз вимог до програмного забезпечення за допомогою розроблення функціональних та нефункціональних вимог, а в постановці задачі вказано чітко сформульоване завдання для подальшої розробки програмного забезпечення, визначено обсяг робіт та конкретизовано цілі.

В цілому, проведений аналіз став основою для подальшого проектування та реалізації програмного забезпечення, дозволяючи визначити оптимальний шлях впровадження та досягнення поставленої задачі у найбільш ефективний спосіб.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для опису бізнес процесу програмного забезпечення використовується BPMN модель (рисунок 2.1)(рисунок 2.2).

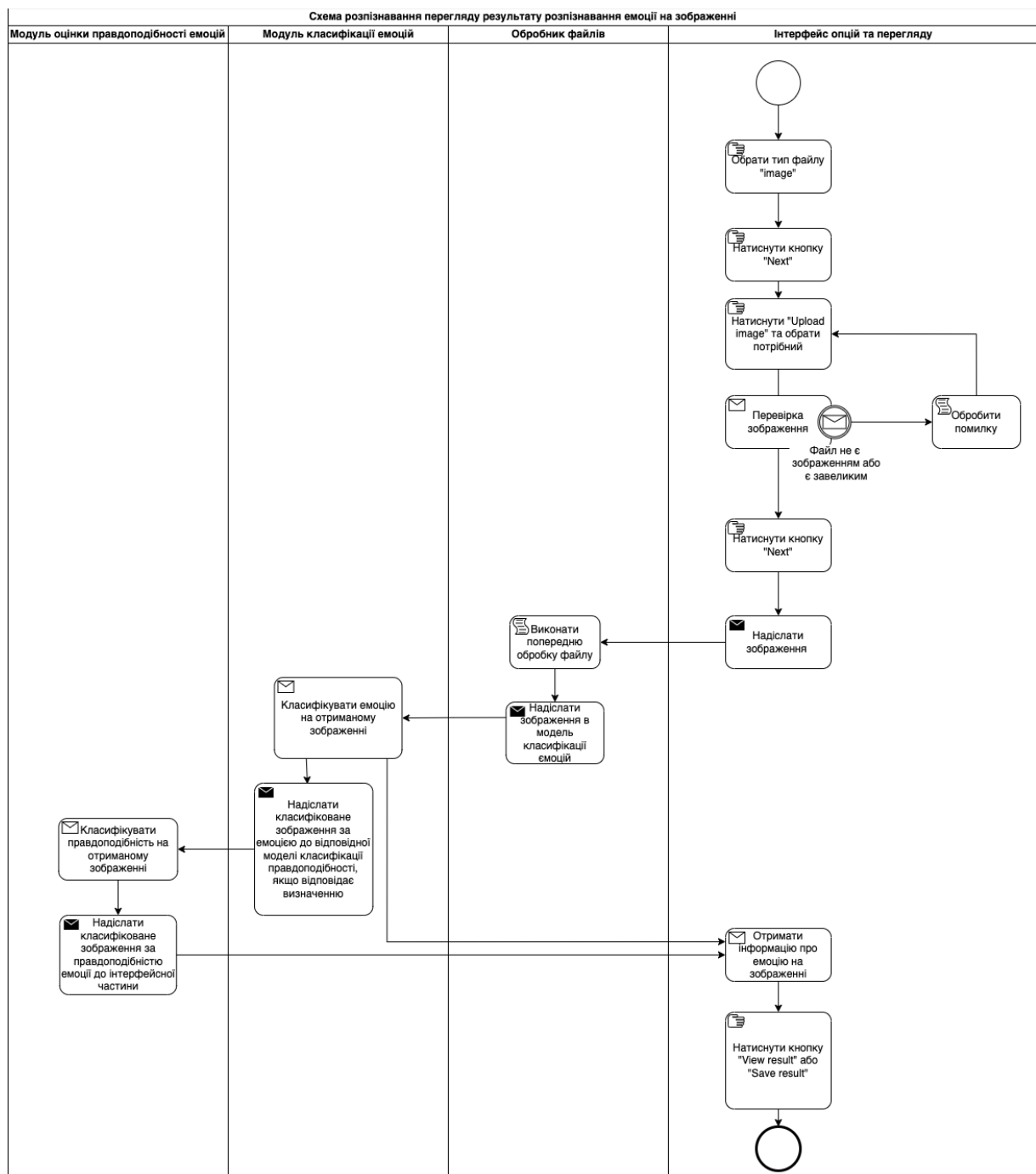


Рисунок 2.1 BPMN “Обработка изображений”

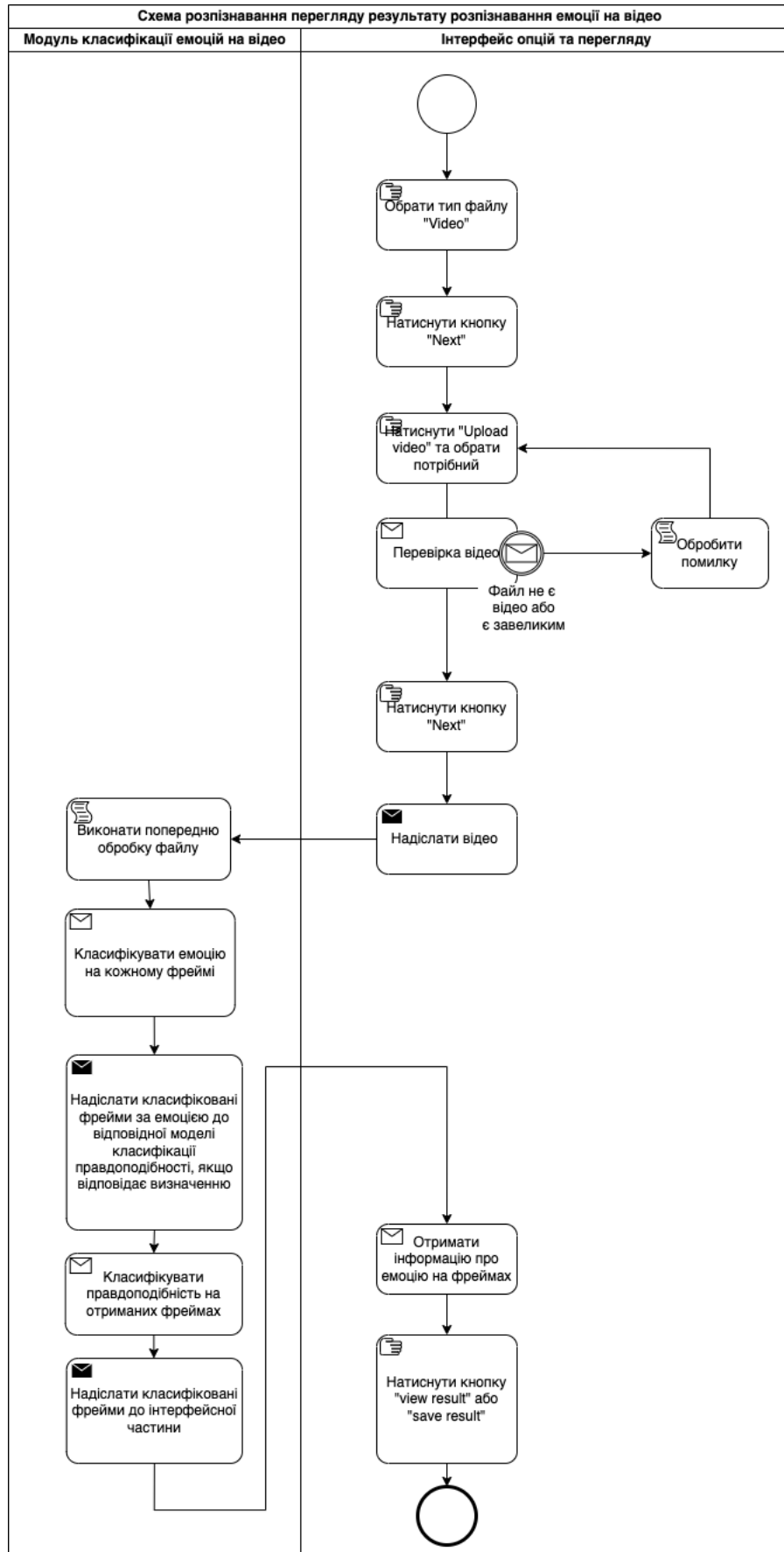


Рисунок 2.2 BPMN “Обробка відео”

2.2 Архітектура програмного забезпечення

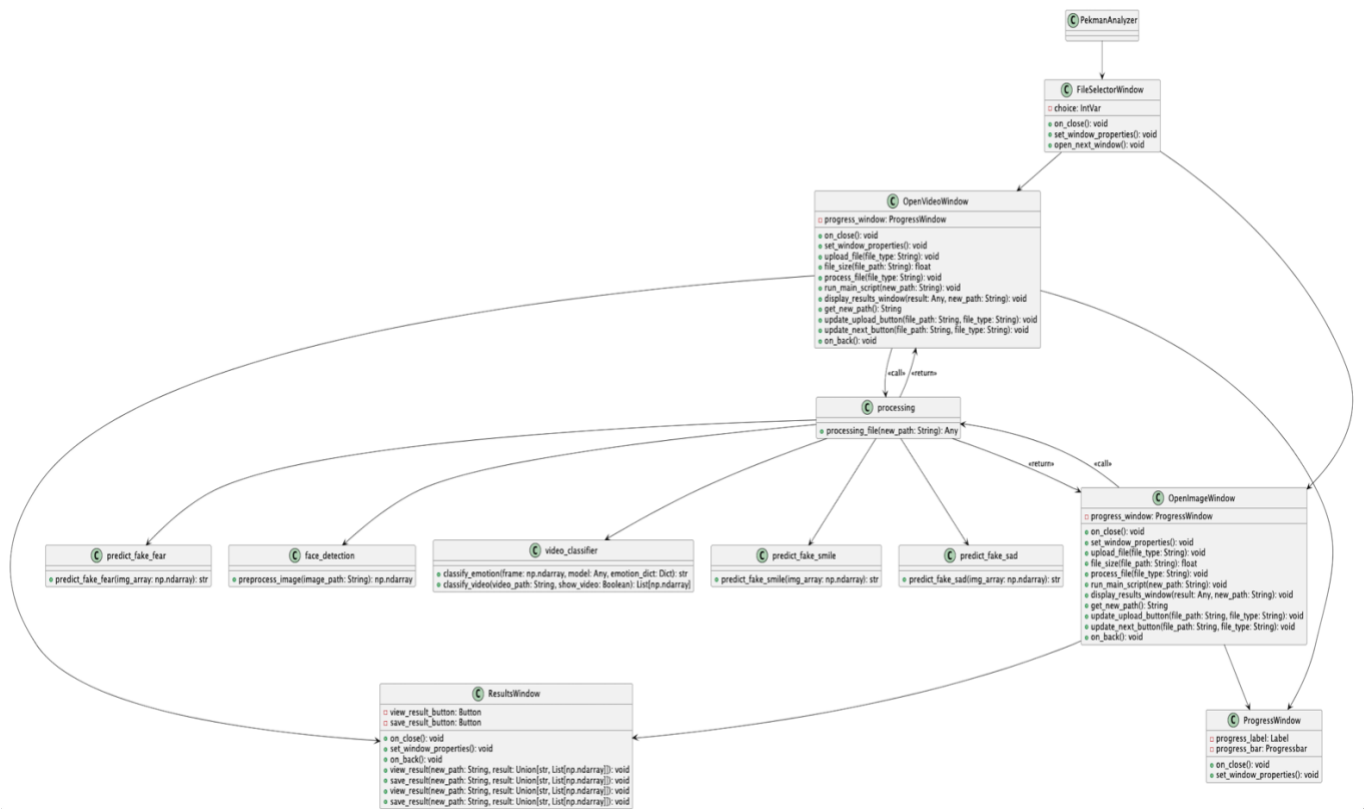


Рисунок 2.3 Детальна діаграма класів

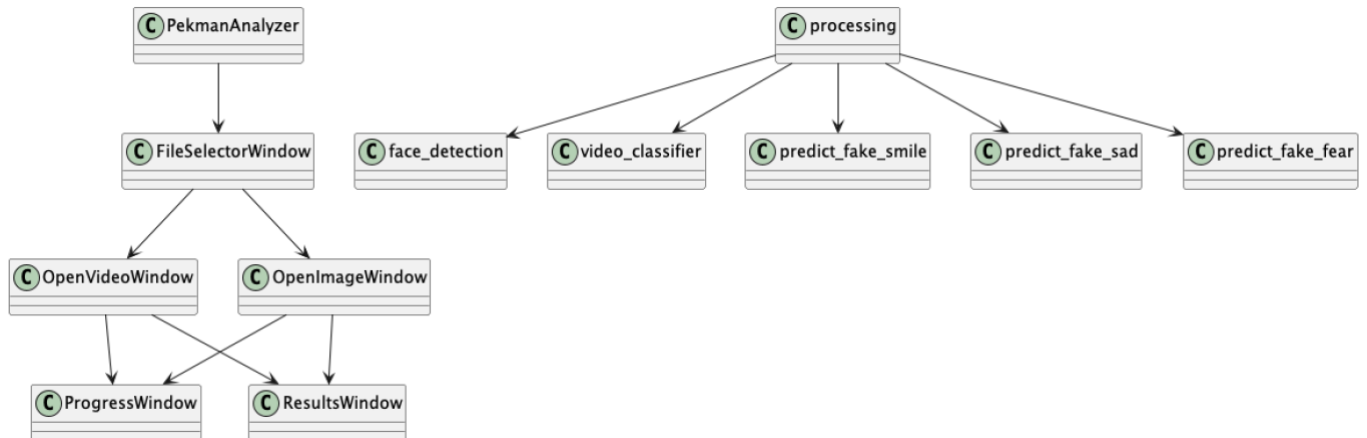


Рисунок 2.4 Загальна діаграма класів

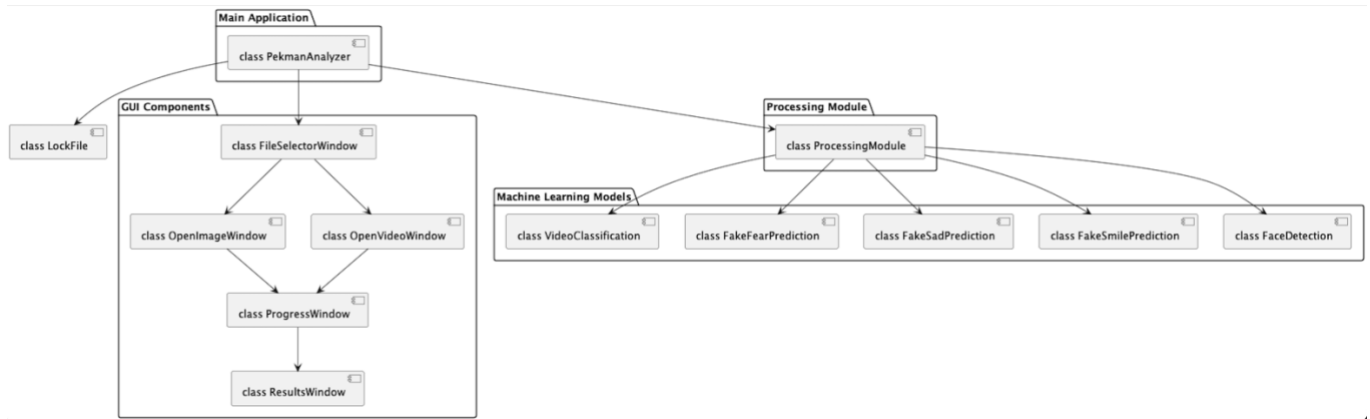


Рисунок 2.5 Діаграма компонентів

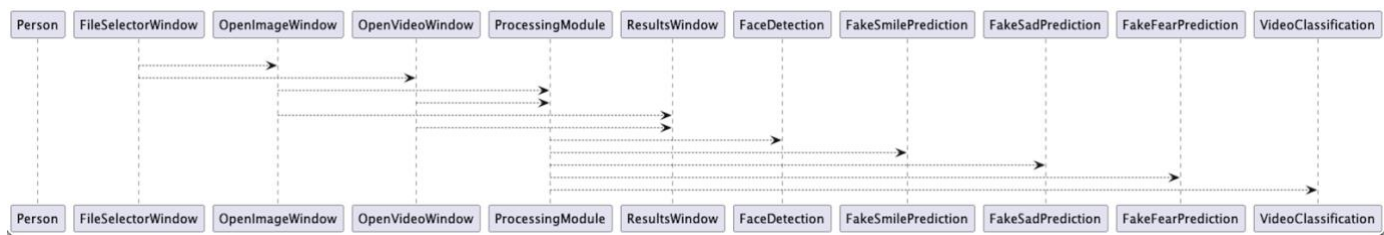


Рисунок 2.6 Діаграма взаємодії

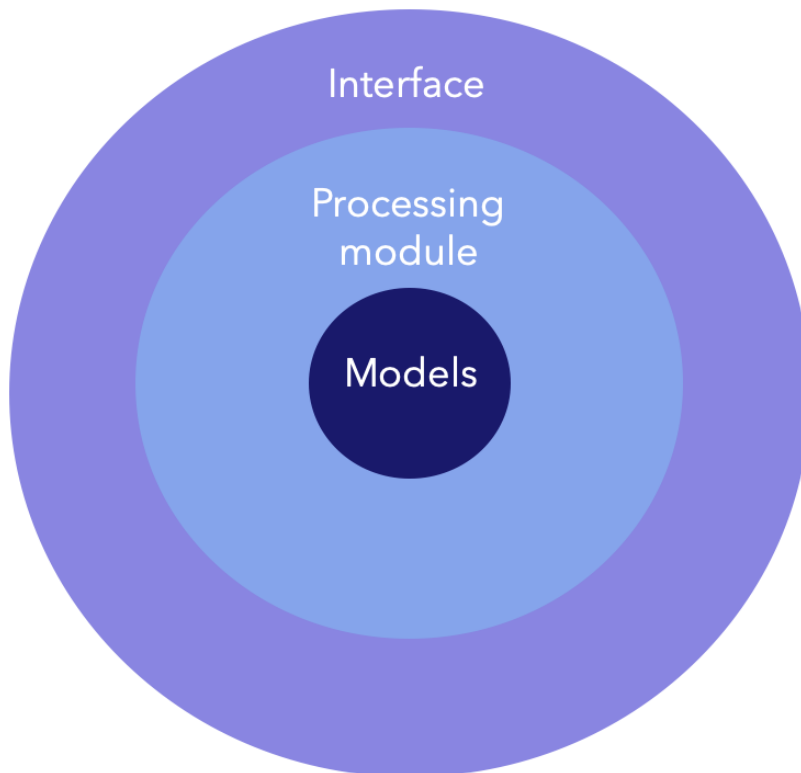


Рисунок 2.7 Пошарова архітектура

2.3 Конструювання програмного забезпечення

Архітектура файлів тренування CNN моделей зображена на рисунку 2.8, рисунку 2.9, рисунку 2.10 та рисунку 2.11

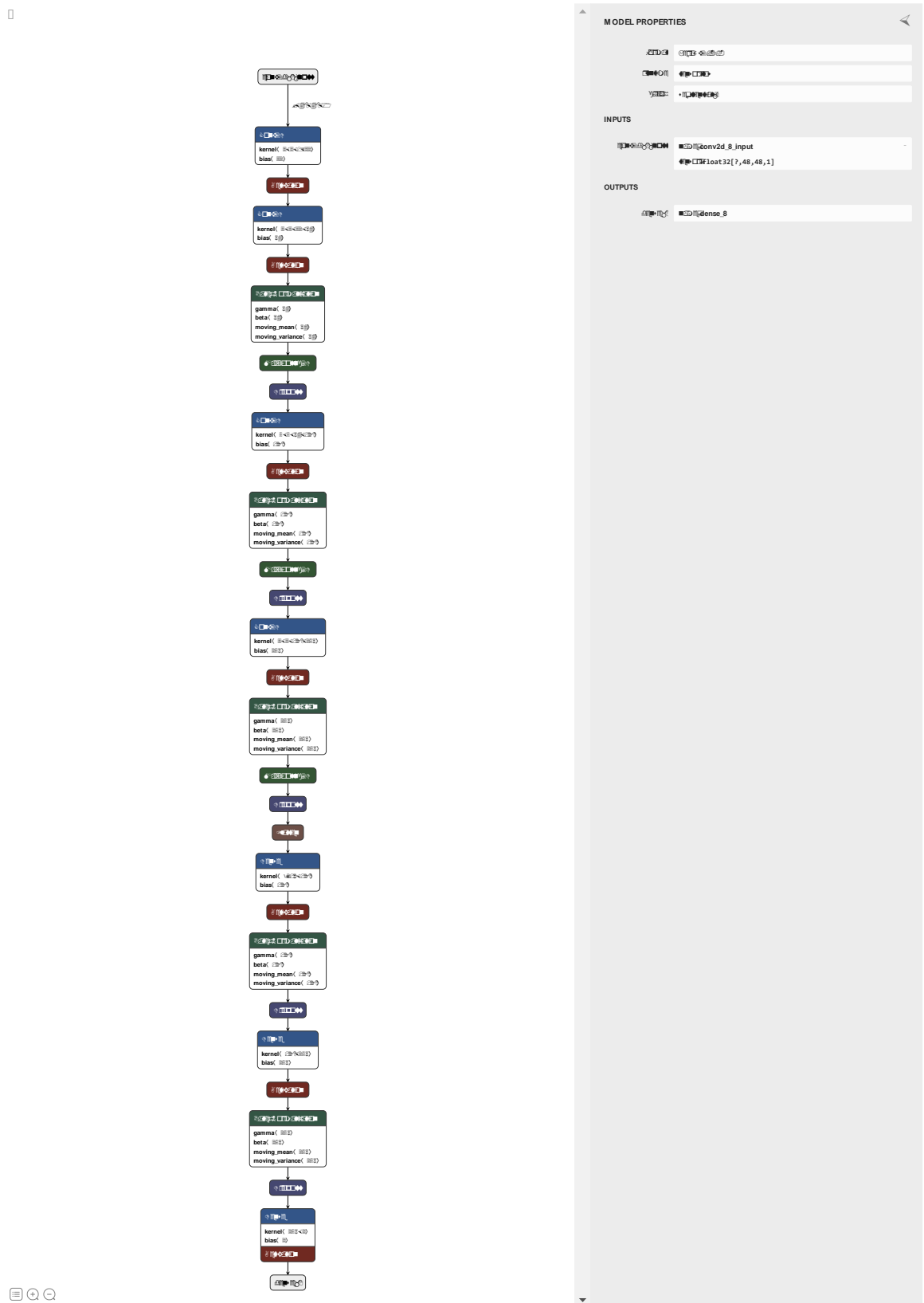
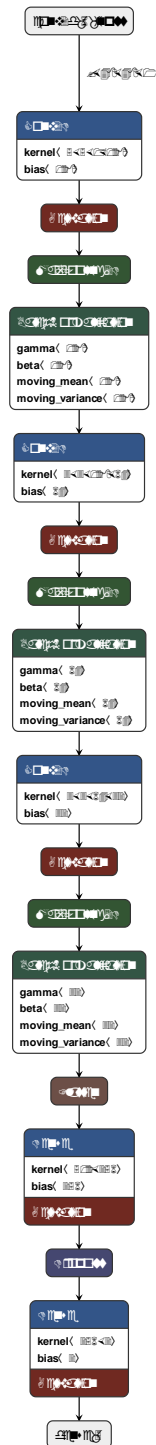


Рисунок 2.8 Зображення архітектури emotion_class_model.h5



MODEL PROPERTIES

Model Name: smile_real_fake_model.h5

Model Type: Convolutional Neural Network

Version: 1.0.0

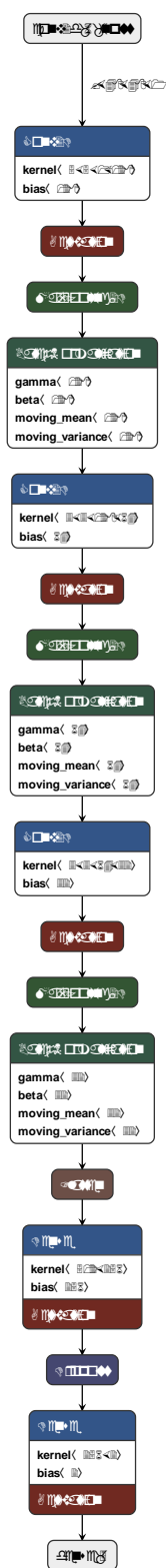
INPUTS

conv2d_6_input
Float32[?, 48, 48, 1]

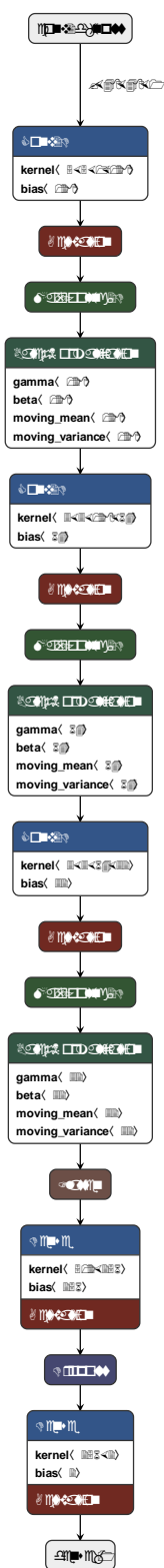
OUTPUTS

dense_5

Рисунок 2.9 Зображення архітектури smile_real_fake_model.h5



34



MODEL PROPERTIES

Model: sad_real_fake_model

Input: conv2d_input

Output: dense_1

INPUTS

conv2d_input
float32[?, 48, 48, 1]

OUTPUTS

dense_1

Рисунок 2.11 Зображення архітектури sad_real_fake_model.h5

2.4 Аналіз безпеки даних

Сканування на вразливості відбувається під час відкриття додатку, ОС використовує спеціалізовані вбудовані інструменти та проводить сканування програмного забезпечення для виявлення потенційних вразливостей.

За допомогою Activity Monitor можна виконувати моніторинг активності для виявлення ненормальної чи підозрілої активності в системі.

Керування доступом до файлів видається як для користувача, тож доступ до видалення обов'язкових пакетів закритий

Висновки до розділу

В цьому розділі були розглянуті виконані ключові етапи проекту: застосовано BPMN моделі для опису бізнес-процесів, зокрема обробки зображень та відео. Ці моделі слугують інструментом для чіткої візуалізації та аналізу внутрішніх процесів системи. Також було представлено детальну діаграму класів, загальну діаграму класів, діаграму компонентів, діаграму взаємодії та пошарову архітектуру проекту. Ці моделі надають глибокий інсайт у взаємозв'язки та структуру програмного забезпечення. Крім того, було показано архітектуру файлів для тренування CNN моделей, що важливо для розуміння процесу конструювання та навчання моделей.

Також описано заходи щодо забезпечення безпеки даних, такі як сканування на вразливості при запуску додатку, моніторинг активності через Activity Monitor та обмеження доступу до важливих файлів.

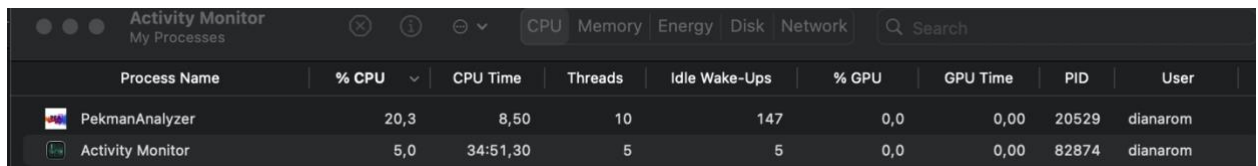
Можемо зазначити, що впроваджені моделі, архітектурні рішення та заходи з безпеки дозволяють ефективно управляти та розвивати програмне забезпечення RekmanAnalyzer, забезпечуючи його стабільність та функціональність.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

Аналіз якості програмного забезпечення є важливим етапом в процесі розробки та управління продуктом. Проведемо детальний аналіз якості ПЗ за допомогою певних метрик та критеріїв.

Однією з ключових метрик є ефективність використання ресурсів та продуктивність додатку. За допомогою інструментів моніторингу ресурсів системи будемо вимірювати завантаження процесора, використання оперативної пам'яті та частоту виконання операцій(рисунок 3.1).



Activity Monitor								
My Processes								
CPU Memory Energy Disk Network Search								
Process Name	% CPU	CPU Time	Threads	Idle Wake-Ups	% GPU	GPU Time	PID	User
PekmanAnalyzer	20,3	8,50	10	147	0,0	0,00	20529	dianarom
Activity Monitor	5,0	34:51,30	5	5	0,0	0,00	82874	dianarom

Рисунок 3.1 Activity Monitor

Після проведення аналізу через Activity Monitor отримані такі результати для додатку PekmanAnalyzer:

- Середнє використання CPU складає 20,3%, що є прийнятним показником для програмного забезпечення. Вказує на те, що програма ефективно використовує ресурси процесора.
- CPU Time: Всього часу процесора, витраченого додатком, під час обробки зображення становить 8,50 секунди. Це також показник ефективності, оскільки він вказує на те, що додаток забирає ресурси протягом обмеженого часу.
- Відсоток використання графічного процесора становить 0,0%, що свідчить про те, що додаток не вимагає значних графічних ресурсів.
- Час використання графічного процесора дорівнює 0,00 секунд. Це підтверджує низьку вимогливість додатку до графічних ресурсів.

Тож додаток PekmanAnalyzer демонструє ефективне використання ресурсів CPU, не справляючи значного навантаження на графічний процесор.

Такі результати свідчать про добру оптимізацію програми та гарну продуктивність.

Додаток виявив високу стійкість до помилок, адже в ході моніторингу не виникало критичних ситуацій, які можуть призвести до відмови системи. PekmanAnalyzer ефективно реагує на виклики користувачів та взаємодіє з системними ресурсами. При використанні додатку виявлено низький рівень затримок та швидку реакцію на команди.

Аналізуючи надійність та стабільність PekmanAnalyzer, можна визначити, що програмне забезпечення виявляє достатній рівень стабільності та надійності у випадках нормального функціонування. Відсутність критичних помилок свідчить про добре спроектований та реалізований код. Усе це сприяє позитивному користувацькому досвіду та забезпечує надійну роботу додатку.

Інтерфейс додатку можна вважати простим для сприйняття користувача та інтуїтивно зрозумілим, оскільки він не містить великої кількості елементів

3.2 Опис процесів тестування

Було виконане мануальне тестування програмного забезпечення, опис відповідних тестів наведено у таблицях 3.1 – 3.26.

Таблиця 3.1 – Тест 1.1

Тест	Запуск додатку
Модуль	Запуск додатку
Номер тесту	1.1
Початковий стан системи	Користувач знаходиться на сторінці вибору типу файлу
Вхідні данні	Результат вибору з radio button
Опис проведення тесту	Користувач натискає на кнопку «далі»
Очікуваний результат	Кнопка не натискається, оскільки radio button не був відмічений

Продовження таблиці 3.1

Фактичний результат	Кнопка не натискається, оскільки radio button не був відмічений
---------------------	---

Таблиця 3.2 – Тест 1.2

Тест	Запуск додатку
Модуль	Запуск додатку
Номер тесту	1.2
Початковий стан системи	Користувач знаходиться на сторінці вибору типу файлу
Вхідні данні	Результат вибору з radio button
Опис проведення тесту	Користувач обирає «image» та натискає кнопку «далі»
Очікуваний результат	Перехід до вікна завантаження зображення
Фактичний результат	Перехід до вікна завантаження зображення

Таблиця 3.3 – Тест 1.3

Тест	Запуск додатку
Модуль	Запуск додатку
Номер тесту	1.3
Початковий стан системи	Користувач знаходиться на вікні вибору типу файлу
Вхідні данні	Результат вибору з radio button
Опис проведення тесту	Користувач обирає «video» та натискає кнопку «далі»

Продовження таблиці 3.3

Очікуваний результат	Відкриття вікна завантаження відео
Фактичний результат	Відкриття вікна завантаження відео

Таблиця 3.4 – Тест 1.4

Тест	Запуск додатку
Модуль	Запуск додатку
Номер тесту	1.4
Початковий стан системи	Користувач знаходиться на сторінці вибору типу файлу
Вхідні данні	Результат вибору з radio button
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.5 – Тест 1.5

Тест	Робота кнопки «Next» (Далі) на вікні завантаження зображення до завантаження
Модуль	Завантаження зображення
Номер тесту	1.5
Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	Немає

Продовження таблиці 3.5

Опис проведення тесту	Користувач натискає кнопку «Next» (Далі)
Очікуваний результат	Кнопка неактивна
Фактичний результат	Кнопка неактивна

Таблиця 3.6 – Тест 1.6

Тест	Робота кнопки «Next» (Далі) на вікні завантаження відео до завантаження
Модуль	Завантаження відео
Номер тесту	1.6
Початковий стан системи	Користувач знаходиться на вікні завантаження відео
Вхідні данні	Немає
Опис проведення тесту	Користувач натискає кнопку «Next» (Далі)
Очікуваний результат	Кнопка неактивна
Фактичний результат	Кнопка неактивна

Таблиця 3.7 – Тест 1.7

Тест	Завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.7

Продовження таблиці 3.7

Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	Зображення розміру більше 10МБ
Опис проведення тесту	Користувач завантажує зображення
Очікуваний результат	Вивід помилки "Error", "The maximum image size should be less than 10 MB."
Фактичний результат	Вивід помилки "Error", "The maximum image size should be less than 10 MB."

Таблиця 3.8 – Тест 1.8

Тест	Завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.8
Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	Файл що не є зображенням
Опис проведення тесту	Користувач завантажує невідповідний файл
Очікуваний результат	Вивід помилки "Error", "Only upload images!"
Фактичний результат	Вивід помилки "Error", "Only upload images!"

Таблиця 3.9 – Тест 1.9

Тест	Завантаження відео
Модуль	Завантаження відео
Номер тесту	1.9
Початковий стан системи	Користувач знаходиться на вікні завантаження відео
Вхідні данні	Відео розміру більше 100МБ
Опис проведення тесту	Користувач завантажує відео
Очікуваний результат	Вивід помилки "Error", "The maximum video size should be less than 100 MB."
Фактичний результат	Вивід помилки "Error", "The maximum video size should be less than 100 MB."

Таблиця 3.10 – Тест 1.10

Тест	Завантаження відео
Модуль	Завантаження відео
Номер тесту	1.10
Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	Файл що не є відео
Опис проведення тесту	Користувач завантажує невідповідний файл
Очікуваний результат	Вивід помилки "Error", "Only upload videos!"

Продовження таблиці 3.10

Фактичний результат	Вивід помилки "Error", "Only upload videos!"
---------------------	--

Таблиця 3.11 – Тест 1.11

Тест	Робота кнопки «Back» (Назад) на вікні завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.11
Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	
Опис проведення тесту	Користувач натискає на кнопку «Back» (Назад)
Очікуваний результат	Повернення на перше вікно програми
Фактичний результат	Повернення на перше вікно програми

Таблиця 3.12 – Тест 1.12

Тест	Робота кнопки «Back» (Назад) на вікні завантаження відео
Модуль	Завантаження відео
Номер тесту	1.12
Початковий стан системи	Користувач знаходиться на вікні завантаження відео
Вхідні данні	

Продовження таблиці 3.12

Опис проведення тесту	Користувач натискає на кнопку «Back» (Назад)
Очікуваний результат	Повернення на перше вікно програми
Фактичний результат	Повернення на перше вікно програми

Таблиця 3.13 – Тест 1.13

Тест	Робота кнопки «Next» (Далі) на вікні завантаження зображення після завантаження
Модуль	Завантаження зображення
Номер тесту	1.13
Початковий стан системи	Користувач знаходиться на вікні завантаження зображення
Вхідні данні	Зображення, що пройшло перевірку
Опис проведення тесту	Користувач натискає на кнопку «Next» (Далі)»
Очікуваний результат	Перехід до вікна відображення процесу обробки зображення
Фактичний результат	Перехід до вікна відображення процесу обробки зображення

Таблиця 3.14 – Тест 1.14

Тест	Робота кнопки «Next» (Далі) на вікні завантаження відео після завантаження
Модуль	Завантаження відео
Номер тесту	1.14
Початковий стан системи	Користувач знаходиться на вікні завантаження відео
Вхідні данні	Відео, що пройшло перевірку
Опис проведення тесту	Користувач натискає на кнопку «Next» (Далі)»
Очікуваний результат	Перехід до вікна відображення процесу обробки відео

Таблиця 3.15 – Тест 1.15

Тест	Закриття вікна завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.15
Початковий стан системи	Користувач знаходиться на сторінці завантаження зображення
Вхідні данні	
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.16 – Тест 1.16

Тест	Закриття вікна завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.16
Початковий стан системи	Користувач знаходиться на сторінці завантаження зображення
Вхідні данні	Зображення, що пройшло перевірку
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.17 – Тест 1.17

Тест	Закриття вікна завантаження зображення
Модуль	Завантаження зображення
Номер тесту	1.17
Початковий стан системи	Користувач знаходиться на сторінці завантаження зображення
Вхідні данні	
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.18 – Тест 1.18

Тест	Закриття вікна завантаження відео
Модуль	Завантаження зображення
Номер тесту	1.18
Початковий стан системи	Користувач знаходиться на сторінці завантаження відео
Вхідні данні	Відео, що пройшло перевірку
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.19 – Тест 1.19

Тест	Вікно відображення прогресу обробки
Модуль	Вікно відображення прогресу обробки
Номер тесту	1.19
Початковий стан системи	Користувач знаходиться на сторінці відображення прогресу обробки
Вхідні данні	Файл був переданий в модуль аналізу
Опис проведення тесту	Аналіз вигляду вікна
Очікуваний результат	Відображення бігунка обробки файлу
Фактичний результат	Відображення бігунка обробки файлу

Таблиця 3.20 – Тест 1.20

Тест	Закриття вікна відображення прогресу обробки
Модуль	Вікно відображення прогресу обробки
Номер тесту	1.20
Початковий стан системи	Користувач знаходиться на сторінці відображення прогресу обробки
Вхідні данні	
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

Таблиця 3.21 – Тест 1.21

Тест	Перегляд результату аналізу зображення
Модуль	Вікно відображення результату
Номер тесту	1.21
Початковий стан системи	Користувач знаходиться на вікні відображення результату
Вхідні данні	Результат був переданий з модуля аналізу
Опис проведення тесту	Користувач натискає на кнопку «View result» (Переглянути результат)
Очікуваний результат	Відображення обробленого зображення з написом результату аналізу
Фактичний результат	Відображення обробленого зображення з написом результату аналізу

Таблиця 3.22 – Тест 1.22

Тест	Збереження результату аналізу зображення
Модуль	Вікно відображення результату
Номер тесту	1.22
Початковий стан системи	Користувач знаходиться на вікні відображення результату
Вхідні данні	Результат був переданий з модуля аналізу
Опис проведення тесту	Користувач натискає на кнопку «Save result» (Зберегти результат)
Очікуваний результат	Надання користувачу можливості вибору шляху збереження обробленого зображення з написом результату аналізу
Фактичний результат	Надання користувачу можливості вибору шляху збереження обробленого зображення з написом результату аналізу

Таблиця 3.23 – Тест 1.23

Тест	Перегляд результату аналізу відео
Модуль	Вікно відображення результату
Номер тесту	1.23
Початковий стан системи	Користувач знаходиться на вікні відображення результату
Вхідні данні	Результат був переданий з модуля аналізу
Опис проведення тесту	Користувач натискає на кнопку «View result» (Переглянути результат)
Очікуваний результат	Відображення обробленого відео з по-кадровою рамкою результату аналізу

Продовження таблиці 3.23

Фактичний результат	Відображення обробленого відео з по-кадровою рамкою результату аналізу
---------------------	--

Таблиця 3.24 – Тест 1.24

Тест	Перегляд результату аналізу відео
Модуль	Вікно відображення результату
Номер тесту	1.24
Початковий стан системи	Користувач знаходиться на вікні відображення результату
Вхідні данні	Результат був переданий з модуля аналізу
Опис проведення тесту	Користувач натискає на кнопку «View result» (Переглянути результат)
Очікуваний результат	Відображення обробленого відео з по-кадровою рамкою результату аналізу
Фактичний результат	Відображення обробленого відео з по-кадровою рамкою результату аналізу

Таблиця 3.25 – Тест 1.25

Тест	Робота кнопки «Back» (Назад) на вікні відображення результату
Модуль	Вікно відображення результату
Номер тесту	1.25
Початковий стан системи	Користувач знаходиться на вікні відображення результату
Вхідні данні	

Продовження таблиці 3.25

Опис проведення тесту	Користувач натискає на кнопку «Back» (Назад)
Очікуваний результат	Повернення на перше вікно програми
Фактичний результат	Повернення на перше вікно програми

Таблиця 3.26 – Тест 1.26

Тест	Закриття вікна відображення результату
Модуль	Вікно відображення результату
Номер тесту	1.26
Початковий стан системи	Користувач знаходиться на сторінці відображення результату
Вхідні данні	
Опис проведення тесту	Натиснути значок «х» для закриття вікна
Очікуваний результат	Додаток припинив роботу
Фактичний результат	Додаток припинив роботу

3.3 Опис контрольного прикладу

Користувач відкриває додаток подвійним кліком на іконку(рисунок 3.2).

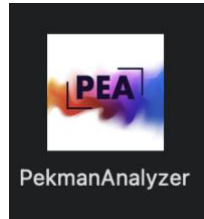


Рисунок 3.2 Іконка додатку PekmanAnalyzer

Після чого відкривається програма PekmanAnalyzer(рисунок 3.3).

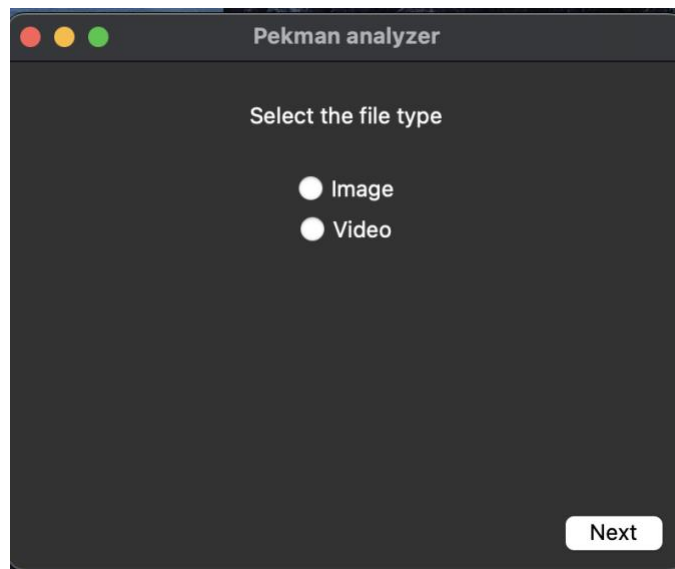


Рисунок 3.3 Початкове вікно вибору типу файлу

Користувачу необхідно обрати який тип файлу він буде завантажувати(рисунок 3.4)

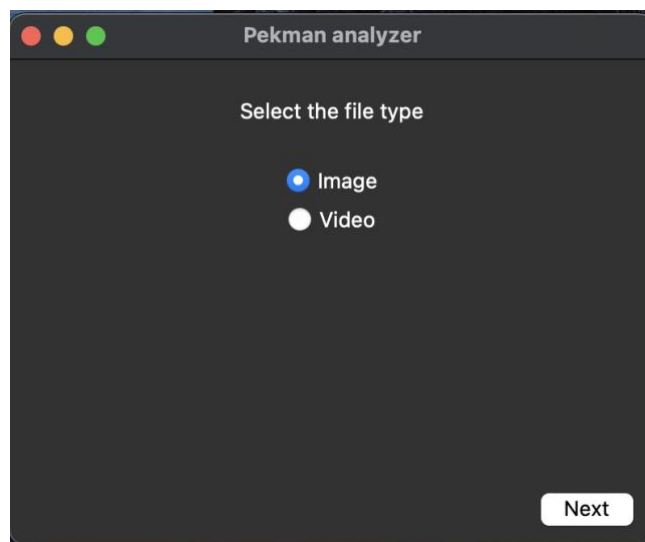


Рисунок 3.4 Вибір типу файлу

Далі користувач переходить на вікно завантаження зображення(рисунок 3.5).



Рисунок 3.5 Вікно завантаження зображення

Та завантажує фото на якому зображене обличчя(рисунок 3.6)(рисунок 3.7)(рисунок 3.8).

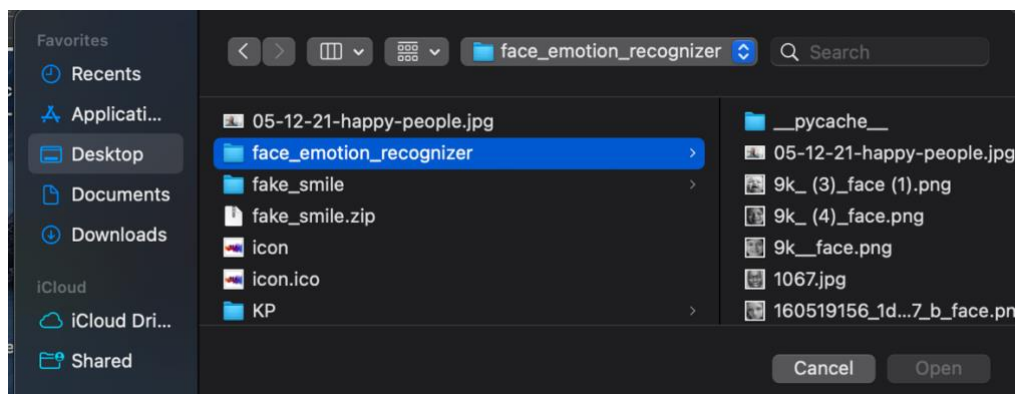


Рисунок 3.6 Вікно вибору зображення



Рисунок 3.7 Приклад вибору зображення

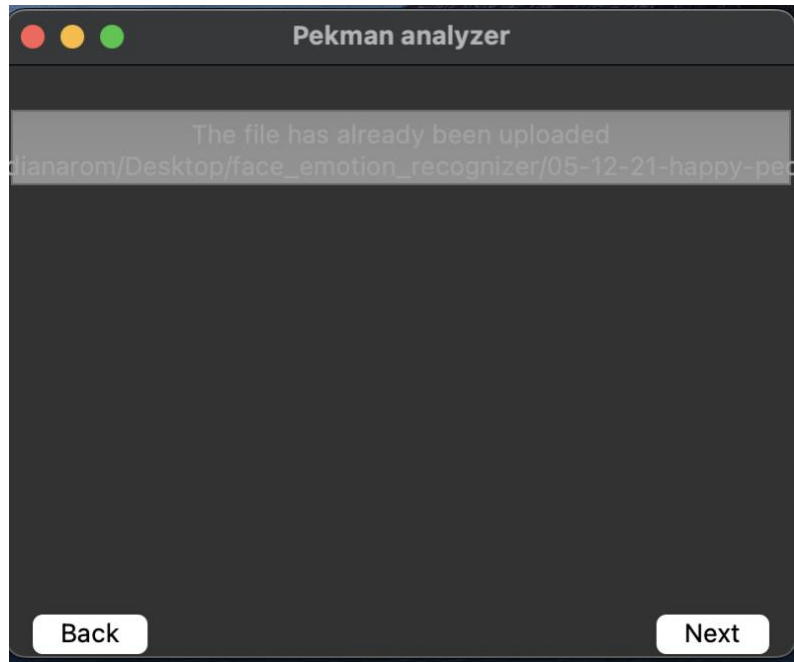


Рисунок 3.8 Відображення успішного завантаження зображення

Якщо користувач намагався завантажити файл що не відповідає розширенням .PNG, .JPEG та .IMEG, то відобразиться повідомлення про помилку(рисунок 3.9)

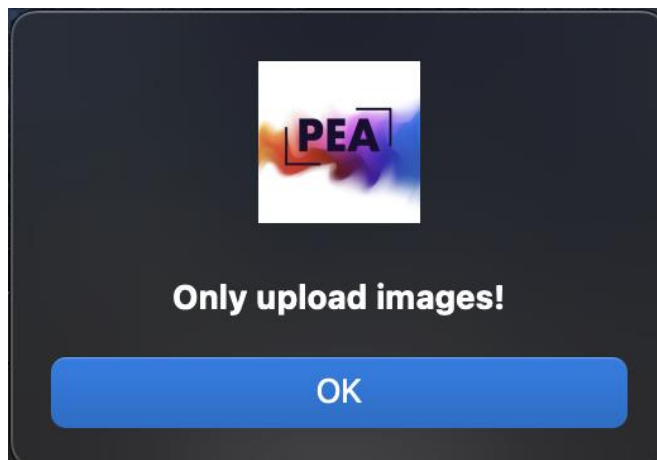


Рисунок 3.9 Помилка завантаження

Якщо користувач передумав і хоче обрати інший тип файлу завантаження, він може повернутись на початкове вікно, натиснувши кнопку «Back» (Назад) та обрати «Video» (Відео) (рисунок 3.10).

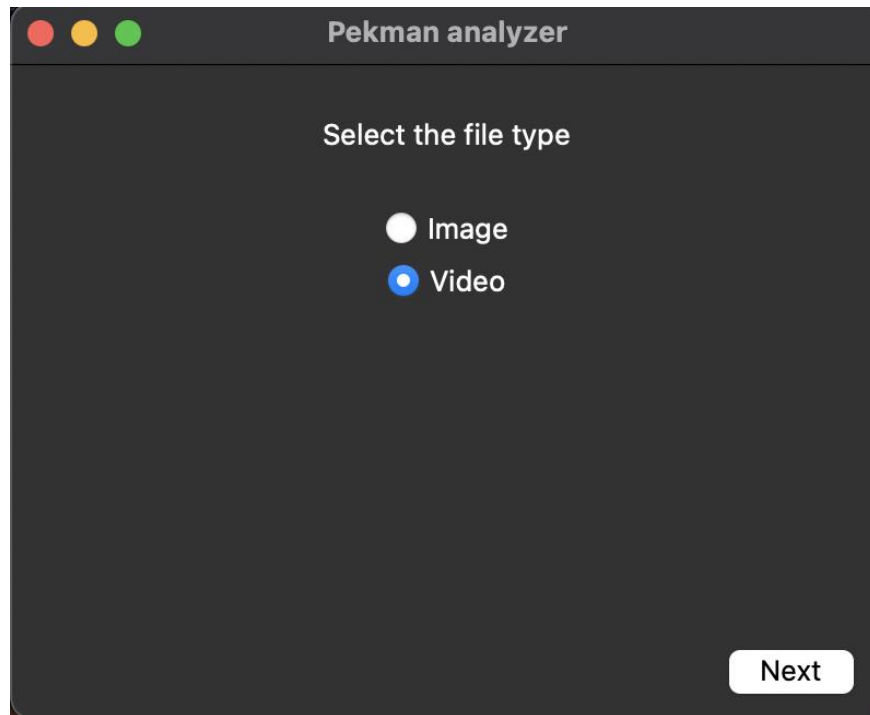


Рисунок 3.10 Вибір типу файлу

Після чого перейти до вікна завантаження відео(рисунок 3.11)

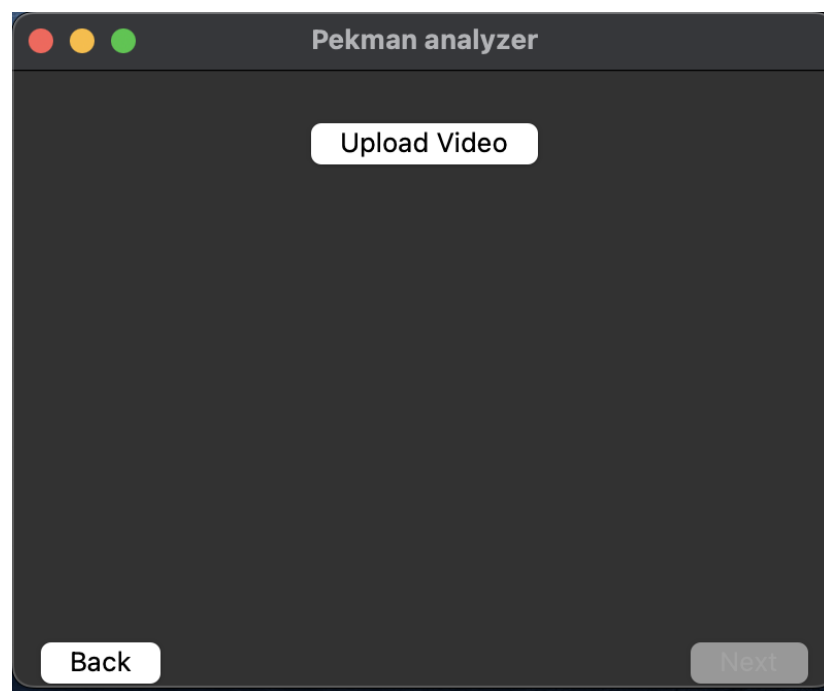


Рисунок 3.11 Вікно завантаження відео

Якщо користувач завантажив файл з розширенням .MP4, .MOV або .AVI розміром до 100МБ, то відобразить успішне завантаження файлу(рисунок 3.12).

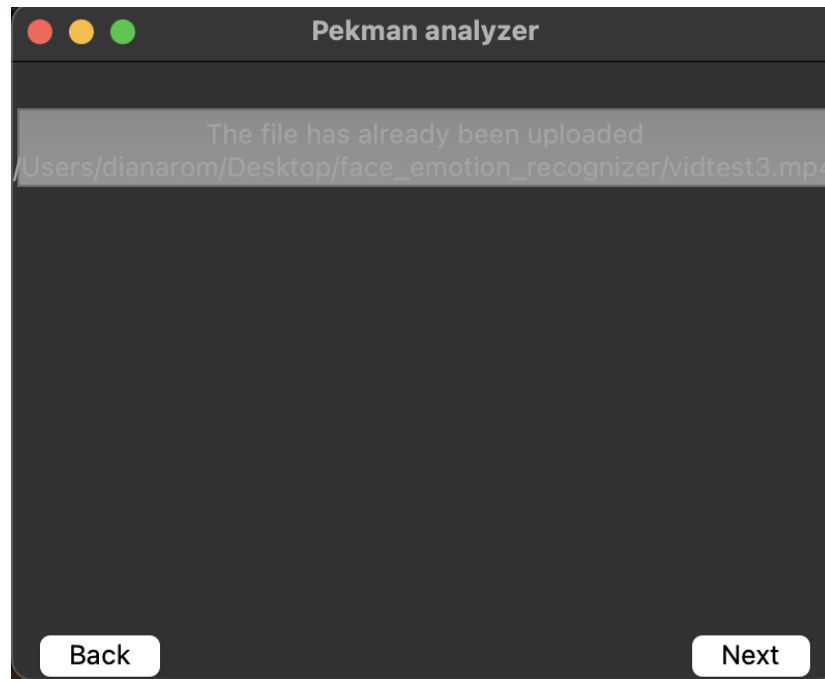


Рисунок 3.12 Успішне завантаження відео-файлу

Якщо користувач порушив обмеження по типу файлу або його розміру, будуть відображені наступні повідомлення про помилку(рисунок 3.13, 3.14).

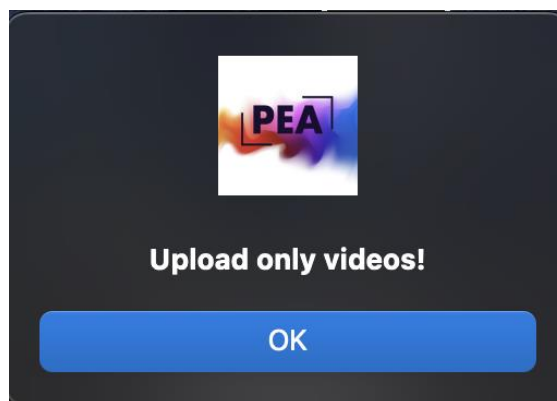


Рисунок 3.13 Повідомлення про помилку формату файлу

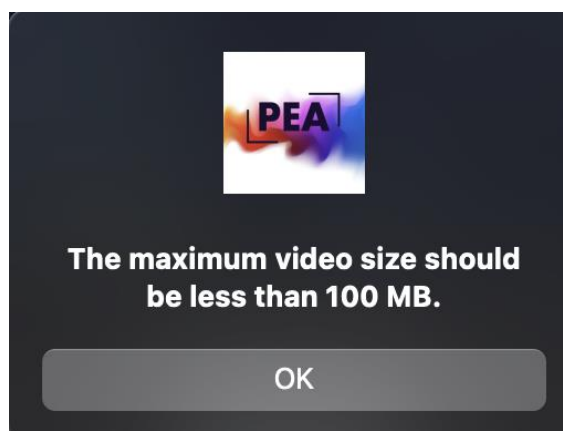


Рисунок 3.14 Повідомлення про помилку перевищення допустимого розміру файлу

Якщо користувач успішно завантажив фото або відео, відбувається аналіз файлу та відображається прогрес обробки(рисунок 3.15).

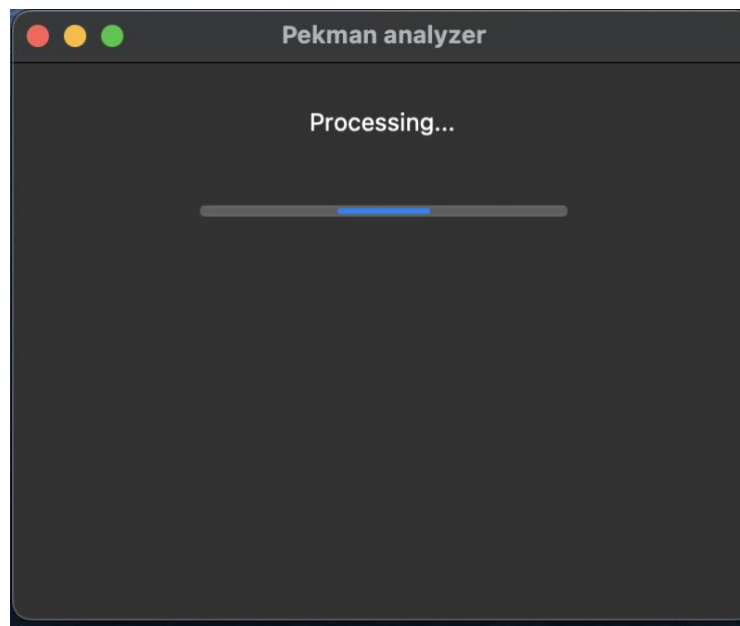


Рисунок 3.15 Відображення вікна прогресу обробки

Після успішного завершення аналізу файлу можемо переглянути результат(рисунок 3.16).

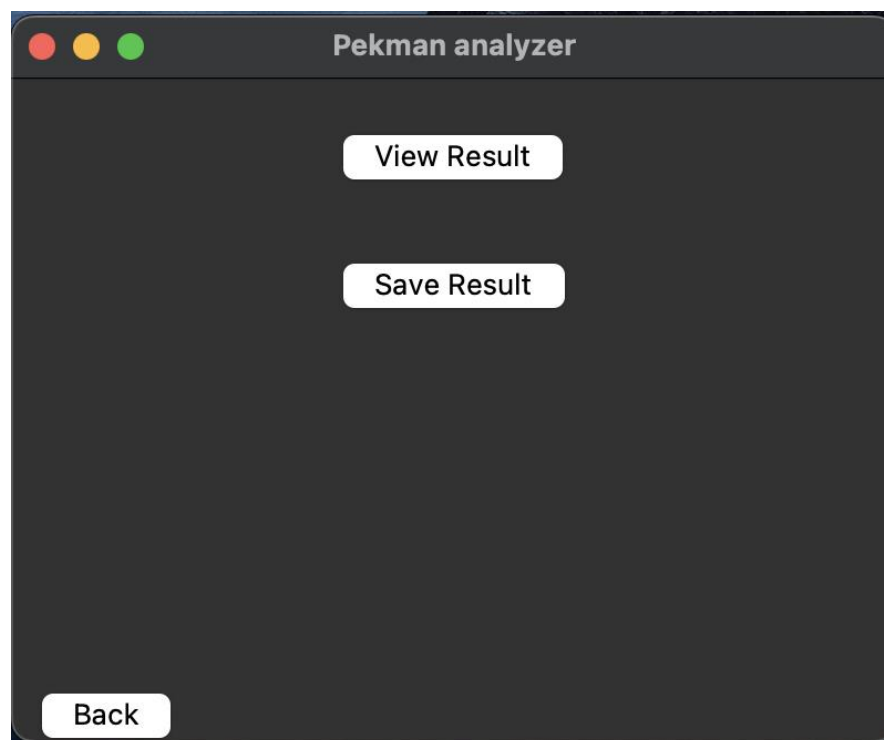


Рисунок 3.16 Вікно відображення результату

Користувач може переглянути оброблене зображення та/або зберегти його (рисунок 3.17)(рисунок 3.18).

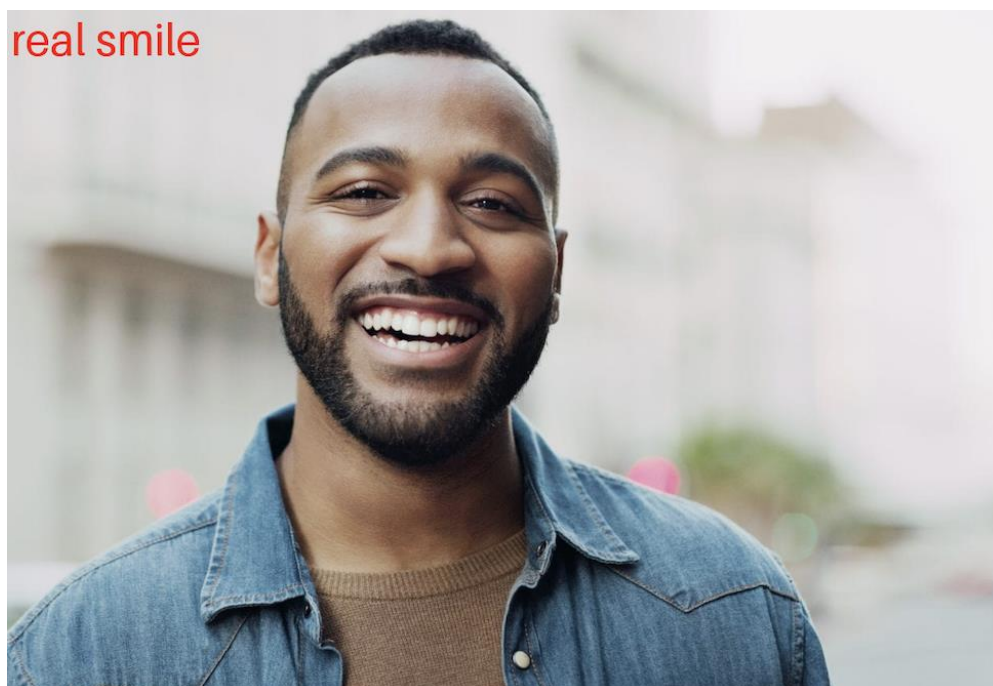


Рисунок 3.17 Приклад результату обробки зображення

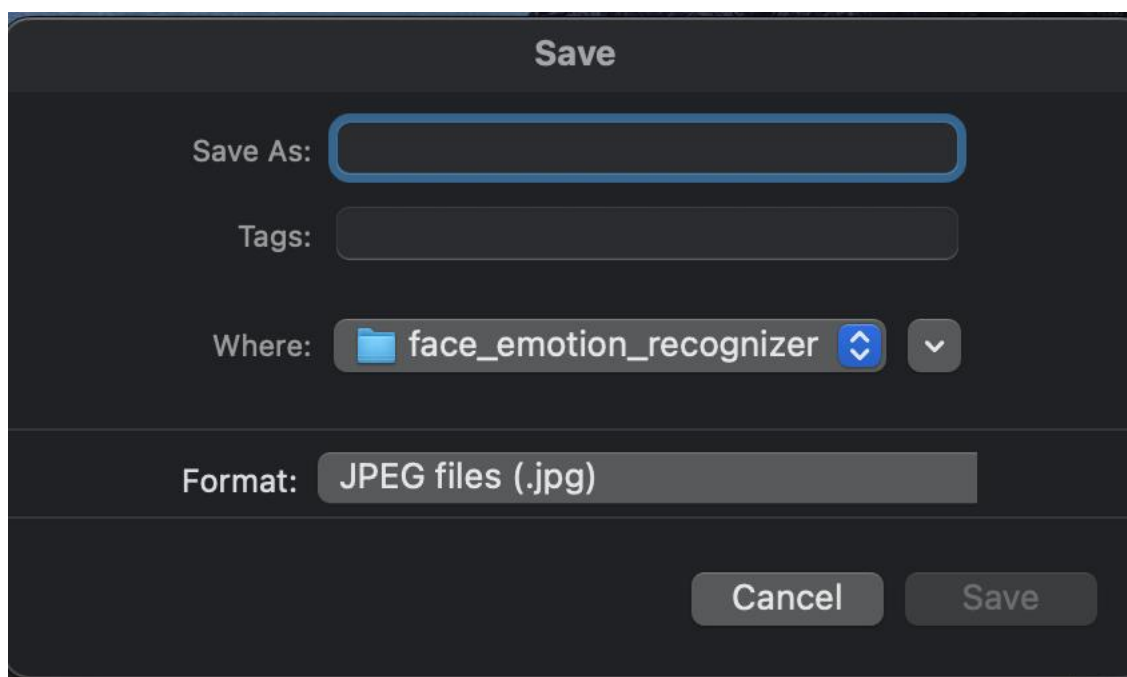


Рисунок 3.18 Вибір шляху збереження обробленого зображення

Якщо користувач завантажував відео, то він може переглянути оброблене відео та/або зберегти його(рисунок 3.19)(рисунок 3.20).

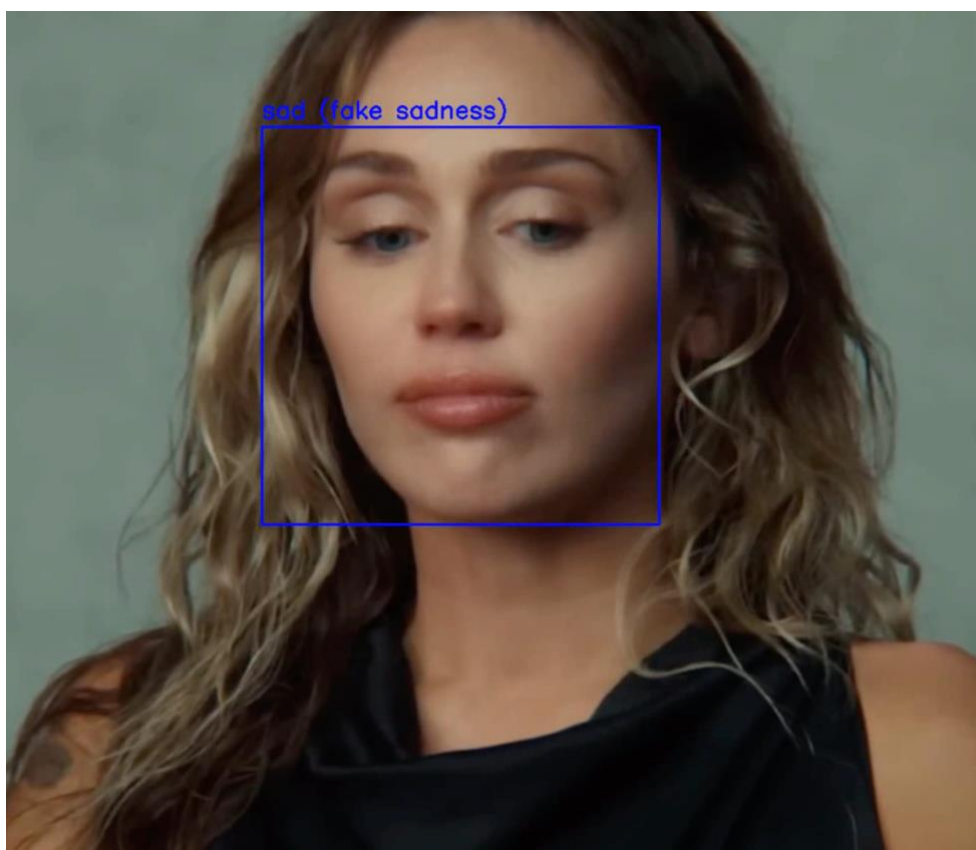


Рисунок 3.19 Приклад результату обробки відео

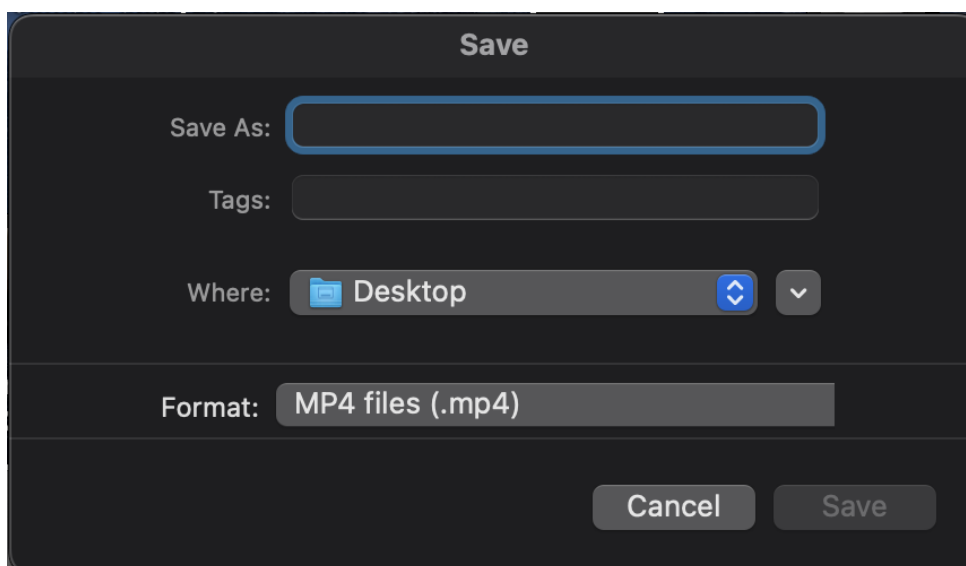


Рисунок 3.20 Вибір шляху збереження обробленого відео

Висновки до розділу

У цьому розділі було проведено дослідження продуктивності та надійності додатку RekmanAnalyzer. Використання ресурсів, таких як центральний процесор, було ефективним, підтверджуючи оптимізований характер програми. Стабільність та надійність виявилися на достатньому рівні, як і демонстрація швидкості реакції на користувацькі введення.

Також в цьому розділі були представлені мануальні тести, що описували різні етапи використання додатку. Контрольний приклад подавав стислий огляд типового сценарію використання, від вибору файлу до завершення обробки та перегляду результатів.

Узагальнено можна зазначити успішне виконання поставлених завдань та достатню якість розробленого додатку.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Розгортання програмного забезпечення

Розгортання додатку починається зі скачування архіву PekmanAnalyzer.zip з папки на гугл диску, посилання на яке вказано в документі How_to_install.txt, який розташований разом з усіма файлами проекту на платформі VCS Github.

Після розпакування архіву необхідно натиснути правою кнопкою миші на файл додатку, натиснути «Open», оскільки MacOS перевіряє файли від незатверджених розробників, з'явиться повідомлення (рисунок 4.1). Це стандартна процедура безпеки MacOS для файлів від непідтверджених джерел.

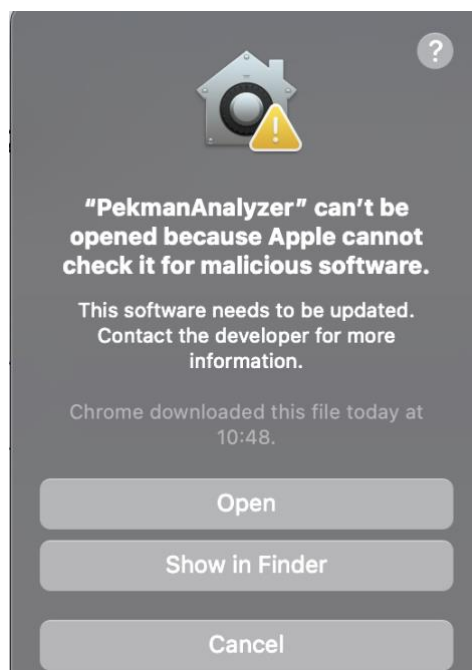


Рисунок 4.1 Попередження

Знову натиснути «Open» і відбудеться відкриття додатку. Користувачу необхідно звернути увагу, що цей додаток підтримується тільки на операційній системі MacOS. Рекомендовано встановлювати на комп'ютерах з процесором Intel Core i7 для оптимальної продуктивності.

4.2 Підтримка програмного забезпечення

Після успішного розгортання додатку PekmanAnalyzer на MacOS-пристрої користувач повинен мати можливість отримати нову версію застосунку з кожною версією. Для автоматизації цього процесу був використаний сервіс GitHub. Щоб забезпечити ефективну підтримку та обслуговування додатку користувач повинен регулярно перевіряти репозиторій PekmanAnalyzer на Github, щоб отримати інформацію про нові версії та оновлення.

В разі наявності нової версії, користувачу треба завантажити архів з оновленою версією у документі How_to_install.txt. Користувачу необхідно перейти по посиланню вказаному у файлі для завантаження та встановлення нової версії. Для кращого розуміння нововведень та змін у новій версії, рекомендується переглянути всі файли проекту, особливо requirements.txt

Цей підхід дозволяє забезпечити користувачу простий та автоматизований спосіб отримання оновлень, зберігаючи продуктивність та функціональність додатку PekmanAnalyzer на достатньому рівні.

Висновки до розділу

У цьому розділі було надано детальний огляд процесу розгортання додатку PekmanAnalyzer на операційній системі MacOS. Зазначено, що розгортання починається зі скачування архіву PekmanAnalyzer.zip з гугл-диску, посилання на який знаходиться у документі How_to_install.txt, розташованому на платформі Github.

Важливим було акцентування, що додаток підтримується тільки на операційній системі MacOS, і встановлення рекомендується на комп'ютерах з процесором Intel Core i7 для оптимальної продуктивності.

Крім того, у цьому розділі було надано рекомендації щодо підтримки програмного забезпечення. Користувачу рекомендувалось періодично перевіряти репозиторій на Github для отримання інформації про нові версії та оновлення. Також зазначена покрокова інструкція для отримання нової версії додатку.

ВИСНОВКИ

У даній курсовій роботі було визначено завдання розробки застосунку, який використовує згорткові нейронні мережі для розпізнавання емоцій та їх фальсифікації, таким чином виявляючи брехню в наданих користувачем для описаного аналізу файлах. Процес включає в себе завантаження коротких відео чи зображень обличчя, уривків інтерв'ю, використання CNN-моделей для аналізу емоцій та перевірки на фальшивість експресії.

Для покращення результатів та оптимізації алгоритмів, в тренуванні моделей були використані сучасні активаційних функції, таких як mish, а також архітектура окремих моделей була вдосконалена та підібрана для навчання нейронних мереж. Було використано загальнодоступний датасет FER-2013, що класично використовується для тренування моделі розпізнавання емоцій, проте також було створено власний – для тренування моделі класифікації правдивих та неправдивих емоцій, які виокремлюються в науковій книзі «Теорія брехні» Пола Екмана та, власне, на цих принципах заснована логіка побудови нового датасету. Для попередньої обробки зображень, порівняно із застарілими методами, такими як алгоритм Віоли-Джонса, вибір був зроблений на користь Dlib для більш ефективного розпізнавання обличчя.

Основний фокус дослідження був на досягненні оптимального балансу між точністю розпізнавання та швидкістю обробки, що важливо для практичного використання нейронних мереж.

Дана курсова робота є прикладом успішної імплементації принципів психології в поєднанні з нейронними мережами. Застосування може охоплювати навчання осіб з алекситимією, підтримку результатів поліграфічних вимірювань та вирішення задач у сфері психології та правосуддя.

Можна відзначити, що результати дослідження та розробки курсової роботи дозволили створити функціональний застосунок, спрямований на розпізнавання емоцій та виявлення навмисно набутих емоцій, що є свідченням брехні, у поданих користувачем матеріалах.

Після реалізації застосунку він був протестований на пристроях з MacOS, але різними версіями процесорів intel7 та M1, що засвідчило коректного відображення додатку на різних пристроях.

Наукова новизна роботи полягає в наступному (достатньо вказати щось одне).

Вперше:

- Реалізовано нейронні мережі для розпізнавання правдивих і підроблених емоцій на підґрунті психолога та професора Пола Екмана;

Модифіковано:

- Архітектуру CNN для розпізнавання емоцій шляхом експериментального підходу пошуку більш ефективної для тренування кількості шарів, фільтрів та епох навчання.

Набуло подальший розвиток:

- Розвиток використання AI в сфері психології.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Екман, П. (Пол). "Теорія брехні" [Паперовий ресурс]
- 2) Kumar, G. A. R., Kumar, R. K., & Sanyal, G. (2017). "Discriminating real from fake smile using convolution neural network" [Електронний ресурс] // 2017 International Conference on Computational Intelligence in Data Science (ICCIDS). DOI: 10.1109/iccids.2017.827265
- 3) Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features" [Електронний ресурс] // Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 1. IEEE, 2001
- 4) Viola, Paul, and Michael J. Jones. "Robust real-time face detection" [Електронний ресурс] // International journal of computer vision 57.2 (2004): 137-154.
- 5) torcheeg. "torcheeg.models.MTCNN" [Електронний ресурс]. - Режим доступу до ресурсу:
<https://torcheeg.readthedocs.io/en/stable/generated/torcheeg.models.MTCNN.html>
- 6) Huynh, C. P., Cheung, N. M., & Lai, V. T. (2017). "Discrimination Between Genuine and Posed Smiles Using Deep Neural Networks" [Електронний ресурс] // ICCV 2017 Workshop on Multimedia for Good. - Режим доступу до ресурсу:
https://openaccess.thecvf.com/content_ICCV_2017_workshops/papers/w44/Huynh_Discrimination_Between_Genuine_ICCV_2017_paper.pdf

- 7) Mish Activation Function [Электронный ресурс]:
[https://paperswithcode.com/method/mish#:~:text=Mish%20is%20an%20activation%20function,x%20σ%20\(%20x%20\)%20.](https://paperswithcode.com/method/mish#:~:text=Mish%20is%20an%20activation%20function,x%20σ%20(%20x%20)%20.)
- 8) Dlib [Электронный ресурс]:
<https://news.ycombinator.com/item?id=27917159>
- 9) Altair Flux User Guide [Электронный ресурс]:
<https://2021.help.altair.com/2021.2/flux/Flux/Help/english/UserGuide/English/topics/Grille2DDefinitionStructure.html>
- 10) Scikit-Learn Accuracy Score Documentation [Электронный ресурс]:
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
- 11) FER2013 Kaggle Dataset [Электронный ресурс]:
<https://www.kaggle.com/datasets/msmbare/fer2013>
- 12) IBM Convolutional Neural Networks Topic [Электронный ресурс]:
<https://www.ibm.com/topics/convolutional-neural-networks>
- 13) Machine Learning Libraries Overview [Электронный ресурс]:
<https://www.analyticsvidhya.com/blog/2023/03/machine-learning-libraries/>
- 14) OpenCV Official Website [Электронный ресурс]: <https://opencv.org/>
- 15) Google Colab vs. Jupyter Notebook Comparative Analysis [Электронный ресурс]: https://medium.com/@navneetsingh_95791/google-colab-vs-jupyter-notebook-for-tensorflow-machine-learning-a-comparative-analysis-e9861af38916