

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

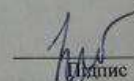
Галузь знань \_\_\_\_\_ 12 – Інформаційні технології \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 –Комп'ютерна інженерія \_\_\_\_\_

на тему «Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору»

КвРКІП. 301153.23.01.48 ПЗ

Виконав: студент 2 курсу, група КІ2м-23-1

  
Підпис

Тимур ІСАЄВ  
Ініціали, прізвище

Керівник \_\_\_\_\_ к.ф.-м.н., доцент  
Науковий ступінь, вчене звання

  
Підпис

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КІС, доктор філософії, доцент

Ольга ПАВЛОВА

29 04 2025 р.

Хмельницький, 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень МАГІСТР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 01 ” 09 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА**

Тимуру ІСАСВУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору

Керівник проекту (роботи) Тетяна КИСІЛЬ, к.ф.-м.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 01.05.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз існуючих рішень у сфері розпізнавання емоцій людини

Застосування згорткових нейронних мереж для розпізнавання емоцій людини

Метод та алгоритм розпізнавання виразів обличчя людини

Результати тренування моделі та способи їх покращення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи магістра

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Сергій ЛИСЕНКО, професор кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 01 » 09 2024р.

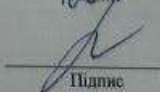
**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики КвРМ з керівником	01.09.2024	Виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.10.2024	Виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	01.11.2024	Виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.12.2024	Виконано
5	Робота над науковою статтею	01.02.2025	Виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2025	Виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2025	Виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2025	Виконано
9	Попередній захист ДРМ	29.04.2025	Виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2025	

Студент

Керівник роботи

  
Підпис

  
Підпис

Тимур ІСАЄВ  
Ініціали, прізвище

Тетяна КИСІЛЬ  
Ініціали, прізвище

## РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору.

Автор роботи: Ісаєв Тимур Сергійович

Керівник роботи: Кисіль Тетяна Миколаївна

Пояснювальна записка: 84 с., 50 рис., 20 табл., 3 дод., 82 джерела.

РОЗПІЗНАВАННЯ ЕМОЦІЙ, КОМП'ЮТЕРНИЙ ЗІР, TENSORFLOW LITE, RASPBERRY PI, CORAL USB, EMOTION, FER2013 .

Об'єктом дослідження є підвищення продуктивності роботи моделей штучної нейронної мережі на пристроях з обмеженою потужністю.

Предметом дослідження є застосування нейронної мережі для створення нового методу розпізнавання емоцій.

Метою кваліфікаційної роботи магістра є створення методу розпізнавання емоцій людини у реальному часі, що працюватиме ефективно на мобільних пристроях.

Щоб виконати мету дослідження було:

- створено власний датасет для підвищення точності розпізнавання емоцій;
- розроблено власні методи розпізнавання;
- натреновано нейронну мережу на власному датасеті;
- покращено результати моделі внаслідок використання прискорювача.

Наукова новизна отриманих результатів полягає в удосконаленні методу оптимізації нейронних мереж для задач розпізнавання емоцій, що дозволяє підвищити їхню продуктивність та забезпечити ефективне функціонування на мобільних пристроях з обмеженими ресурсами.

Практична значимість отриманих результатів полягає у імплементації моделі на пристроях з обмеженою потужністю.

За темою дипломної роботи були опубліковані статті:

- у збірнику “Вчені записки” Таврійського національного університету імені В.І. Вернадського, Том 35 (74) №3;
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (С. 15-21);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (С. 37-44).

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....</b>	<b>5</b>
<b>ВСТУП.....</b>	<b>6</b>
<b>1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЕМОЦІЙ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО ЗОРУ.....</b>	<b>8</b>
1.1 Останні досягнення в галузі розпізнавання емоцій .....	8
1.2 Принципи та області застосування комп'ютерного зору для ідентифікації емоцій .....	13
1.3 Аналіз типових помилок у системах розпізнавання емоцій .....	16
1.4 Постановка задачі щодо методу розпізнавання емоцій людини та вибір засобів для реалізації .....	21
1.5 Висновки до першого розділу .....	23
<b>2 ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО СТАНУ .....</b>	<b>24</b>
2.1 Методики корекції помилок у системах визначення емоцій .....	24
2.2 Огляд нейронних мереж, що застосовуються у дослідженні .....	27
2.3 Підготовка набору даних та датасету для тренування мережі .....	31
2.4 Конфігурація нейронної мережі для навчання.....	42
2.5 Висновки до другого розділу .....	47
<b>3 МЕТОДИКА ТА АЛГОРИТМ ВИЗНАЧЕННЯ ЕМОЦІЙ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ.....</b>	<b>49</b>
3.1 Процес тренування мережі для ідентифікації емоцій.....	49
3.2 Алгоритмічні основи тренування мережі для визначення емоцій .....	59
3.3 Висновок до третього розділу .....	68
<b>4 РЕЗУЛЬТАТИ ТРЕНУВАННЯ МОДЕЛІ .....</b>	<b>69</b>

4.1 Аналіз результатів тренування моделі.....	69
4.2 Перевірка роботи моделі на мобільних пристроях.....	74
4.3 Можливість покращення результатів .....	81
4.4 Висновки до четвертого розділу.....	87
<b>ВИСНОВКИ .....</b>	<b>88</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>90</b>
<b>ДОДАТОК А</b> Лістинг програмного забезпечення.....	<b>97</b>
<b>ДОДАТОК Б</b> Презентаційні матеріали.....	<b>104</b>
<b>ДОДАТОК В</b> Копії статей .....	<b>116</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

МРЕ - метод розпізнавання емоцій

РЧ - реальний час

КЗ - комп'ютерний зір

ВО - вираз обличчя

БД - база даних

ПЗ - програмне забезпечення

ОС - операційна система

ЕС - емоційний стан

CV - computer vision (комп'ютерний зір)

НМ - нейронна мережа

ДО - детектор обличчя

## ВСТУП

Робота в напрямку розпізнавання емоцій людини за допомогою комп'ютерного зору є досить актуальною та перспективною для багатьох сфер застосування, таких як навчальний процес, медична сфера, рекламні послуги, психологічні науки тощо. Одним з важливих аспектів такої роботи є здатність визначати емоційний стан людини наживу, тобто з мінімальною затримкою від моменту отримання зображення до виведення результату. Для досягнення цього потрібно використовувати ефективні та швидкі методи та алгоритми обробки відеоданих, а також забезпечити високу точність та надійність розпізнавання.

Протягом останніх років значний прогрес у цій галузі було досягнуто завдяки застосуванню штучних нейронних мереж, які показують високу здатність до навчання та адаптації до різних умов. Зокрема, глибокі нейронні мережі дозволяють ефективно використовувати великі обсяги даних для тренування моделей, які можуть вирішувати складні завдання класифікації та регресії, пов'язані з розпізнаванням емоцій.

Актуальність даної роботи полягає у розробці та імплементуванню методу розпізнавання емоцій людини у реальному часі за допомогою машинного зору, який би використовував глибоку нейронну мережу та забезпечував високу швидкість та точність аналізу відеоданих.

Метою кваліфікаційної роботи магістра є розробка та дослідження методу розпізнавання емоцій людини у реальному часі за допомогою машинного зору на основі глибокої нейронної мережі.

Поставлена мета досягається розв'язанням таких основних задач:

- проведення аналізу існуючих методів та алгоритмів розпізнавання емоцій за допомогою комп'ютерного зору;
- вибору архітектури та гіперпараметрів глибокої згорткової нейронної мережі для розпізнавання емоцій;
- тренування та тестування моделі нейронної мережі на різних датасетах;

– розробка програмно-технічного засобу для розпізнавання емоцій у реальному часі.

Об'єктом дослідження у цій роботі є самий процес розпізнавання емоцій людини за допомогою комп'ютерного зору.

Предметом дослідження цієї роботи є застосування глибокої згорткової нейронної мережі для розпізнавання емоцій у реальному часі.

Наукова новизна отриманих результатів полягає в удосконаленні методу оптимізації нейронних мереж для задач розпізнавання емоцій, що дозволяє підвищити їхню продуктивність та забезпечити ефективне функціонування на мобільних пристроях з обмеженими ресурсами.

Практична цінність отриманих результатів полягає в розробці програмно-технічного засобу для розпізнавання емоцій у реальному часі, який може бути використаний для різних цілей, пов'язаних з аналізом емоційного стану людини.

За темою дипломної роботи була опублікована стаття [80-82]:

- у збірнику “Вчені записки” Таврійського національного університету імені В.І. Вернадського, Том 35 (74) №3 Частина 1. (Київ – 2024 – С. 77-85);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 15-21);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 37-44).

# 1 ОГЛЯД СУЧАСНИХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ЕМОЦІЙ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО ЗОРУ

## 1.1 Останні досягнення в галузі розпізнавання емоцій

Розпізнавання емоцій за допомогою комп'ютерного зору є актуальною та складною проблемою, яка має досить високий потенціал для покращення якості життя людей і розуміння поведінки. Емоції є важливою частиною людської комунікації, вони впливають на настрій, сприйняття, прийняття рішень та соціальні взаємодії. Розпізнавання емоцій може мати застосування в різних сферах, таких як освіта, медицина, безпека, реклама, розваги тощо. Однак, розпізнавання емоцій не є простим завданням, оскільки емоції є суб'єктивними, багатовимірними та динамічними явищами, які виражаються через різні модальності, такі як обличчя, голос, жести, постава, фізіологія тощо.

Однією з найбільш вивчених модальностей для розпізнавання емоцій є обличчя, оскільки воно містить багато інформації про емоційний стан людини. Обличчя може відображати як базові емоції (радість, смуток, гнів, страх, зневага, відраза, здивування), так і складніші емоційні стани (наприклад, сарказм, ревності, засмучення). Крім того, обличчя може показувати інтенсивність, тривалість та перехід між різними емоціями, що робить його багатограним та виразним засобом комунікації.

Для дослідження емоцій людини було створено багато графіків та робіт, за якими вчені досліджували емоції та намагалися краще зрозуміти поведінку людини через емоції, які вона виражає, що дозволило б спрогнозувати її поведінку.

Шкала емоцій Екмана [53] - це інструмент для вимірювання інтенсивності семи базових емоцій за допомогою фасетних мікровиразів обличчя. Ця шкала була розроблена американським психологом Полом Екманом та його колегами у 1970-х роках на основі їх досліджень крос-культурних виразів обличчя. Шкала емоцій Екмана складається з 60 одиниць, які описують різні характеристики обличчя, такі як зміщення брів, розширення носогубної складки, стиснення губ тощо. Кожна одиниця має певне значення від 0 до 5, яке вказує на силу виразу

обличчя. Сума значень усіх одиниць дає оцінку загальної інтенсивності емоції від 0 до 25.

Шкала емоційного стану Екмана призначена для того, щоб допомогти дослідникам та практикам визначати емоційний стан людини за її обличчям. Ця шкала може використовуватися для оцінки емоційних порушень, ефективності терапії, реакції на подразники, виявлення брехні, розпізнавання емоцій за відео та інших цілей. Шкала емоцій Екмана [53] є однією з найбільш авторитетних та надійних методик для вимірювання емоцій за обличчям. Шкала емоцій зображена на рисунку 1.1.

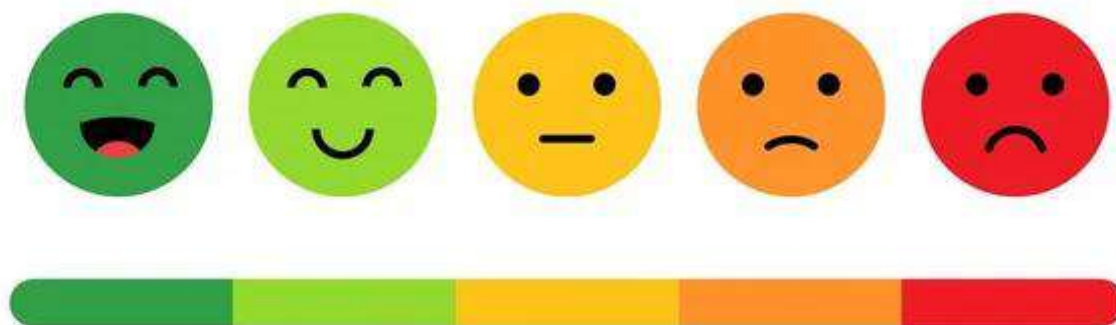


Рисунок 1.1 – Шкала емоцій Екмана [53]

Окремо можна виділити методику про діагностування емоційного інтелекту шляхом проведення тестування.

Методика “діагностики емоційного інтелекту” за Н. Холлом [54] призначена для оцінки здатності особистості розуміти емоції та керувати ними.

Вона включає 30 тверджень, які оцінюються за шкалою від -3 до +3, де “-3” означає повну незгоду та заперечення, а “+3” - повну згоду та схвалення з твердженням.

Таким чином, можна легко дослідити емоційний стан людини та зробити все для його покращення. Детальніше про п’ять шкал [54], які використовуються в цій методиці наведено у таблиці 1.1.

Таблиця 1.1 - П'ять шкал, які використовуються в методиці Холла

Шкала	Опис
Емоційна обізнаність	Здатність особистості спостерігати та розуміти свої власні емоції
Управління власними емоціями	Включає емоційну стресостійкість та емоційну негідність, що вказує на здатність особистості регулювати та контролювати свої емоційні реакції
Самотивація	Відображає довільне керування своїми емоціями та здатність особистості мотивувати себе до досягнення цілей
Емпатія та співчуття	Здатність відчувати, читати, розуміти емоції інших людей
Здатність розрізняти емоції інших людей	Включає вміння впливати на емоційний стан інших та розуміння емоцій за виразом обличчя

Кожне твердження в тесті відображає різні аспекти емоційного інтелекту, і респонденти оцінюють свою згоду з кожним твердженням. Результати можуть допомогти особистості краще розуміти та керувати своїми емоціями, а також емоціями інших.

Розпізнавання емоцій за виразом обличчя може бути сформульоване як класифікація, регресія або генерація. Завдання класифікації полягає в тому, щоб призначити обличчю одну або кілька попередньо визначених категорій емоцій. Завдання регресії полягає в тому, щоб оцінити неперервну величину емоційного стану, наприклад, за допомогою координат на двовимірному просторі валентності (приємності) та ароузалу (збудження). Завдання генерації полягає в тому, щоб створити синтетичне обличчя, яке відповідає заданій емоції або перетворити існуюче обличчя так, щоб воно виражало бажану емоцію.

Останні досягнення в галузі розпізнавання емоцій за виразом лиця пов'язані з застосуванням глибокого навчання, яке дозволяє автоматично вивчати репрезентації обличчя з великих наборів даних з різною роздільною здатністю, освітленням, позою, виразом тощо. Глибоке навчання також дозволяє інтегрувати різні модальності, такі як аудіо, текст, контекст, для збільшення правдивих результатів під час розпізнавання емоцій за обличчям. Крім того, глибоке навчання може використовуватися для побудови емоційно-сенситивних систем, які можуть адаптуватися до емоцій користувачів і здійснювати емоційну взаємодію.

Розпізнавання емоцій є ключовим компонентом у створенні інтуїтивно зрозумілих та емпатичних штучних систем, які можуть взаємодіяти з людьми на глибшому рівні. Останнім часом у цій галузі відбулися значні прориви, які відкривають нові можливості для покращення людсько-комп'ютерної взаємодії.

Одним з найбільш обнадійливих напрямків є використання механізмів уваги в поєднанні з моделями BERT та CNN для аналізу емоцій. Ці моделі дозволяють обробляти як аудіо, так і текстові дані, оптимізуючи CNN для виявлення емоційних коливань у мові, тоді як BERT обробляє текстові дані, використовуючи свої двонаправлені шари для глибокого семантичного розуміння. Сучасні дослідження показують, що інтеграція механізмів уваги може значно покращити точність розпізнавання емоцій. Використання CNN для аналізу аудіо даних дозволяє виявляти нюанси емоційних коливань у мові, тоді як BERT забезпечує глибоке розуміння контексту та емоційного забарвлення тексту. Об'єднання цих двох підходів через механізми уваги дозволяє створити більш точну та адаптивну систему розпізнавання емоцій.

Інший напрямок досліджень полягає у розробці безконтактних методів збору даних, які дають змогу збирати інформацію про емоційний стан без необхідності фізичного контакту з людиною. Це надзвичайно корисно у ситуаціях, де необхідно слідкувати за психологічним станом людини або взаємодіяти зі штучним інтелектом. Розробка безконтактних систем відкриває шлях до створення пристроїв, які можуть бути використані під час збору даних про емоційний стан без необхідності фізичного контакту та втручання. Це може включати в себе

використання датчиків, які аналізують голос, вираз обличчя, а також фізіологічні показники, такі як серцебиття та шкірну провідність. Такі системи можуть бути інтегровані у різноманітні пристрої, від смартфонів до носимих гаджетів, забезпечуючи новітні способи для відстеження емоційного стану людини.

Найновітніші дослідження, що об'єднують обчислювальні науки, психологію та сценічне мистецтво, розробили набір даних з анотованими рухами тіла, який має можливість покращити здатність штучного інтелекту розрізняти емоційний стан, виражені через мову тіла. Дослідження, що використовують анотовані рухи тіла, дозволяють штучному інтелекту краще розуміти емоції, виражені через мову тіла. Це може включати в себе аналіз постави, жестів, та рухів обличчя, які часто передають емоційну інформацію, що не може бути виявлена через слова або тон голосу.

Також, було проведено безліч конференцій та заходів щодо дослідження емоцій. Ось деякі з них:

1. ISRE 2024 Conference [55]. Міжнародне товариство дослідження емоцій (International Society for Research on Emotion, ISRE) провело свою конференцію в 2024 році з 17 по 20 липня в Університеті Королеви Белфаст у Північній Ірландії. Це важлива подія, яка збирає дослідників з різних дисциплін для обговорення нових ідей та результатів у сфері дослідження емоцій.

2. CERE - Consortium of European Research on Emotion [56]. Консорціум європейських досліджень емоцій (CERE) є неформальною мережею, яка сприяє міждисциплінарним дослідженням емоцій в Європі. Основною діяльністю CERE є організація дворічних конференцій.

3. Emotions 2023 Conference [57]. Університет Тілбурга був господарем конференції "Emotions 2023" з 4 по 6 жовтня. Конференція була присвячена передовим емпіричним дослідженням, теоретичним внескам та методологічним інноваціям у надзвичайно захоплюючій області дослідження емоцій.

Ці досягнення відкривають нові перспективи для створення більш емпатичних та інтуїтивних штучних систем, здатних розуміти та реагувати на людські емоції, що може кардинально змінити спосіб взаємодії машин з людьми.

## 1.2 Принципи та області застосування комп'ютерного зору для ідентифікації емоцій

Детальний опис ключових принципів та областей застосування цієї технології наведений у таблиці 1.2. Области застосування комп'ютерного зору наведені у таблиці 1.3.

Таблиця 1.2 - Детальний опис ключових принципів та областей застосування комп'ютерного зору

Етап	Опис
Збір даних	Системи комп'ютерного зору використовують великі набори даних з зображеннями людських облич, які містять різноманітні емоції для подальшого навчання різноманітних моделей
Обробка зображень	Перед тренуванням моделі, зображення обличчя піддаються процесам нормалізації, вирівнювання та масштабування для забезпечення консистентності даних
Виявлення особливостей обличчя	Застосовуються алгоритми для виявлення ключових точок обличчя, таких як куточки очей, носа, рота, які є важливими для ідентифікації емоцій
Екстракція ознак	Використовуються конволюційні нейронні мережі (CNN) для екстракції ознак з обличчя, які корелюють з емоційними станами
Класифікація емоцій	Навчені моделі аналізують екстраговані ознаки та класифікують емоції на основі визначених категорій, для кращого розуміння поведінки моделі, таких як радість, смуток, гнів, здивування тощо

Таблиця 1.3 – Области застосування комп'ютерного зору

Сфера	Застосування комп'ютерного зору	Мета
Медицина та психологія	Моніторинг емоційних станів пацієнтів	Допомога у діагностиці та лікуванні психологічних розладів
Освіта	Аналіз реакцій студентів	Адаптація навчального процесу та підвищення ефективності
Автомобільна промисловість	Виявлення ознак втоми або відволікання водія	Підвищення системи безпеки дорожнього руху
Ігрова та розважальні сфери	Створення інтерактивних та персоналізованих досвідів	Аналіз емоційного стану після розваг
Маркетинг та реклама	Аналіз емоційних реакцій споживачів	Краще уявлення поведінки клієнтів та адаптація маркетингових стратегій

Окрім того, дуже активно розвивається процес навчання роботів емоціям людини для кращого аналізу поведінки у сфері ресторанного бізнесу (рисунок 1.2) та правопорядку (рисунок 1.3). Саме у цих сферах зосереджена найбільша кількість взаємодій між людьми, що дає змогу якісно навчати та використовувати роботів при роботі з людьми.

Ключові моменти роботи робота офіціанта [59]:

- використання розпізнавання емоцій для підвищення якості обслуговування;
- можливість адаптувати свою поведінку та рекомендації в залежності від емоційного стану клієнтів;

– допомога у створенні більш персоналізованого досвіду для відвідувачів закладів.

Ключові моменти роботи робота поліцейського [58]:

– використання розпізнавання емоцій для оцінки потенційних загроз та намірів осіб;

– можливість визначати агресивні або підозрілі емоції, що можуть вказувати на небезпечну поведінку;

– допомога у взаємодії з громадянами для забезпечення більш ефективного та безпечного виконання поліцейських обов'язків.

Технології комп'ютерного зору та розпізнавання емоцій продовжують розвиватися, і їх застосування стає все більш розповсюдженим у різних галузях.

Вони відіграють ключову роль у покращенні взаємодії між людьми та комп'ютерами, а також у поліпшенні якості життя людей у повсякденному побуті та для забезпечення кращого існування людству.

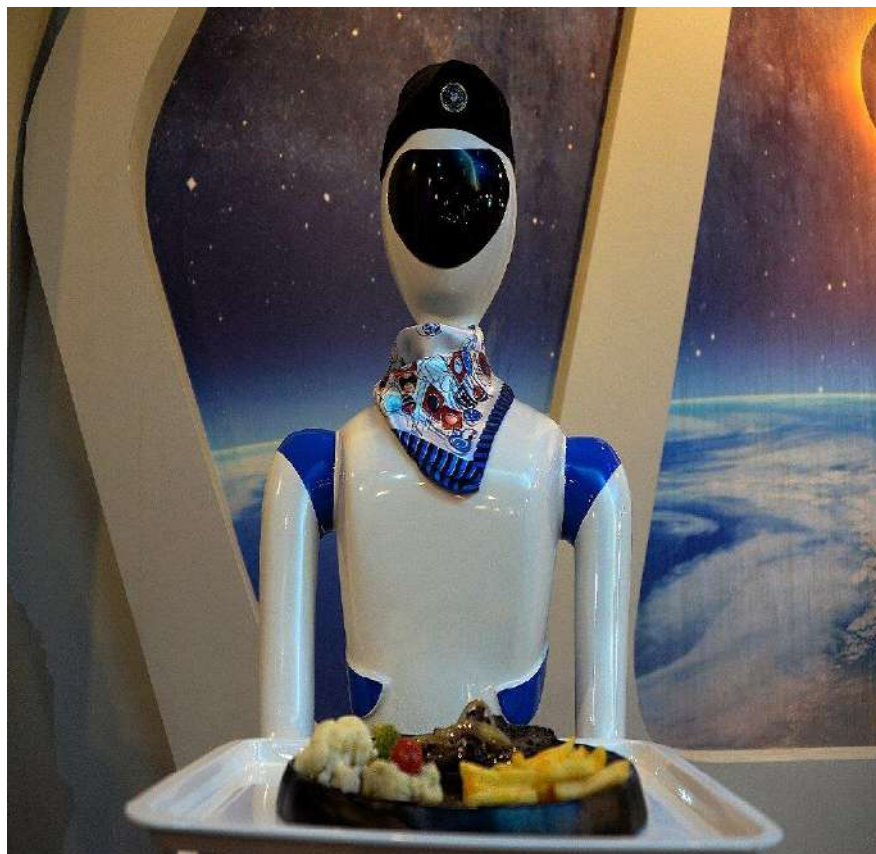


Рисунок 1.2 – Робот офіціант у ресторані [59]



Рисунок 1.3 – Робот поліцейський [58]

### 1.3 Аналіз типових помилок у системах розпізнавання емоцій

Системи розпізнавання емоцій, хоча й є досить передовими, все ще схильні до помилок, які можуть впливати на їхню точність та надійність. Детальний аналіз типових помилок, з якими можуть зіткнутися системи наведено у таблиці 1.4.

Таблиця 1.4 - Детальний аналіз типових помилок, з якими можуть зіткнутися системи

Проблема	Опис
Неправильне виявлення особливостей обличчя	Однією з основних проблем є неправильне виявлення ключових точок обличчя
Варіативність виразів обличчя	Людські емоції виражаються різними способами
Оклюдія обличчя	Ефективне розпізнавання емоцій може бути ускладнене при наявності перешкод між обличчям та камерою, щ

Кінець таблиці 1.4

Проблема	Опис
Фізіологічні сигнали	Деякі системи розпізнавання емоцій використовують фізіологічні сигнали, такі як серцебиття або шкірну провідність
Динамічні та складні середовища	Неперервне розпізнавання емоцій у динамічних або складних середовищах
Вплив освітлення та якості зображення	Недостатнє або надмірне освітлення, а також низька якість зображення можуть вплинути на точність системи розпізнавання емоцій
Контекстуальні помилки	Системи можуть не враховувати контекст, в якому виражаються емоції
Культурні та індивідуальні відмінності	Культурні норми та індивідуальні особливості впливають на те, як люди виражають емоції
Емоційна амбівалентність	Люди часто виражають більше однієї емоції одночасно, що може створити складнощі для систем

Формули для оцінки похибок під час розпізнавання емоцій людини вказані нижче.

Шум. Це вид похибки сенсора, який пов'язаний з випадковими варіаціями сигналу. Його можна охарактеризувати за допомогою стандартного відхилення  $\sigma$  та відношення сигнал-шум SNR, які обчислюються за формулою 1.1:

$$SNR = \mu/\sigma, \quad (1.1)$$

де  $\mu$  – це середнє значення сигналу;

$\sigma$  – це стандартне відхилення шуму;

SNR – це відношення сигнал-шум.

Співвідношення сигнал-шум (СШІ або SNR) визначається як співставлення потужності корисного сигналу до значення потужності шуму. Це співвідношення може бути виражене формулою 1.2:

$$SNR = \frac{PS}{PN}, \quad (1.2)$$

де PS – це середня потужність сигналу;

PN – це середня потужність шуму;

SNR – це відношення сигнал-шум.

Також SNR може бути обчислений як квадрат відношення амплітуд за формулою 1.3:

$$SNR = \left( \frac{A_{\text{сигнал}}}{A_{\text{шум}}} \right)^2, \quad (1.3)$$

де  $A_{\text{сигнал}}$  – це середнє квадратичне сигналу;

$A_{\text{шум}}$  – це середнє квадратичне шуму;

SNR – це відношення сигнал-шум.

Часто співвідношення сигнал-шум виражається у децибелах (дБ), і для цього використовується логарифмічна шкала у формулі 1.4:

$$SNR_{\text{дб}} = 10 * \log_{10} \left( \frac{P_{\text{сигнал}}}{P_{\text{шум}}} \right), \quad (1.4)$$

де  $P_{\text{сигнал}}$  – це потужність сигналу;

$P_{\text{шум}}$  – це потужність шуму;

SNR – це відношення сигнал-шум.

Також, можливо використовувати амплітуди, як зазначено у формулі 1.5:

$$SNR_{\text{дб}} = 20 * \log_{10} \left( \frac{A_{\text{сигнал}}}{A_{\text{шум}}} \right), \quad (1.5)$$

де  $A_{\text{сигнал}}$  – це середнє квадратичне значення амплітуди сигналу;

$A_{\text{шум}}$  – це середнє квадратичне значення амплітуди шуму;

SNR – це відношення сигнал-шум.

Помилка. Це різновид похибки сенсора, який відбувається, коли вимірюване значення відхиляється від реального значення за певним законом.

Найпростіше, помилку можна поділити на лінійну та нелінійну.

Лінійна помилка. Вимірюване значення залежить від реального значення, але з наявністю сталої похибки. Лінійна помилка сенсора визначається як різниця між вимірюваним значенням та реальним значенням, яка зберігається пропорційною до реального значення. Обчислюється за формулою 1.6:

$$E = (K * R) + B, \quad (1.6)$$

де  $E$  – лінійна помилка;

$K$  – це коефіцієнт пропорційності;

$R$  – реальне значення;

$B$  – це стала похибка.

Нелінійна помилка. Нелінійна помилка виникає, коли відносини між вимірюваним та реальним значенням не є лінійними. Наприклад, це може бути квадратична залежність або будь-яка інша нелінійна функція. Нелінійні помилки можуть бути виражені через більш складні функції, такі як формула 1.7:

$$E = (K * R^n) + B, \quad (1.7)$$

де  $E$  – лінійна помилка;

$K$  – це коефіцієнт пропорційності для нелінійної помилки;

$R^n$  – реальне значення зі ступенем нелінійності;

$B$  – це стала похибка нелінійної помилки.

Похибки алгоритмів комп'ютерного зору. Оцінка якості алгоритмів комп'ютерного зору включає аналіз таких ключових показників, як:

1. Precision – відсоток правильно ідентифікованих дійсно позитивних результатів з усіх позитивних прогнозів під час роботи, зроблених моделлю.

2. Accuracy – загальний відсоток дійсно правильних прогнозів, зроблених моделлю під час роботи, відносно всіх випробувань.

3. Recall – відсоток фактичних позитивних результатів, які модель змогла правильно ідентифікувати.

4. F1 score – гармонійне середнє між точністю та відгуком, що забезпечує збалансовану міру якості моделі.

Формули для обчислення цих показників включають метрики: Precision, Accuracy, Recall, F1-Score.

Precision визначається за формулою 1.8:

$$\text{Precision} = \frac{TP}{TP+FP}, \quad (1.8)$$

де TP – істино позитивні результати;

FP – фальшиво позитивні результати.

Accuracy визначається за формулою 1.9:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}, \quad (1.9)$$

де TP – істино позитивні результати;

TN – істино негативні результати;

FP – фальшиво позитивні результати;

FN – фальшиво негативні результати.

Recall визначається за формулою 1.10:

$$\text{Recall} = \frac{TP}{TP+FN}, \quad (1.10)$$

де TP – істино позитивні результати;

FN – фальшиво негативні результати.

F1-score визначається за формулою 1.11:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (1.11)$$

де Precision – точність;

Recall – повнота.

1.4 Постановка задачі щодо методу розпізнавання емоцій людини та вибір засобів для реалізації

У цій роботі планується розробити метод розпізнавання емоцій людини, який буде працювати на пристроях з обмеженою потужністю, що дозволить покращити витрату енергоресурсів та дозволить системам бути більш компактними.

Основними завданнями, які стоять перед методом розпізнавання емоцій людини, є: визначення джерела емоційної інформації (голос, обличчя, постава тощо); виділення релевантних ознак, які характеризують емоційний стан; класифікація емоцій за певними критеріями (наприклад, за базовими емоціями або за валентністю та активацією); оцінка точності та надійності отриманих результатів. Для вирішення цих завдань застосовуються різні методи обробки та аналізу даних, які базуються на статистичних, нейромережових, еволюційних або інших підходах. Кожен з цих методів має свої власні переваги та недоліки, а також досить специфічні умови застосування. Тому важливо вибирати оптимальний

метод для конкретної задачі розпізнавання емоцій, враховуючи якість, швидкість, вартість та інші фактори.

Потрібно розробити метод, який зменшує похибки в системі розпізнавання емоцій, використовуючи комп'ютерний зір, для реалізації в системах з великою кількістю камер спостереження.

Також необхідно підготувати набір даних з емоційними виразами облич для тренування нейронної мережі.

Після підготування датасету необхідно провести навчання та валідацію нейронної мережі для точного розпізнавання емоцій.

Потім, провести тестування у реальному часі для перевірки навчання моделі.

Необхідно інтегрувати систему розпізнавання емоцій людини до інших систем для коректної роботи.

Для виконання завдання у цій роботі необхідно визначитись з інструментами:

- вибір нейронної мережі. Для виконання цього завдання буде використано нейронну мережу Tensorflow [16], а саме Tensorflow Lite [60];
- вибір мови програмування та версії мови. Буде використано мову програмування Python; як основне середовище для роботи з нейромережею та обробки даних буде використано версію 3.9.12 [61].

Для коректної роботи методу буде проведено аналіз впливаючих факторів:

- врахування різноманітності умов експлуатації, включаючи різні типи камер та їх сенсорів, розміри зображення, та умови освітлення для коректної роботи системи в цілому;
- розробка алгоритмів, стійких до помилкових спрацьовувань, викликаних зовнішніми факторами, такими як опади, вітер, та світлові ефекти, для уникнення збоїв всередині системи;
- мінімізація ризиків та стратегії їх уникнення для забезпечення довготривалої роботи системи упродовж усього циклу інсування системи;
- планування стратегій для уникнення пікових навантажень та забезпечення стабільності системи під час інтеграції нового методу без створення затримок у роботі системи.

## 1.5 Висновки до першого розділу

У першому розділі було розглянуто сучасні дослідження розпізнавання емоцій людини, технології, сфери та можливості застосування комп'ютерного зору (КЗ) у повсякденному житті.

Було зроблено аналіз похибок, що виникають під час розпізнавання емоцій людини, причини їх виникнення та засоби запобігання виникнення помилок із можливістю мінімізації ризиків.

Було проаналізовано наукові публікації на схожу тематику для кращого розуміння; також проаналізовані варіанти реалізації вирішення схожих проблем.

Аналіз предметної галузі та проведених досліджень показав, що для роботи більшості систем розпізнавання емоцій людини необхідно мати досить потужні пристрої, і зважаючи на необхідність роботи системи на обмежених потужностях, найкращим вибором буде використання нейромережі Tensorflow.

## 2 ВИКОРИСТАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ ВИЗНАЧЕННЯ ЕМОЦІЙНОГО СТАНУ

### 2.1 Методики корекції помилок у системах визначення емоцій

Для виявлення усіх можливих похибок під час розпізнавання емоцій слід мати розуміння про узагальнену схему функціонування та класифікування. Схема зображена на рисунку 2.1.



Рисунок 2.1 – Загальна схема функціонування та класифікування

Для реалізації розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору необхідно забезпечити високий рівень точності та низьку затримку. Цього можна досягти, якщо врахувати різноманітні фактори, які впливають на швидкість та якість обробки даних. Ці фактори можна розділити на кілька головних категорій, які стосуються різних аспектів роботи системи комп'ютерного зору та її взаємодії з оточенням:

- помилки ідентифікації;
- помилки під час сприйняття даних;
- помилки під час роботи системи.

Детальніше розглянемо кожен з цих категорій.

Помилки ідентифікації. Помилки через недостатність даних для навчання нейромережі. Наприклад, система може не ідентифікувати певні емоції або вираження обличчя через малу кількість зображень серед навчальних наборів даних та датасетів. Також до цієї категорії належать помилки безпосередньо

обробки та кваліфікації даних. Наприклад, хибна класифікація та кластеризація даних під час тренування нейромережі може спричинити вказання неправильних емоцій під час розпізнання на зображеннях. Крім того, до помилок ідентифікації відносяться погане освітлення та низька роздільна здатність камери під час процесу розпізнання. Як приклад можна привести камеру, яка працює без освітлення, або камеру, яка не підходить для розпізнавання через свої низькі технічні характеристики.

Помилки під час сприйняття даних. До цих помилок можна віднести проблеми із сприйняттям інформації алгоритмами через невірне написання коду, або неправильне оптимізування програми. Як приклад, створений алгоритм може бути неприйнятним для даних із певним типом, а саме, розпізнавання емоцій може не працювати правильно на різних етнічних групах через малу кількість зображень для навчання та класифікування. Ще одним прикладом можна назвати алгоритм, який під час детекції невірно визначає положення та межі обличчя. До того ж, до даного виду помилок належать помилки, які відносяться до апаратної складової системи. Зокрема, це можуть бути помилки, що виникають через апаратні обмеження. Моделі можуть некоректно працювати через занадто малу кількість пам'яті та недостатній обсяг сховища, або, можуть виникати помилки, створені хибною роботою усього обладнання, наприклад, некоректної роботи процесора під час розпізнавання.

Помилки під час роботи системи. Це помилки, спричинені порушенням зв'язку та некоректною роботою системи в цілому.

Певні набори даних можуть зникнути під час транслювання (наприклад, під час виникнення перешкоди на шляху), або приходити із певними затримками під час транслювання. Усе це може призвести до створення помилок у системі ідентифікації емоцій.

Також, до цього ж виду категорій можна віднести й помилки синхронізування даних з різних камер.

Приклад усіх загальних помилок, методи для їх уникнення, мінімізації та їх класифікація наведені у таблиці 2.1.

Таблиця 2.1 – Методи запобігання помилок

Метод	Опис
Використання кодів для виправлення помилок (ЕСС)	ЕСС можуть бути застосовані у багатокласових сценаріях розпізнавання емоцій для підвищення точності
Аналіз впливу помилок автоматичного розпізнавання мови (ASR)	Дослідження показують, що помилки ASR можуть суттєво впливати на системи розпізнавання емоцій
Корекція ваги уваги	У системах, що використовують моделі уваги, можна вдосконалити корекцію ваги для словесного рівня
Використання вихідних кодів для виправлення помилок (ЕСОС)	ЕСОС можуть бути застосовані для багатоміткової класифікації емоцій
Моделювання помилок та їх виправлення	Розробка моделей, які можуть імітувати помилки, зазвичай зустрічаються в системах визначення емоцій
Адаптивні порогові значення	Встановлення адаптивних порогових значень для класифікації емоцій зменшить кількість помилок
Використання контекстної інформації	Інтеграція контекстної інформації може допомогти системі краще розуміти поточний емоційний стан та вправновідреагувати зважаючи на емоцію

Кінець таблиці 2.1

Метод	Опис
Багатозадачне навчання	Інтеграція багатозадачного навчання може покращити загальну точність системи
Аналіз чутливості та специфічності	Регулярний аналіз чутливості та специфічності системи має змогу ідентифікувати слабкі місця
Інтерактивне навчання	Залучення користувачів до процесу навчання системи може значно покращити точність розпізнавання емоцій
Глибоке навчання з підкріпленням	Використання методів глибокого навчання з підкріпленням може допомогти системі самостійно виявляти та коригувати помилки, що дозволить збільшити точність моделі та покращить результати

## 2.2 Огляд нейронних мереж, що застосовуються у дослідженні

Під час проведення дослідження було зроблено аналіз останніх наукових робіт які стосуються розпізнавання емоцій за допомогою різних нейронних мереж та способи їх класифікування. Наукові роботи були написані, щоб показати застосування моделей на основі CNN, таких як Tensorflow, AlexNet, VGGNet та бібліотек YOLO для різних галузей у реальному житті, таких як хімія, біологія, медицина, smart city та кібербезпека. Переглянуто фізичні системи, розпізнавання виразів облич.

Результати аналізу наукових робіт наведено в таблиці 2.2.

На основі аналізу вже існуючих рішень по розпізнаванню було вирішено створити власний датасет із 12 зображень для тестування і зробити порівняльний аналіз правильності розпізнавання емоцій людини.

Для проведення тестування були обрані такі нейромережі, як Tensorflow, Pytorch, YOLOv8, YOLOv9, YOLOv10. Датасет, який був підготовлений та зібраний для тестування якості моделей, зображений на рисунку 2.2. Він містить 12 чорно-білих картинок із зображенням людей, які показують різні емоції. Цей датасет дозволив порівняти показники якості серед вибраних моделей.

Таблиця 2.2 – Результати аналізу досліджень

Назва моделі	Рік створення	Створена	Особливості	Вплив/Застосування
AlexNet [63]	2012	Алекс Крижевський у співпраці з Іллею Суцкевером	Вісім шарів, включаючи п'ять згорткових та три повнозв'язних шари	Покращила результати у змаганні ImageNet у 2012, знизивши помилку top-5 до 15.3%
VGGNet [64]	2014	Карен Симонян та Ендрю Зіссерман з Оксфордського університету	Багатошарова архітектура з дрібними (3x3) згортковими фільтрами для покращення роботи моделі	Висока точність на ImageNet, майже 92.7% точності у top-5 тестуванні

Кінець таблиці 2.2

Назва моделі	Рік створення	Створена	Особливості	Вплив/Застосування
ResNet [65]	2015	Каймін Хе, Сянгю Чжан, Шаокін Рен, та Цзянь Сунь	Використовує “залишкові блоки” для тренування мереж	Виграла ILSVRC 2015, помилка тренування лише 3.57%
Inception (GoogLeNet)	2014	Команда дослідників Google	22 шари, використовує модулі Inception	Знизила помилку top-5 у ImageNet до 6.7%
EfficientNet [66]	2019	Команда дослідників Google AI	Compound scaling для масштабування ширини, глибини	Висока точність у різних задачах комп'ютерного зору
MobileNet [67]	2017	Дослідники з Google	Оптимізована для мобільних пристроїв	Класифікація зображень та об'єктного виявлення
SqueezeNet [68]	2016	Дослідники з DeepScale,	Використовує “fire modules”	Класифікація зображень з ImageNet
DenseNet [69]	2017	Gao Huang,	Прямі з'єднання	Покращує потік інформації



Рисунок 2.2 – Датасет для перевірки якості моделей [70]

Оскільки, датасет містить лише зображення однієї людини та лише одну емоцію на фото, то тестування моделей полягало у правильності виокремлення облич та правильності встановлення емоції відповідно до зображення. Уся інформація та результати були класифіковані та наведені у таблиці 2.3.

Таблиця 2.3 – Результати досліджень датасету

Критерії	Tensorflow [16]	Tensorflow Lite [60]	YOLOv8 [71]	YOLOv9 [72]	YOLOv10 [73]
Лице та емоція	11	10	11	11	11
Розпізнала обличчя але не емоцію	1	1	0	0	1

Кінець таблиці 2.3

Критерії	Tensorflow [16]	Tensorflow Lite [60]	YOLOv8 [71]	YOLOv9 [72]	YOLOv10 [73]
Випадків, мережа не змогла розпізнати обличчя	0	1	1	1	0
% правильних відповідей	91.6	83.3	91.6	91.6	91.6
Переваги	Гнучкість та широкий спектр застосувань у сфері розпізнавання	Оптимізована для мобільних пристроїв	Висока точність розпізнавання	Висока точність	Новітні засоби розпізнавання

### 2.3 Підготовка набору даних та датасету для тренування мережі

Підготовка набору даних для тренування нейронної мережі з використанням TensorFlow та TensorFlow Lite вимагає ретельного підходу, який забезпечить високу якість та релевантність даних для ефективного навчання моделі. Основним завданням є створення датасету, який точно відображає різноманітність емоційних станів, які модель повинна розпізнавати.

На початку процесу збору даних необхідно зібрати велику кількість зображень облич з різними емоційними виразами. Ці зображення повинні включати різні вікові групи людей, етнічні та расові групи, стать, а також повинні бути зроблені в різних умовах освітлення навколишнього середовища та з різних ракурсів. Це забезпечить, що модель буде здатна розпізнавати емоції в широкому спектрі ситуацій.

Кожне зібране зображення потребує анотації, яка вказує на присутню емоцію. Анотація може бути виконана вручну або з використанням автоматизованих інструментів, але важливо, щоб кожна анотація була перевірена людиною для забезпечення точності.

Після збору та анотації даних, вони повинні бути підготовлені до тренування моделі. Це включає нормалізацію зображень до однакового розміру та масштабування кольорових каналів до формату, який приймається моделлю. Наприклад, якщо модель очікує дані у форматі float32 з діапазоном значень від 0 до 1, всі зображення повинні бути відповідно нормалізовані.

Для забезпечення узагальнюючої здатності моделі, датасет розділяється на тренувальний, валідаційний та тестовий набори. Тренувальний набір використовується для навчання моделі, валідаційний - для налаштування гіперпараметрів та уникнення перенавчання, а тестовий - для оцінки кінцевої точності моделі.

Аугментація даних є ще одним важливим кроком, який допомагає моделі стати більш робастною. Це може включати зміни в освітленні, додавання шуму, зміни в масштабі та повороти зображень. Такі техніки дозволяють моделі навчитися розпізнавати емоції навіть коли вхідні дані не ідеальні.

Балансування класів є важливим, особливо якщо деякі емоції представлені менше, ніж інші. Це може бути досягнуто за допомогою технік ресемплінгу, які забезпечують рівномірне представлення кожної емоції в тренувальному наборі даних.

Після підготовки даних, вони використовуються для тренування моделі за допомогою TensorFlow. Важливо вибрати відповідну архітектуру нейронної

мережі, яка підходить для задачі розпізнавання емоцій. TensorFlow Lite дозволяє оптимізувати треновану модель для швидкої роботи на мобільних пристроях, зменшуючи її розмір та затримку без значної втрати точності.

Оцінка та тестування моделі на перевіркового та тестовому наборах даних дозволяє перевірити її точність та здатність до узагальнення. Це включає аналіз показників, таких як точність, втрата, чутливість та специфічність, щоб забезпечити, що модель працює належним чином.

В цілому, підготовка набору даних для тренування нейронної мережі є доволі складним процесом, який вимагає уваги до найменших деталей на кожному етапі, від збору даних до оцінки моделі. Використання TensorFlow та TensorFlow Lite дозволяє ефективно виконувати цей процес, забезпечуючи високу якість та точність моделі для розпізнавання емоцій.

Поділ зображень на тренувальні, валідаційні та тестові набори є важливим етапом у підготовці даних для тренування нейронної мережі. Цей процес забезпечує, що модель буде здатна не тільки навчитися розпізнавати емоції на даних, на яких вона тренувалася, але й ефективно узагальнювати свої знання на нових, невидимих даних.

Тренувальний набір даних використовується для безпосереднього навчання моделі. Це основний набір, на якому модель “вчиться” розпізнавати емоції, адаптуючись до властивостей та особливостей даних. Тренувальний набір повинен бути достатньо великим та різноманітним, щоб модель могла навчитися розпізнавати широкий спектр емоційних виразів.

Валідаційний набір даних використовується для тонкої настройки та валідації моделі під час процесу тренування. Він допомагає визначити, коли модель починає перенавчатися, тобто коли її продуктивність на тренувальному наборі продовжує покращуватися, але погіршується на валідаційному наборі. Валідаційний набір також використовується для вибору найкращої моделі та гіперпараметрів.

Тестовий набір даних є кінцевим іспитом для моделі. Він використовується після завершення всіх етапів тренування та валідації, щоб дати оцінку, наскільки

дobre модель може адаптовувати своє навчання на повністю нових даних. Тестовий набір повинен бути ізольованим від тренувального та валідаційного наборів і не використовуватися в процесі тренування.

Правильний поділ даних на ці три набори допомагає забезпечити, що оцінка моделі буде об'єктивною та відображатиме її реальну здатність до розпізнавання емоцій у різних умовах при збереженні початкових налаштувань та конфігурацій без необхідності коригування самої моделі.

Датасети можуть бути як відкритими, тобто створені користувачами, так і закритими, тобто створені вченими для спеціального навчання моделей. Переваги та недоліки кожного з датасетів зображені у таблицях 2.4 - 2.5.

Таблиця 2.4 – Переваги та недоліки використання відкритих датасетів

Переваги використання відкритих датасетів	Недоліки використання відкритих датасетів
Доступність. Відкриті датасети легко доступні для дослідників та розробників	Якість даних. Іноді відкриті датасети можуть містити помилки або бути неповними
Економія часу. Використання готових датасетів значно скорочує час	Репрезентативність. Датасети можуть не відображати різноманітність реального світу
Спільнота. Велика спільнота користувачів та дослідників, які можуть надати підтримку та поради	Актуальність. Деякі датасети можуть бути застарілими, і новітні дані можуть бути недоступні
Бенчмаркинг. Відкриті датасети часто використовуються для порівняння продуктивності різних моделей та алгоритмів	Надмірне використання. Популярні датасети можуть бути “занадто знайомими” моделям, що призводить до перенавчання

Кінець таблиці 2.4

Переваги використання відкритих датасетів	Недоліки використання відкритих датасетів
Стандартизація. Багато відкритих датасетів мають стандартизовані формати, що полегшує їх інтеграцію та використання у різних проектах	Ліцензування. Не всі відкриті датасети можуть використовуватися для комерційних цілей

Таблиця 2.5 – Переваги та недоліки використання спеціалізованих датасетів

Переваги використання спеціалізованих датасетів	Недоліки використання спеціалізованих датасетів
Висока релевантність. Датасети можуть бути спеціально розроблені для відповідності конкретним вимогам проекту, що забезпечує високу релевантність даних	Обмежена різноманітність. Спеціалізовані датасети можуть не відображати всю різноманітність емоційних виразів, які модель може зустріти в реальному світі
Контрольована якість. Можливість контролювати процес збору та анотації даних забезпечує високу якість датасету, що надає кращу можливість взаємодії всередині набору даних	Висока вартість. Створення спеціалізованих датасетів може бути дорогим та часозатратним процесом
Точність анотацій. Анотації, виконані експертами, можуть бути більш точними, що покращує навчання та валідацію моделі для зручності користування набору даних	Складність збору даних. Збір спеціалізованих даних може вимагати специфічних умов або обладнання

Кінець таблиці 2.5

Переваги використання спеціалізованих датасетів	Недоліки використання спеціалізованих датасетів
Уникнення упередженості. Спеціалізовані датасети можуть бути спроектовані таким чином, щоб уникнути упередженості, присутньої в деяких відкритих датасетах	Обмеження у використанні. Спеціалізовані датасети можуть мати обмеження на використання, що може ускладнити їх подальше розповсюдження або використання
Адаптація до специфічних умов. Датасети можуть бути адаптовані до специфічних умов використання, наприклад, освітлення або ракурсів	Потреба в експертах. Для анотації та перевірки даних може знадобитися залучення кваліфікованих експертів

Наявність засобів обробки та класифікації зображень є надзвичайно важливою для процесу навчання моделей машинного навчання, особливо в областях, де використовуються візуальні дані, такі як комп'ютерний зір. Ці інструменти відіграють критичну роль у підготовці даних, які є фундаментом для будь-якої ефективної моделі.

Перш за все, обробка зображень дозволяє стандартизувати вхідні дані, що сприяє більш ефективному навчанню. Наприклад, зміна розміру зображень до єдиного формату, корекція освітленості та контрасту, а також нормалізація кольорів забезпечують, що модель не витрачає ресурси на вивчення нерелевантних варіацій у даних.

Класифікація зображень, з іншого боку, допомагає в структуруванні датасету. Вона визначає категорії або мітки, які модель повинна вивчити, і забезпечує чітке розмежування між різними класами. Це особливо важливо в задачах розпізнавання образів, де модель повинна визначити, до якої категорії належить кожне зображення.

Крім того, засоби обробки та класифікації допомагають виявити та виправити помилки в даних, такі як неправильні мітки або пошкоджені зображення. Це забезпечує високу якість тренувального датасету, що є ключовим для досягнення високої точності моделі.

Нарешті, ці інструменти дозволяють виконувати аугментацію даних, тобто штучне збільшення різноманітності тренувального набору шляхом внесення змін до зображень, таких як повороти, масштабування, або додавання шуму. Аугментація даних може значно покращити здатність моделі узагальнювати та працювати ефективно на нових даних, які вона не бачила під час тренування.

Таким чином, засоби обробки та класифікації зображень є незамінними для створення потужних та точних моделей машинного навчання, які можуть ефективно працювати з візуальними даними. Приклад деяких засобів для обробки зображень наведено у таблиці 2.6.

Таблиця 2.6 – Засоби для обробки даних

Засіб / Бібліотека	Опис	Переваги	Недоліки
OpenCV [74]	Бібліотека з відкритим кодом для комп'ютерного зору	Широкий вибір інструментів для обробки зображень	Може бути складною для новачків
Pillow (PIL Fork) [75]	Бібліотека для обробки зображень	Проста у використанні	Обмежені можливості порівняно з OpenCV
scikit-image [76]	Бібліотека для обробки зображень	Інтеграція з іншими бібліотеками	Менш підходить для роботи в реальному часі

Кінець таблиці 2.6

Засіб / Бібліотека	Опис	Переваги	Недоліки
TensorFlow/Keras	Бібліотеки для глибокого навчання, які можуть використовуватися для класифікації зображень	Висока гнучкість; підтримка складних мережевих архітектур	Вимагає більше ресурсів для тренування
PyTorch [77]	Бібліотека для машинного навчання, яка підтримує динамічні обчислювальні графи	Гнучкість у проектуванні моделей; широка підтримка дослідницької спільноти	Може бути складною для новачків
Fastai	Бібліотека, яка побудована на базі PyTorch, спрощує тренування моделей глибокого навчання	Висока продуктивність; простота використання	Обмежена гнучкість порівняно з чистим PyTorch
Albumentations	Бібліотека для аугментації зображень	Швидка та ефективна аугментація	Спеціалізована на аугментації, не для загальної обробки

Після аналізу усіх засобів для обробки даних нам необхідно вибрати платформу з датасетами для навчання нашої моделі. Після порівняння усіх плюсів та мінусів як закритих, так і відкритих датасетів, було вирішено застосовувати відкриті датасети. Найбільшою платформою з відкритими датасетами є Roboflow.

Roboflow є інноваційною платформою, яка надає значні можливості для роботи з датасетами в області комп'ютерного зору. Вона пропонує широкий спектр відкритих датасетів, які можуть бути використані для тренування та вдосконалення моделей машинного навчання. Ці датасети охоплюють різноманітні сценарії використання, від простого розпізнавання об'єктів до складних задач, таких як семантична сегментація та класифікація.

Roboflow не лише надає доступ до готових датасетів, але й пропонує інструменти для їх аугментації та оптимізації. Користувачі можуть легко зменшувати розміри датасетів, збільшувати їх різноманітність за допомогою аугментації, що включає зміни освітлення, повороти, масштабування та інші трансформації. Такі функції дозволяють підготувати датасети, які краще відповідають специфічним потребам проекту. Переваги та недоліки платформи наведено у таблиці 2.7.

Таблиця 2.7 – Переваги та недоліки використання Roboflow

Переваги	Недоліки
Велика колекція датасетів. Roboflow має одну з найбільших колекцій відкритих датасетів, що постійно оновлюється	Залежність від сервісу. Використання датасетів з Roboflow може створити залежність від платформи та її інструментів
Готовість до використання. Датасети на Roboflow доступні у багатьох форматах	Обмеження на використання. Не всі датасети можуть бути вільно використані для комерційних цілей

Кінець таблиці 2.7

Переваги	Недоліки
Інструменти аугментації. Платформа пропонує потужні інструменти для аугментації, які допомагають збільшити різноманітність даних та покращити узагальнюючу здатність моделей	Кількість моделей. Недостатня кількість моделей для навчання мереж
Спільнота та підтримка. Roboflow має активну спільноту користувачів та розробників, які можуть ділитися знаннями та надавати підтримку	
Простота використання. Інтуїтивно зрозумілі інструменти роблять Roboflow доступним для користувачів різного рівня досвіду	

Після визначення з платформою для пошуку датасетів для тренування нашої нейромережі, потрібно власне визначитись з базою даних. Серед багатьох наборів зображень було вибрано саме Fer2013, оскільки цей датасет має найбільшу кількість зображення, що є плюсом для тренування нашої моделі.

Датасет FER2013 (Facial Emotion Recognition 2013) є одним з найбільш відомих та широко використовуваних датасетів у сфері розпізнавання емоцій за допомогою комп'ютерного зору. Він був створений для конкурсу Kaggle, де учасники зі всього світу могли позмагатися у сфері навчання моделей. Цей датасет містить понад 35,000 зображень обличч людей, які виражають сім основних емоцій: гнів, огида, страх, щастя, смуток, здивування та нейтральний стан. Це дозволяє навчити модель на досить високому рівні та надасть можливість покращувати

результати моделі шляхом додавання нових зображень та нових емоцій вцілому. Особливості цього датасету наведені у таблиці 2.8.

Таблиця 2.8 – Особливості датасету Fer2013

Особливість	Деталі
Кількість зображень	Понад 35,000 зображень у відтинках сірого (grayscale), розмір 48x48 пікселів
Категоризація емоцій	Кожне зображення має мітку однієї з семи емоцій
Розподіл даних	Поділений на тренувальний, валідаційний та тестовий набори
Різноманітність виразів	Широкий спектр виразів облич, а саме 7 емоцій
Доступність	FER2013 доступний через платформу Kaggle
Стандартизація	FER2013 є стандартизованим датасетом
Велика кількість даних	Дозволяє тренувати глибокі нейронні мережі
Різноманітність емоцій	Сім категорій емоцій
Підтримка спільноти	Велика спільнота користувачів, які щодня завантажують безліч нових моделей та покращують вже наявні

## 2.4 Конфігурація нейронної мережі для навчання

Раніше, у цьому документі, було розглянуто та порівняно деякі моделі нейромереж, які підходять для розпізнавання емоцій. Так, як задача дослідження полягає у розробленні методу який буде працювати на пристроях з обмеженою потужністю, то найкращим варіантом буде Tensorflow Lite модель. Для того, щоб отримати Lite версію моделі, спочатку необхідно натренувати модель Tensorflow, а потім конвертувати модель у .tflite(Tensoflow Lite) формат. Загальний процес навчання та перетворення зображень на рисунку 2.3.

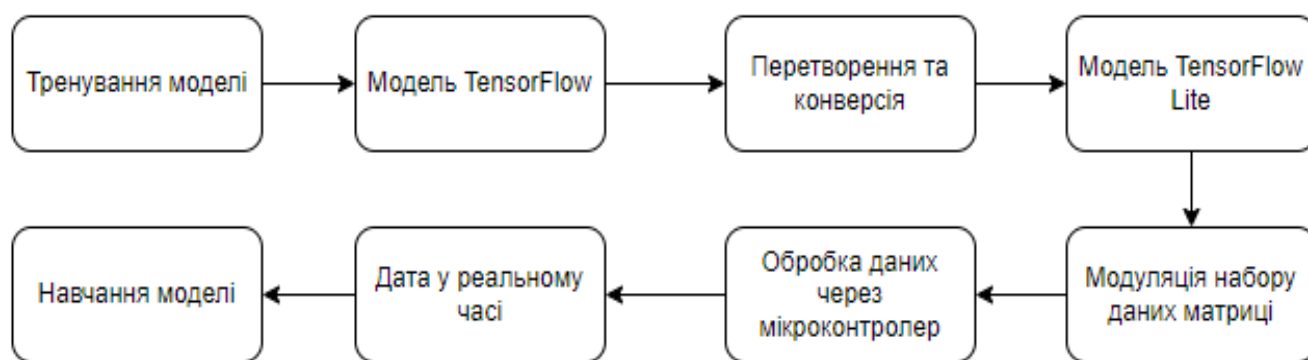


Рисунок 2.3 – Процес створення Tensorflow Lite моделі

Для зручності було використано API для Tensorflow, а саме Keras.

Keras - це високорівневе API для TensorFlow, яке спрощує та прискорює процес машинного навчання, зосереджуючись на сучасних методах глибокого навчання. Keras дозволяє керувати кожним кроком робочого процесу машинного навчання, від підготовки даних до налаштування гіперпараметрів та розгортання.

Основні елементи Keras - це шари (layers) та моделі (models). Шари в Keras - це прості перетворення вхідних/вихідних даних, а модель - це направлений ациклічний граф шарів. Це означає, що ви можете побудувати складні архітектури нейромереж, поєднуючи шари в різні конфігурації. Це значно спрощує роботу з тренуванням різних моделей та пришвидшує взаємодію з користувачем.

Порівняно з TensorFlow, Keras надає більш простий та зрозумілий інтерфейс, який дозволяє швидко розпочати роботу з глибоким навчанням, не заглиблюючись

у складності низькорівневих операцій TensorFlow. Основні переваги Keras полягають у:

1. Інтуїтивність. Інтуїтивно зрозумілі абстракції для шарів та моделей, що дозволяє легко створювати та модифікувати нейронні мережі;
2. Модульність. Модульність дозволяє легко та швидко експериментувати з різними архітектурами;
3. Гнучкість. Гнучкість у виборі високорівневих або низькорівневих API, залежно від потреб проекту.

TensorFlow Lite є легковаговою версією TensorFlow, оптимізованою для мобільних та вбудованих пристроїв з обмеженою продуктивністю. Вона дозволяє використовувати моделі машинного навчання, які були навчені та розроблені з використанням TensorFlow, але в більш компактному та ефективному форматі. Особливості цієї моделі наведені у таблиці 2.9.

Таблиця 2.9 – Особливості моделі Tensorflow Lite

Особливість	Опис
Ефективність	Моделі TensorFlow Lite оптимізовані для швидкої роботи на пристроях з обмеженою продуктивністю, таких як мобільні телефони та IoT пристрої
Компактність	Моделі TensorFlow Lite мають менший розмір порівняно з оригінальними TensorFlow моделями
Гнучкість	TensorFlow Lite підтримує широкий спектр платформ та архітектур, включаючи Android, iOS та мікроконтролери для продуктивної роботи

Кінець таблиці 2.9

Особливість	Опис
Спільнота та підтримка	TensorFlow Lite має сильну підтримку від спільноти та розробників, що забезпечує регулярні оновлення та покращення
Обмеження функціональності	Деякі складні операції та архітектури, які доступні в TensorFlow, можуть не підтримуватися в TensorFlow Lite
Точність	Квантовані моделі можуть мати трохи нижчу точність порівняно з повнорозмірними моделями
Швидкість інференції	Модель оптимізована для швидкої роботи на мобільних пристроях

Отже, після того як було визначено усі аспекти щодо вибору моделі, потрібно розглянути налаштування моделі для максимальної продуктивності після навчання і під час розпізнавання емоцій.

Процес тренування моделей машинного навчання часто вимагає ітераційного підходу, який спрямований на поліпшення ключових показників ефективності, таких як акуратність, точність та відгук. Зазвичай, якщо набір даних є об'ємним та коректно анотованим, можливо досягти задовільних результатів без необхідності змінювати модель або її налаштування. Однак, перед будь-якими модифікаціями, рекомендується спочатку виконати тренування з використанням стандартних налаштувань. Це дозволяє встановити базовий рівень продуктивності та визначити можливості для подальшого вдосконалення. Далі розглянемо декілька ключових аспектів, які слід враховувати при налаштуванні тренування моделей, зокрема:

- визначення кількості епох (циклів) тренування;

- налаштування специфікацій вхідних зображень;
- регулювання розміру пакету даних для обробки;
- вказання гіперпараметрів необхідних для тренування.

Тривалість тренувального процесу, виражена в епохах, має безпосередній вплив на якість моделі. Збільшення числа епох може покращити деталізацію навчання, хоча існує ризик, що модель занадто “запам’ятає” тренувальні дані, що відомо як перенавчання. Для моделей, створених за допомогою TensorFlow з використанням Keras API, початкова рекомендація може бути встановленням приблизно в 100 епох. Якщо аналіз показує відсутність перенавчання, кількість епох можна збільшити до 250 або навіть 400. TensorFlow з Keras також пропонує інструменти для моніторингу перенавчання, такі як параметр ‘patience’ у функції зворотного виклику EarlyStopping, що дозволяє припинити тренування, якщо модель не покращується протягом визначеного числа епох.

Щодо розміру зображень, оптимальні результати часто досягаються, коли розміри зображень для розпізнавання відповідають або близькі до розмірів зображень, використаних під час тренування. Якщо зображення містять багато дрібних деталей, може бути корисним використання більших зображень для тренування. У TensorFlow, за замовчуванням, моделі можуть бути натреновані на зображеннях розміром 640 пікселів, але параметри тренування дозволяють легко змінювати розміри зображень за допомогою опції ‘-img’, наприклад, ‘-img 1280’ (розмір картинки у 1200 пікселів).

Розмір пакету, або ‘batch size’, визначає кількість зразків, які обробляються одночасно під час одного проходу алгоритму. Це впливає на швидкість тренування, оскільки більші пакети дозволяють моделі обробляти більше даних за одну епоху. Проте, занадто великі пакети можуть сповільнити оновлення ваг моделі. Більші пакети також можуть згладжувати градієнти, що сприяє швидшому досягненню оптимального рішення, але великі пакети можуть зменшити здатність моделі до узагальнення. Крім того, розмір пакету пов’язаний з обчислювальними ресурсами, оскільки більші пакети вимагають більше пам’яті для обробки.

Нарешті, гіперпараметри є структурними налаштуваннями, які встановлюються перед початком тренування і не змінюються протягом процесу. Вони включають такі параметри, як швидкість навчання, кількість шарів, типи функцій активації та оптимізаторів. Вибір правильних гіперпараметрів є ключовим для успішного тренування моделі та для запобігання необхідності повторного навчання. Основні параметри та їх опис представлено у таблиці 2.10.

Таблиця 2.10 – Основні гіперпараметри

Параметр	Опис	Вплив
Швидкість навчання (Learning Rate)	Визначає, наскільки швидко модель адаптується до проблеми	Занадто висока швидкість може призвести до того, що модель “перестрибне” через оптимальні рішення, а занадто низька – до повільного навчання або зупинки на локальних мінімумах
Кількість епох (Number of Epochs)	Кількість повних проходів через тренувальний датасет	Більша кількість епох може покращити точність моделі, але також збільшує ризик перенавчання
Розмір пакету (Batch Size)	Кількість тренувальних прикладів, які обробляються за один раз	Великі пакети можуть прискорити тренування, але вимагають більше пам’яті та можуть зменшити точність

Кінець таблиці 2.10

Параметр	Опис	Вплив
Регуляризація (Regularization)	Техніки, такі як L1 або L2 регуляризація, додають штраф до функції втрат	Запобігає перенавчанню, змушуючи модель зберігати ваги невеликими
Функція активації (Activation Function)	Визначає вихід шару нейронів	Різні функції активації можуть впливати на здатність моделі до навчання та її поведінку
Оптимізатор (Optimizer)	Алгоритм, який використовується для оновлення ваг моделі під час тренування	Різні оптимізатори можуть впливати на швидкість та стабільність навчання

## 2.5 Висновки до другого розділу

Штучні нейронні мережі (ШНМ) швидко набувають популярності як ефективний засіб для коригування помилок у системах комп'ютерного зору (СКЗ). Їхня здібність до адаптації на основі обширних даних та ідентифікація складних взаємозв'язків робить їх оптимальним рішенням для таких завдань. Основні переваги застосування ШНМ для корекції помилок у СКЗ порівняно з класичними методами обробки зображень включають високий рівень точності, адаптивність та розширені можливості для автоматизації. Нижче представлено декілька прикладів використання ШНМ у цій сфері:

- реставрація зображень. ШНМ можуть бути застосовані для реконструкції ушкоджених чи розмитих зображень, відновлюючи втрачені або пошкоджені пікселі;

- класифікація зображень. ШНМ ефективні для розподілу зображень по категоріях, навіть у присутності шуму чи перекриття;
- детекція об'єктів. ШНМ можуть виявляти об'єкти на зображеннях, незалежно від часткового приховування або складних умов освітлення;
- сегментація зображень. ШНМ здатні розділяти зображення на окремі сегменти, навіть коли границі між ними розмиті або нерівномірні;
- трекінг об'єктів. ШНМ можуть бути використані для слідкування за об'єктами у відеопотоках, що є корисним для контролю трафіку, систем безпеки та автономного водіння.

ШНМ є потужним інструментом, який може істотно підвищити ефективність СКЗ. З огляду на їхній постійний розвиток, очікується, що вони відіграватимуть ще більш значущу роль у виправленні помилок у СКЗ у майбутньому.

### 3 МЕТОДИКА ТА АЛГОРИТМ ВИЗНАЧЕННЯ ЕМОЦІЙ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ НЕЙРОННОЇ МЕРЕЖІ

#### 3.1 Процес тренування мережі для ідентифікації емоцій

Процес тренування нейронної мережі для ідентифікації емоцій включає кілька ключових етапів:

1. Підготовка даних. Навчання нейронної мережі починається з підготовки датасету, який складається з зображень облич людей та відповідних міток емоцій. Зображення можуть бути нормалізовані та аугментовані для покращення різноманітності та якості даних.

2. Вибір архітектури моделі. Вибір правильної архітектури моделі є важливим кроком. Для задач розпізнавання емоцій часто використовуються згорткові нейронні мережі (CNN), які підходять для обробки та класифікації зображень.

3. Навчання моделі. Під час цього етапу, модель “вчиться” розпізнавати емоції, коригуючи свої параметри на основі помилок, які вона робить. Це включає в себе прямий прохід (forward pass), де модель робить прогнози на основі поточних параметрів, та зворотний прохід (backward pass), де градієнти помилок розповсюджуються назад через мережу, щоб оновити параметри.

4. Валідація та тестування. Після тренування моделі, вона перевіряється на валідаційному наборі даних, який не використовувався під час тренування. Це дозволяє оцінити, наскільки добре модель узагальнює своє навчання на нових даних. Після цього, модель тестується на тестовому наборі даних, щоб оцінити її кінцеву продуктивність.

5. Оптимізація та налаштування. На основі результатів валідації та тестування, модель та її параметри можуть бути додатково оптимізовані та налаштовані. Це може включати зміну гіперпараметрів, таких як швидкість навчання, розмір пакету, кількість епох, а також архітектуру моделі.

6. Розгортання. Після того, як модель була навчена, валідована, протестована та оптимізована, вона може бути розгорнута для реального

використання. Це може включати конвертацію моделі до формату, який може бути використаний на цільовій платформі, наприклад, TensorFlow Lite для мобільних пристроїв.

Цей процес є ітеративним, і може бути повторений кілька разів, доки не буде досягнуто задовільної продуктивності. Кожен етап вимагає ретельного планування та виконання, але результатом є потужна модель, здатна розпізнавати емоції з високою точністю та надійністю.

Процес навчання штучної нейронної мережі - це серія кроків, які використовуються для оптимізації ваг мережі. На початку всі ваги ініціалізуються випадковим чином. Потім вхідні дані проходять через мережу в процесі, який називається прямим поширенням, щоб отримати прогнозоване вихідне значення.

Після цього обчислюється втрата, яка є мірою того, наскільки прогнозоване вихідне значення відрізняється від реального вихідного значення. Ця втрата потім використовується в процесі, який називається зворотним поширенням помилки, для оновлення ваг мережі таким чином, щоб зменшити втрату. Процес навчання моделі зображений на рисунку 3.1.

Існують три основні типи навчання, які використовуються в штучних нейронних мережах: навчання з наглядом, навчання без наглядача та підсилене навчання.

Навчання з наглядом – це найпоширеніший тип навчання, де мережі навчаються на основі вхідних даних та відповідних вихідних даних. Це дуже ефективно для задач прогнозування та класифікації. Однак, цей метод вимагає великої кількості маркованих даних та може бути витратним у часі та ресурсах.

Навчання без наглядача – це тип навчання, де мережі навчаються виявляти шаблони та структури в даних без вказівок або вихідних даних. Це корисно для задач кластеризації та зниження розмірності. Проте, цей метод може бути складним для інтерпретації та вимагає великої кількості даних.

Підсилене навчання – це тип навчання, де мережі навчаються за допомогою випробувань та помилок, отримуючи винагороду або покарання за свої дії. Це

ефективно для задач, де є чітка ціль, така як гра в шахи. Однак, цей метод може бути повільним та вимагає великої кількості випробувань та помилок.

Штучні нейронні мережі Tensorflow використовують навчання з учителем.

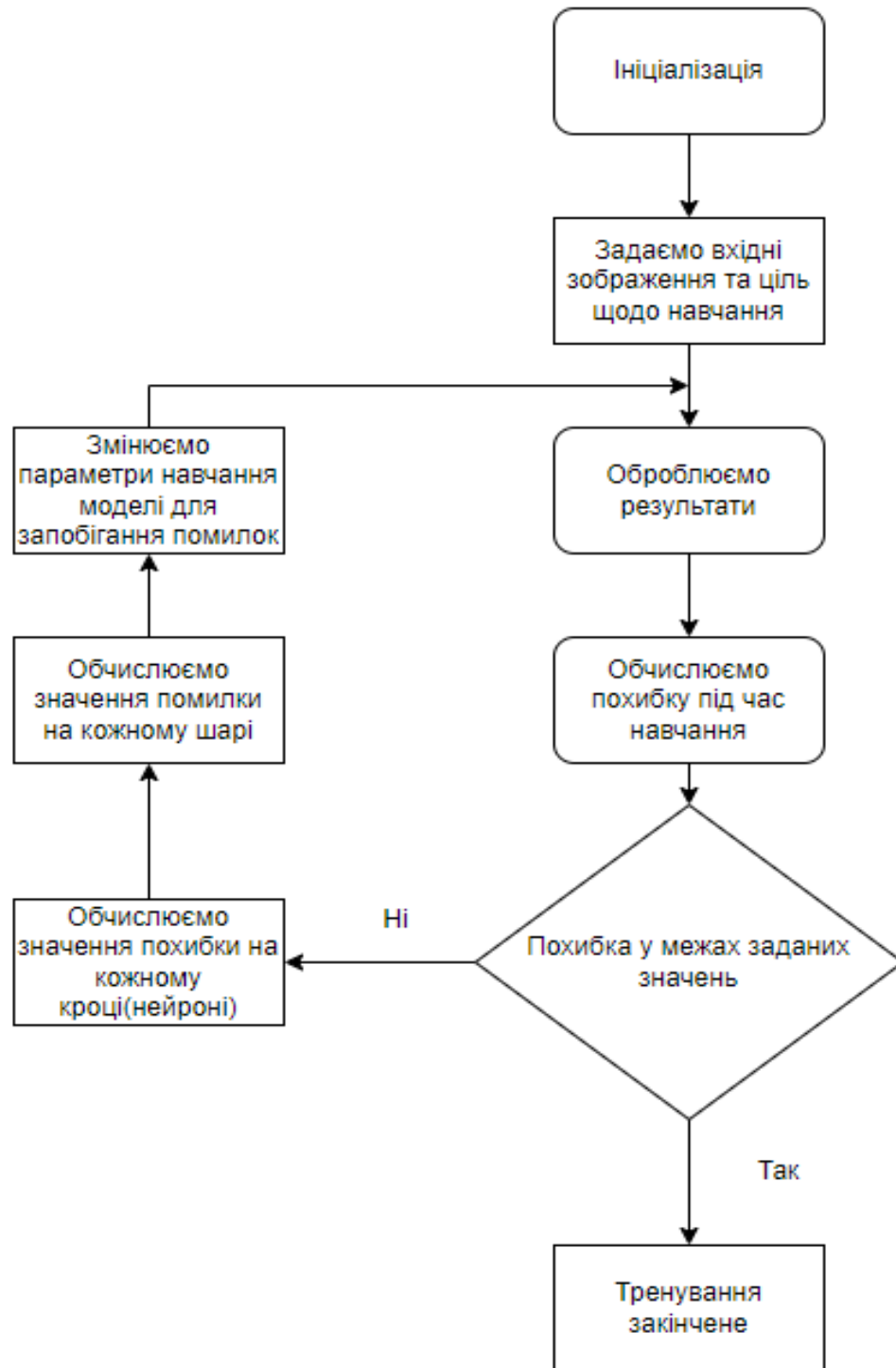


Рисунок 3.1 – Процес навчання моделі

Оскільки було використано датасет Fer2013, необхідно було виокремити зображення згідно емоцій. Датасет FER2013 (Facial Expression Recognition 2013) містить зображення обличч людей, які виражають сім основних емоцій. Кожне зображення має мітку, що вказує на одну з наступних емоцій. Емоції, що представлені у датасеті наведено у таблиці 3.1. Приклад зображень датасету Fer2013 наведений на рисунку 3.2.

Ці класи допомагають моделі визначити конкретну емоцію, яку виражає обличчя на зображенні. Зазначимо, що кількість зображень для кожної емоції може відрізнятися. Наприклад, вираз “Відраза” має найменшу кількість зображень - близько 600, тоді як інші класи мають приблизно по 5000 зразків кожен. Зважаючи на те, що кількість зображень різна, то і точність розпізнавання певних емоцій може бути різною.

Таблиця 3.1 – Емоції у датасеті Fer2013 [70]

Номер у датасеті	Емоція	Опис емоції
0	Angry	Вираз обличчя виражає гнів
1	Disgust	Емоція – відраза
2	Fear	Людина відчуває страх
3	Happy	Людина щаслива та задоволена
4	Sad	Емоція на фото – сум
5	Surprise	Вираз обличчя – здивування
6	Neutral	Людини на фото у стані спокою

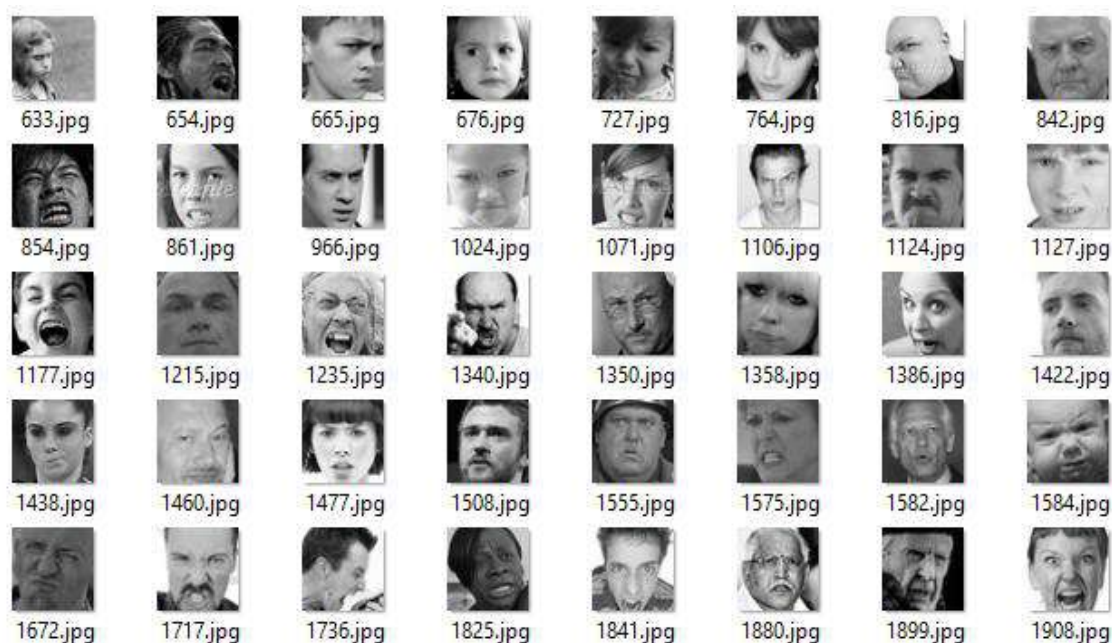


Рисунок 3.2 – Приклад зображень датасету Fer2013 [70]

Для кращого результату тренування моделі було вирішено використати власні зображення, щоб модель змогла більш якісно навчитися розпізнавати емоції. Для цього було використано сервіс Roboflow, який дозволяє створювати власні датасети та допомагає з найменуванням зображень для навчання з учителем, що підходить до цього дослідження та для Tensorflow вцілому. Для початку роботи необхідно створити проект та базові налаштування. Процес створення проекту зображений на рисунку 3.3. Після вказання усього необхідного натискаємо Create Public Project. Далі, сервіс пропонує завантажити зображення для навчання, для створення датасету. Процес завантаження зображень вказано на рисунку 3.4. Наступним кроком буде збереження та перехід до маркування зображень. Основний екран для маркування зображень представлений на рисунку 3.5. Для зручності маркування було використано метод побудови обмежувальних рамок та вказання емоції, яка представлена на рисунку. Процес маркування показаний на рисунках 3.6 – 3.7. Так само необхідно промаркувати кожне зображення датасету, для коректної роботи та навчання моделі. Останнім кроком є вказання загальних налаштувань датасету (розмір зображень, аугментація тощо). Усі налаштування зображені на рисунках 3.8 – 3.10.

## Let's create your project.

New Workspace > New Public Project

Project Name

Facial Expression

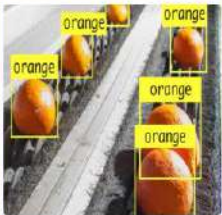
License

CC BY 4.0

Annotation Group


Emotion-Facial

Project Type



**Object Detection**  
Identify objects and their positions with bounding boxes.


Best For  
# Counting # Tracking



**Classification**  
Assign labels to the entire image.


Classification Type  
 Multi-Label  Single-Label

Best For  
# Filtering # Content Moderation



**Instance Segmentation**  
Detect multiple objects and their actual shape.

Best For  
# Measurements # Odd Shapes



**Keypoint Detection**  
Identify keypoints ('skeletons') to subjects.

Best For  
# Pose Estimation

Show More

Cancel

Create Public Project

Рисунок 3.3 – Створення проекту у Roboflow

Drag and drop images, annotations, and videos.

Select Files Select Folder

in .jpg, .png, .webp in 16 forests in .avi, .mp4, .mkv




Рисунок 3.4 – Завантаження зображень

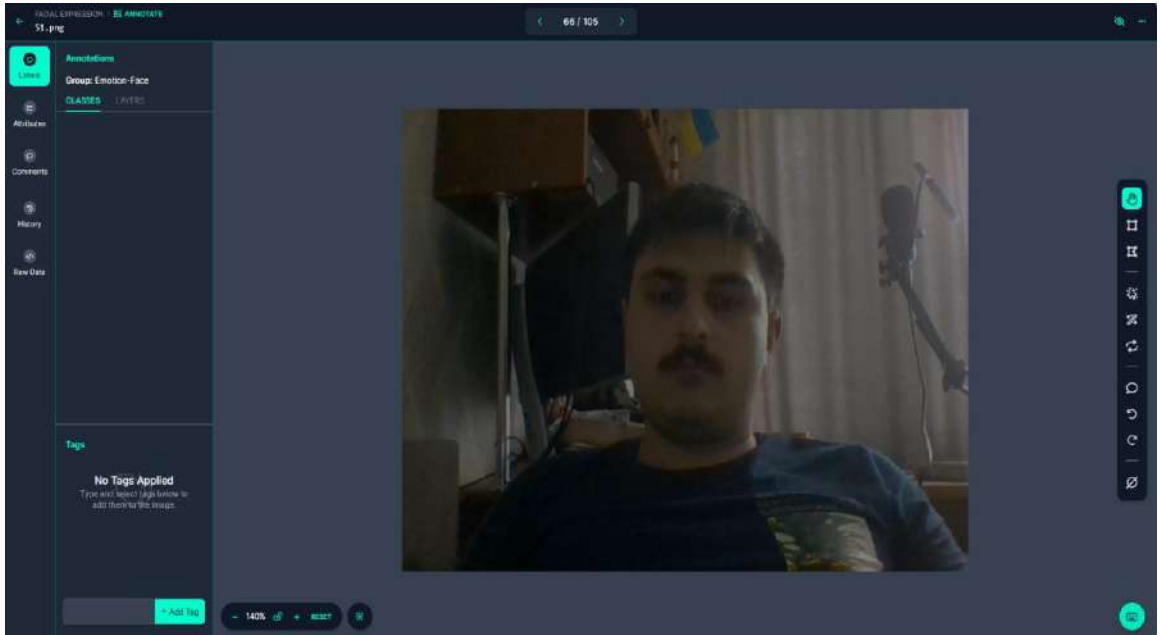


Рисунок 3.5 – Головний екран маркування зображень

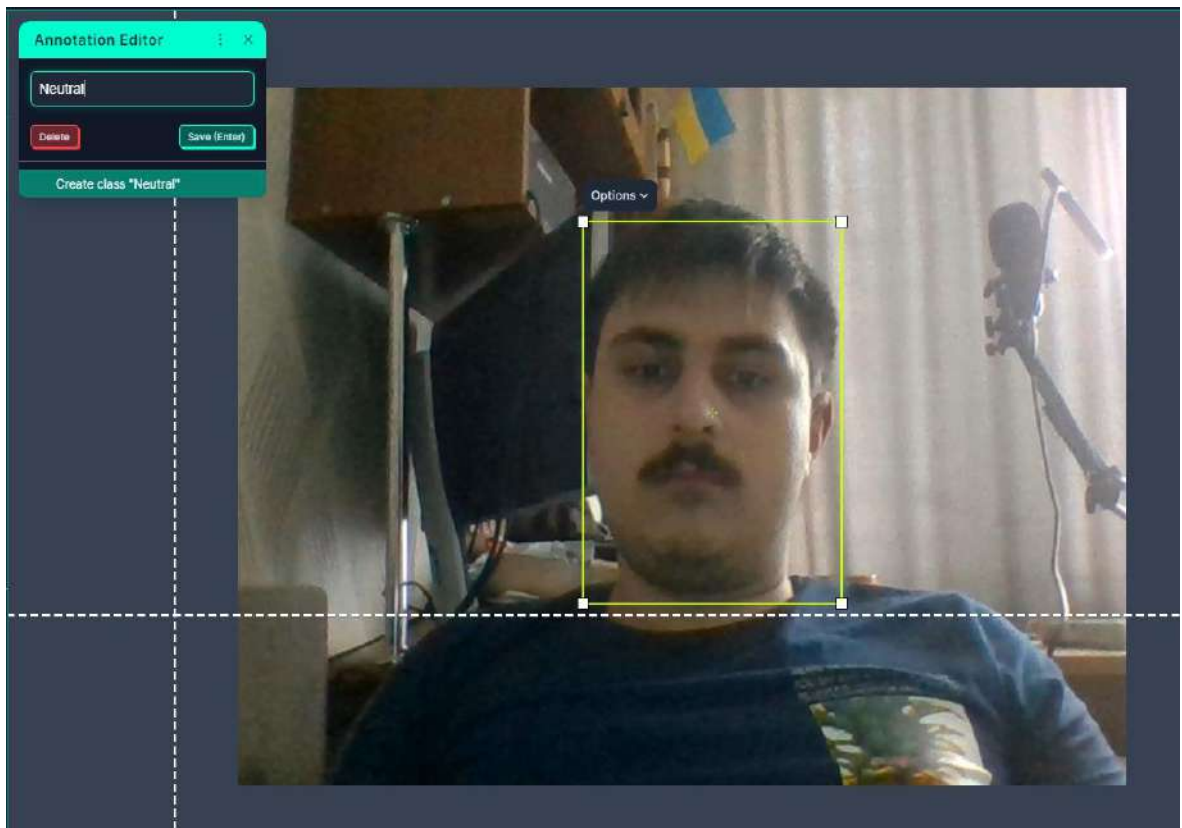


Рисунок 3.6 – Процес маркування (Neutral)

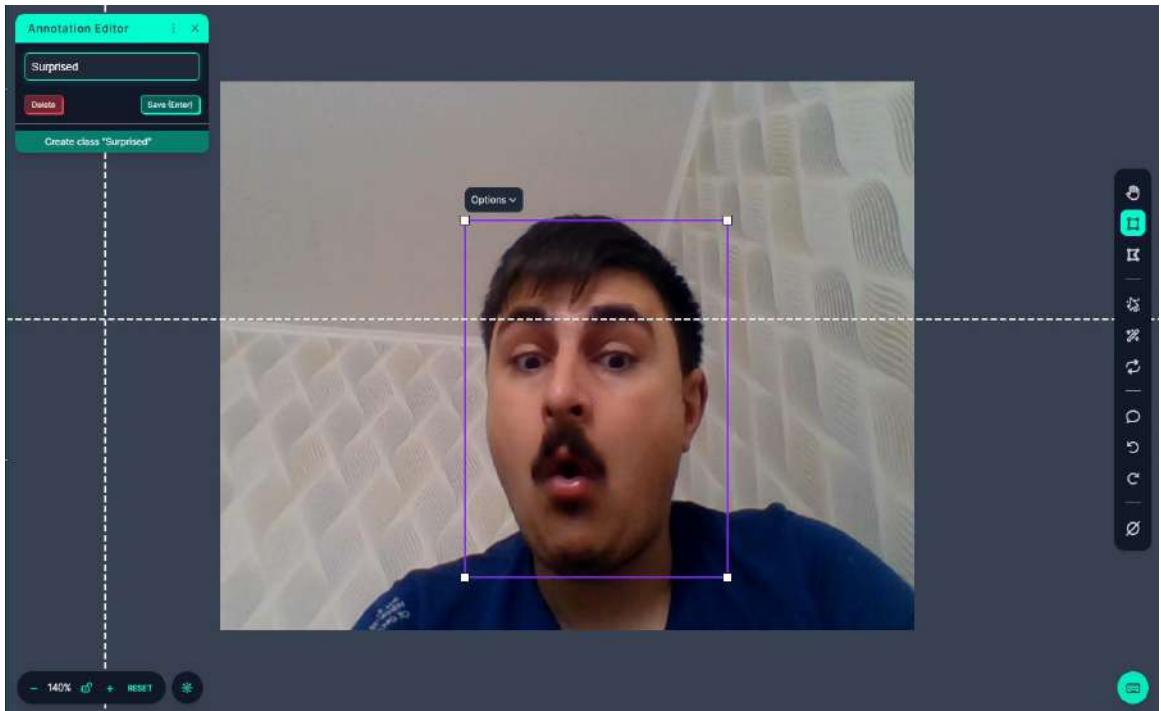


Рисунок 3.7 – Процес маркування (Surprised)

**3** Preprocessing

[What can preprocessing do?](#)

Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient	Edit	×
Resize Stretch to 640×640	Edit	×
<a href="#">+</a> Add Preprocessing Step		

[Continue](#)

Рисунок 3.8 – Налаштування розміру зображень

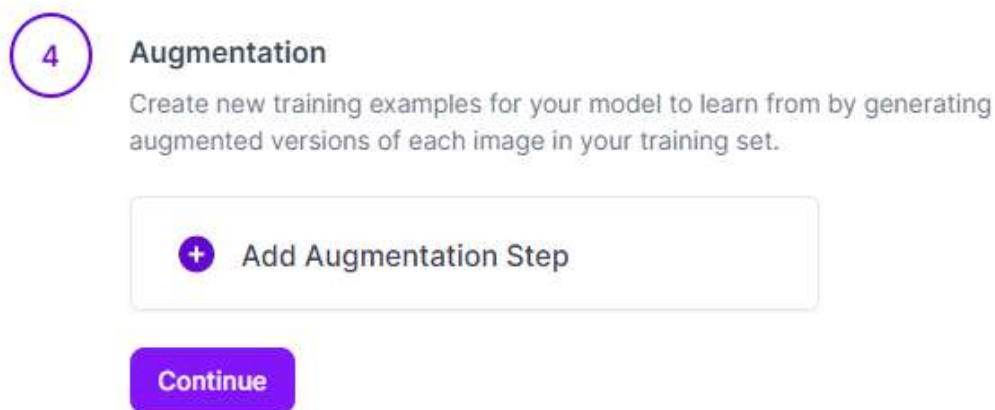


Рисунок 3.9 – Налаштування аугментації

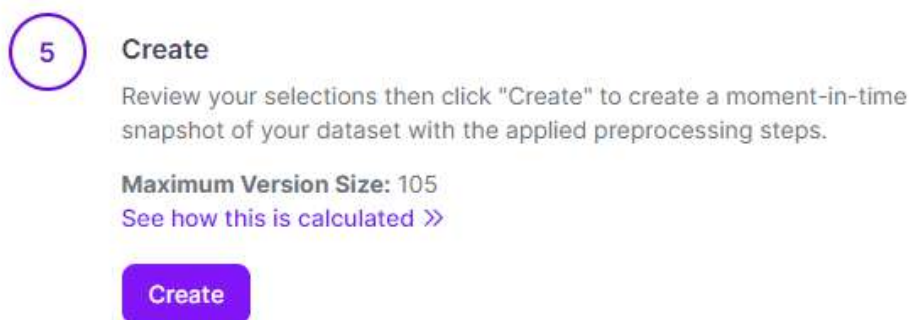


Рисунок 3.10 – Завершення налаштувань та підтвердження створення

Після цього вибираємо для якої саме моделі необхідно налаштувати датасет та завантажуюмо його для подальшого навчання моделі.

Для зручності, вказуємо, що завантаження необхідно здійснити для моделі Tensorflow, оскільки під час вибору доступна не лише модель, яка нам потрібна, а й інші моделі, створені для розпізнавання та класифікації об'єктів, зокрема YOLOv8, YOLOv9 та інші.

Процес вибору та завантаження датасету зображений на рисунках 3.11 – 3.12.

Після завантаження архіву з датасетом та маркуваннями, можна перевірити маркування зображень шляхом відкриття .csv файлу з анотаціями. Там будуть вказані усі параметри моделі.

Приклад вмісту .csv файлу зображений на рисунку 3.13.

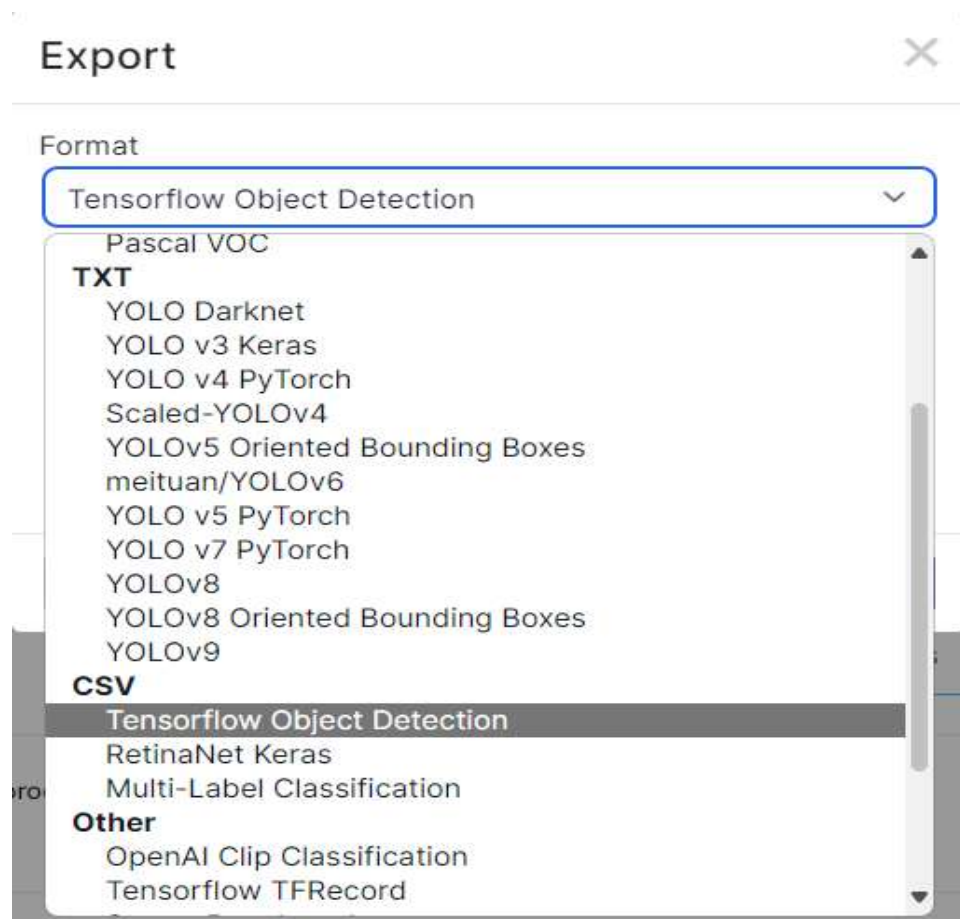


Рисунок 3.11 – Вибір моделі для датасету

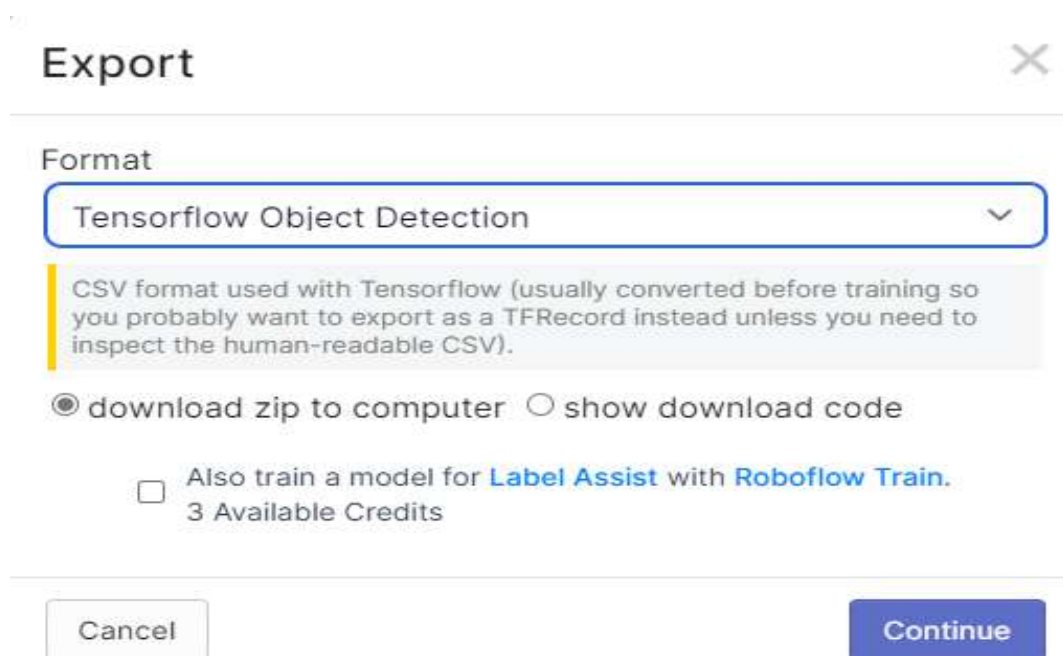


Рисунок 3.12 – Підтвердження завантаження

	A	B	C	D	E	F	G	H
1	filename	width	height	class	xmin	ymin	xmax	ymax
2	24_png.rf.1c3a6423e5ff09cabee828ab199d0973.jpg	640	640	Fear	237	205	464	609
3	71_png.rf.33e9dc653facc8d7f0eb45361110130.jpg	640	640	Sad	204	105	426	492
4	44_png.rf.1e92bc4aaea0cfd459ad566f292c74bc.jpg	640	640	Happy	226	100	433	492
5	75_png.rf.0dc63957f9857e73a63016ec2a7bf131.jpg	640	640	Sad	232	109	434	481
6	79_png.rf.2a8ae3aef08ecd99ea437e03a8c398b.jpg	640	640	Surprised	225	133	459	597
7	37_png.rf.2e1a505eec44258800425ff6be616e0a.jpg	640	640	Happy	231	95	428	490
8	captured_image_11_png.rf.184381477173ec3bc47cc987c223e7d.jpg	640	640	Angry	230	94	444	529
9	26_png.rf.35e09a23d2cfa65d28abe9f477bb4fc1.jpg	640	640	Fear	250	177	469	612
10	64_png.rf.257bc1b87707558abd461e0869a785c1.jpg	640	640	Sad	235	96	446	503
11	78_png.rf.0921351535c2e14297d2bc6ecbe969d.jpg	640	640	Surprised	229	154	457	581
12	2_png.rf.23d802444575b1eab8c79666514ffed0.jpg	640	640	Disgust	195	132	407	509
13	90_png.rf.3b2d8ca2769c342249c78be6a941731a.jpg	640	640	Surprised	224	131	490	617
14	80_png.rf.1929c88bddadb99881ecb353a6b16dd9.jpg	640	640	Surprised	222	125	467	598
15	22_png.rf.23155cb0e07553bb4314785ad2332e20.jpg	640	640	Fear	248	217	456	606
16	66_png.rf.0536e4b48f804706a1a957e1c7336d75.jpg	640	640	Sad	214	111	431	512
17	73_png.rf.417bb0454af3dd2af2ddc3f611de2cd6.jpg	640	640	Sad	205	99	422	497
18	53_png.rf.2dad3f44c2aa8faffbda051a47eb6463.jpg	640	640	Neutral	235	92	431	490
19	38_png.rf.29a49656ba2d019435d4229ed180f9c.jpg	640	640	Happy	227	97	435	478
20	7_png.rf.3c067b760550d56f861847235f2b7258.jpg	640	640	Disgust	207	119	409	501
21	35_png.rf.23ba17a971af9b45dde49707e73deafc.jpg	640	640	Happy	242	96	422	478
22	87_png.rf.441b6920771f91840391ec549398b68.jpg	640	640	Surprised	233	133	484	617
23	captured_image_0-checkpoint_png.rf.13a7d3b430bc9c19425684b4577cdd6.jpg	640	640	Angry	205	134	436	505
24	51_png.rf.41ffe9acc171846e13eb8f571c65a81b.jpg	640	640	Neutral	249	123	435	473
25	20_png.rf.492211078e6f6b0f655fa7104573e05c.jpg	640	640	Fear	252	219	452	607
26	65_png.rf.4d2fa50f62afd08326d0e2cf4ba66123.jpg	640	640	Sad	216	102	439	503
27	captured_image_12_png.rf.4fb5e09076683091738367a23acd7df7.jpg	640	640	Angry	226	111	436	530
28	54_png.rf.505575c49c88003db5772995ad230c9a.jpg	640	640	Neutral	241	111	428	466
29	captured_image_8_png.rf.6c75e63a160be7396525537bca6cac20.jpg	640	640	Angry	235	122	447	495
30	27_png.rf.61b91917213d0523289f32fd8845cd05.jpg	640	640	Fear	239	185	466	590
31	19_png.rf.4bb7e5568845d44c012d2d17f0a4798.jpg	640	640	Fear	237	205	452	580
32	45_png.rf.419ec9f7b764dba6bbf5bc1a35b6fe37.jpg	640	640	Happy	237	102	457	495
33	33_png.rf.5cc6dea371e30696f6d975da2e8e7bf4.jpg	640	640	Happy	232	77	434	487
34	46_png.rf.8736d8150621c8299241beb00e332b9a.jpg	640	640	Neutral	224	103	439	498
35	63_png.rf.76470f830409480b10f6f745e254afb.jpg	640	640	Sad	223	92	435	489

Рисунок 3.13 – Вміст annotations.csv

### 3.2 Алгоритмічні основи тренування мережі для визначення емоцій

Для тренування вибраної моделі було використано середовище Jupyter Notebook. Для входу до Jupyter Notebook, після завантаження, необхідно в терміналі написати команду для входу. Процес входу у середовище зображений на рисунку 3.14. Перед початком роботи необхідно завантажити усі бібліотеки для роботи моделі. Різновиди бібліотек та їх взаємодія між собою зображені на рисунку 3.15. Опис бібліотек, необхідних для моделі вказаний у таблиці 3.2.

```
C:\Users\timha>jupyter notebook
[I 2024-07-03 12:36:47.734 ServerApp] Package notebook took 0.0000s to import
[I 2024-07-03 12:36:47.780 ServerApp] Package jupyter_lsp took 0.0425s to import
[W 2024-07-03 12:36:47.780 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp.
function name will be deprecated in future releases of Jupyter Server.
[I 2024-07-03 12:36:47.804 ServerApp] Package jupyter_server_terminals took 0.0294s to import
[I 2024-07-03 12:36:47.804 ServerApp] Package jupyterlab took 0.0000s to import
[I 2024-07-03 12:36:47.963 ServerApp] Package notebook_shim took 0.0000s to import
[W 2024-07-03 12:36:47.963 ServerApp] A '_jupyter_server_extension_points' function was not found in notebook_shim
```

Рисунок 3.14 – Процес входу у Jupyter Notebook

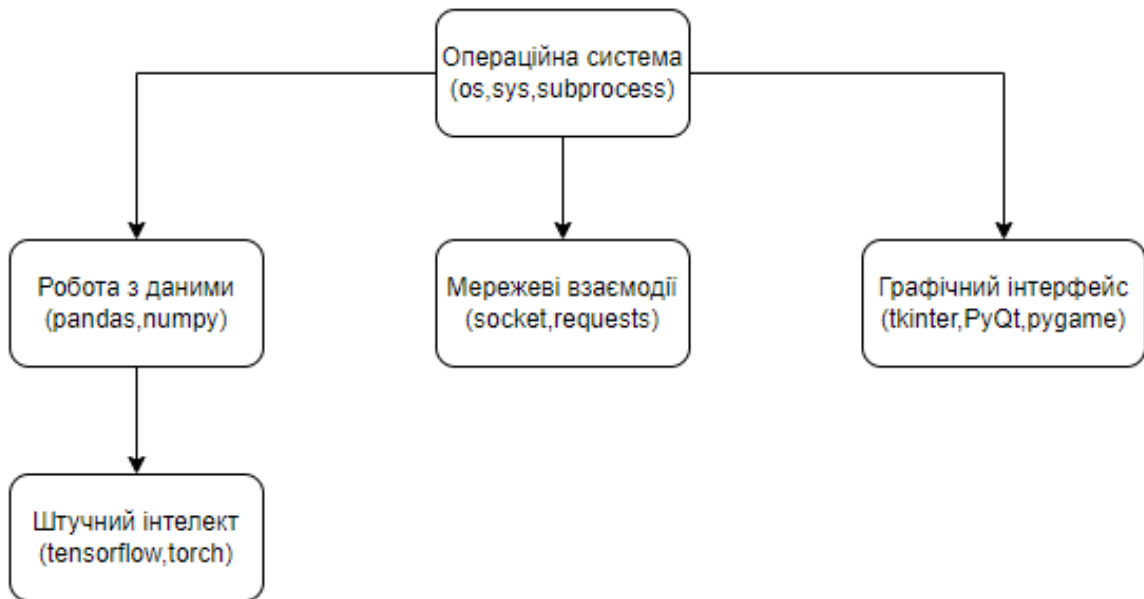


Рисунок 3.15 – Різновиди бібліотек та їх взаємодія між собою

Таблиця 3.2 – Опис бібліотек, необхідних для моделі

Назва бібліотеки	Опис та призначення
<code>keras.utils.to_categorical</code>	Використовується для конвертації вектора класів в бінарну матрицю
<code>keras.preprocessing.image.load_img</code>	Використовується для завантаження зображення у формат PIL для зручності використання
<code>keras.models.Sequential</code>	Sequential - модель, яка використовується для створення нейронних мереж, які складаються з послідовного стеку шарів
<code>keras.layers.Dense</code>	Dense - шар, який використовується для створення повнозв'язного шару нейронів в нейронній мережі для стабільності роботи нейронної мережі

Кінець таблиці 3.2

Назва бібліотеки	Опис та призначення
<code>keras.layers.Conv2D</code>	<code>Conv2D</code> - шар, який використовується для створення двовимірного шару згортки в нейронній мережі
<code>keras.layers.Flatten</code>	<code>Flatten</code> - це шар, який використовується для перетворення вхідних даних в одновимірний вектор
<code>keras.layers.Dropout</code>	<code>Dropout</code> - шар, який використовується для випадкового відключення деяких нейронів в шарі під час тренування
<code>keras.layers.MaxPooling2D</code>	<code>MaxPooling2D</code> - це шар, який використовується для застосування операції максимального пулінгу до двовимірних даних
<code>Os</code>	Це модуль Python, який надає функції для взаємодії з операційною системою
<code>Pandas</code>	Це бібліотека Python, яка надає високопродуктивні, прості в використанні структури даних та інструменти аналізу даних
<code>Numpy</code>	Це бібліотека Python, яка надає підтримку великим многовимірним масивам та матрицям, разом з великою бібліотекою математичних функцій

Після того, як було завантажено усі необхідні бібліотеки для роботи моделі, необхідно завантажити зображення для навчання моделі. Оскільки, зображення вже заздалегідь поділені на емоції, то зображення було завантажено шляхом частинки коду, яка дозволяє завантажити фото та згрупувати їх для навчання та тестування. Процес виконання завантаження зображень наведено на рисунку 3.16. Наступний крок – це налаштування метрик для навчання моделі. Це є найважливішим кроком, оскільки від налаштувань залежить продуктивність та точність моделі. Поетапне налаштування моделі зображено на рисунку 3.17.

```

train_features = extract_features(train['image'])
0%|          | 0/28821 [00:00<?, ?it/s]
C:\Users\timha\AppData\Local\Programs\Python\Python311'
ease use color_mode = "grayscale"
warnings.warn('grayscale is deprecated. Please use '
test_features = extract_features(test['image'])
0%|          | 0/7066 [00:00<?, ?it/s]

```

Рисунок 3.16 – Процес виконання завантаження зображень

Налаштування для тренування моделі включають у себе:

- `model = Sequential()`. Створює нову послідовну модель. Послідовна модель - це тип моделі, який складається з ряду шарів, де кожен шар має рівно одне вхідне та вихідне з'єднання;
- `model.add(Conv2D(128, kernel_size=(3,3), activation='relu', input_shape=(48,48,1)))`. Додає перший шар згортки до моделі. Шар згортки використовується для виявлення просторових шаблонів у вхідних даних. У цьому випадку він має 128 фільтрів, розмір ядра 3x3, функцію активації ReLU та форму вхідних даних 48x48x1;

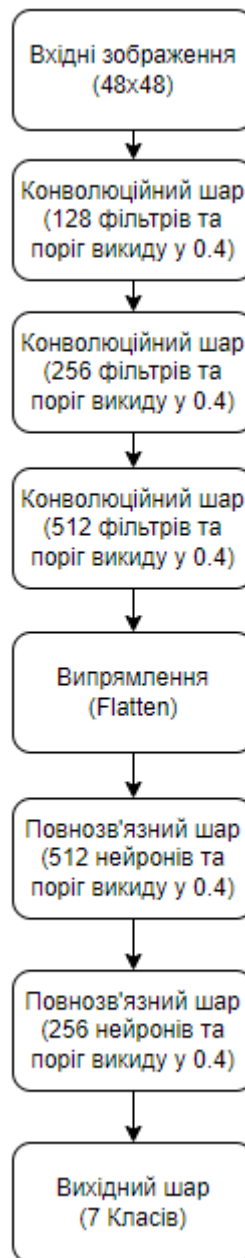


Рисунок 3.17 – Поетапне налаштування моделі для тренування

- `model.add(MaxPooling2D(pool_size=(2,2)))`. Додає шар максимального пулінгу до моделі. Шар максимального пулінгу використовується для зменшення розміру вхідних даних, зберігаючи при цьому найважливіші характеристики;
- `model.add(Dropout(0.4))`. Додає шар відмови до моделі. Шар відмови використовується для випадкового відключення деяких нейронів в шарі під час тренування, що допомагає запобігти перенавчанню;

- `model.add(Flatten())`. Додає шар згладжування до моделі. Шар згладжування використовується для перетворення вхідних даних в одновимірний вектор;
- `model.add(Dense(512, activation='relu'))`. Додає повнозв'язний шар до моделі. Повнозв'язний шар використовується для виконання класифікації на основі виявлених шаблонів;
- `model.add(Dense(7, activation='softmax'))`. Додає вихідний шар до моделі. Вихідний шар використовується для виведення прогнозованих класів. У цьому випадку він має 7 нейронів (один для кожного класу) та функцію активації `softmax`, яка використовується для перетворення вихідних значень у ймовірності.

Після вказання налаштувань для моделі, вказуємо налаштування для самого процесу навчання, для легкості його проведення та зручності коригування у подальшому. Процес тренування моделі зображено на рисунку 3.18.

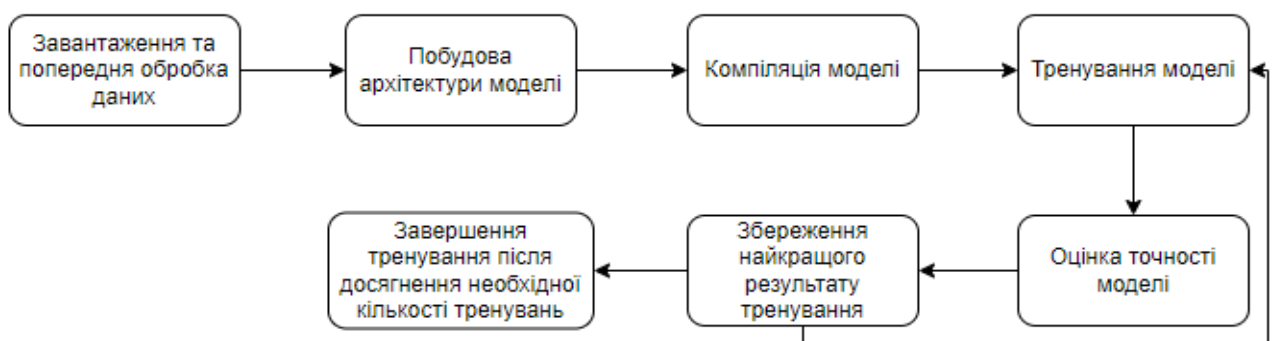


Рисунок 3.18 – Поетапний процес тренування моделі

Після тренування моделі було отримано модель формату `emotiondetection.h5`, що є одним із розширень моделі `Tensorflow` та дозволить використовувати дану модель надалі для роботи у сфері розпізнавання виразів обличчя людини. Окрім цього це дозволить перетворити модель у формат `Lite` для використання на малопотужних пристроях.

Після цього необхідно завантажити модель, щоб мати змогу її протестувати та доопрацювати при необхідності.

Необхідні команди для завантаження моделі вказані у таблиці 3.3.

Таблиця 3.3 – Опис команд для завантаження моделі та їх призначення

Команда	Опис та призначення
<code>model_json = model.to_json()</code>	Команда конвертує структуру моделі в формат JSON. Це не включає ваги моделі, лише її архітектуру
<code>with open("emotiondetector.json",'w') as json_file: json_file.write(model_json)</code>	Ця команда відкриває файл "emotiondetector.json" для запису ('w') і записує JSON представлення моделі в цей файл
<code>model.save("emotiondetector.h5")</code>	Команда зберігає всю модель, включаючи її архітектуру, ваги та стан оптимізатора, в один файл HDF5
<code>json_file = open("facialemotionmodel.json", "r")</code>	Команда відкриває файл для читання ('r')
<code>model_json = json_file.read()</code>	Команда читає вміст файлу і зберігає його в змінну <code>model_json</code>
<code>json_file.close()</code>	Команда закриває файл для звільнення ресурсів
<code>model = model_from_json(model_json)</code>	Команда створює нову модель з JSON, який щойно був прочитаний з файлу
<code>model.load_weights("facialemotionmodel.h5")</code>	Команда завантажує ваги моделі з файлу "facialemotionmodel.h5" в модель.

Так, як для дослідження потрібно модель TensorFlow Lite, необхідно перетворити модель у формат Lite.

Після того, як було створено модель TensorFlow, необхідно використати TensorFlow Lite Converter для перетворення моделі формату TensorFlow у формат TensorFlow Lite. Конвертер оптимізує модель для виконання на пристроях з обмеженими ресурсами.

Конвертер виконує кілька оптимізацій. Він зменшує розмір моделі, видаляючи непотрібні вузли з графа обчислень і замінюючи деякі операції на більш ефективні аналоги. Він також квантує ваги моделі, зменшуючи їх розмір і покращуючи швидкість виконання та ефективність використання енергії.

Після оптимізації модель зберігається в форматі файлу `.tflite`. Цей файл можна використовувати в додатках на пристроях, що використовують TensorFlow Lite. Перед початком конвертування необхідно завантажити бібліотеку tensorflow та усі необхідні параметри для її роботи. Процес перетворення моделі зображений на рисунку 3.20.

Команди, необхідні для конвертації моделі, наведені у таблиці 3.4.

Після проведення конвертації у доступі буде дві моделі для формату TensorFlow та TensorFlow Lite.

Таблиця 3.4 – Опис команд для конвертації моделі

Команда	Опис та призначення
<code>Import tensorflow as tf</code>	Завантажує бібліотеку TensorFlow для роботи з моделями машинного навчання
<code>Tf.lite.tfliteConverter.from_saved_model(saved_model_dir)</code>	Створює конвертер для перетворення моделі формату TensorFlow у формат TensorFlow Lite

Кінець таблиці 3.4

Команда	Опис та призначення
<code>Converter.convert()</code>	Конвертує модель у формат TF Lite, щоб вона була придатна для роботи на мобільних пристроях
<code>Open('model.tflite','wb') as f</code>	Команда відкриває модель у режимі запису
<code>f.write(tflite_model)</code>	Записує конвертовану модель

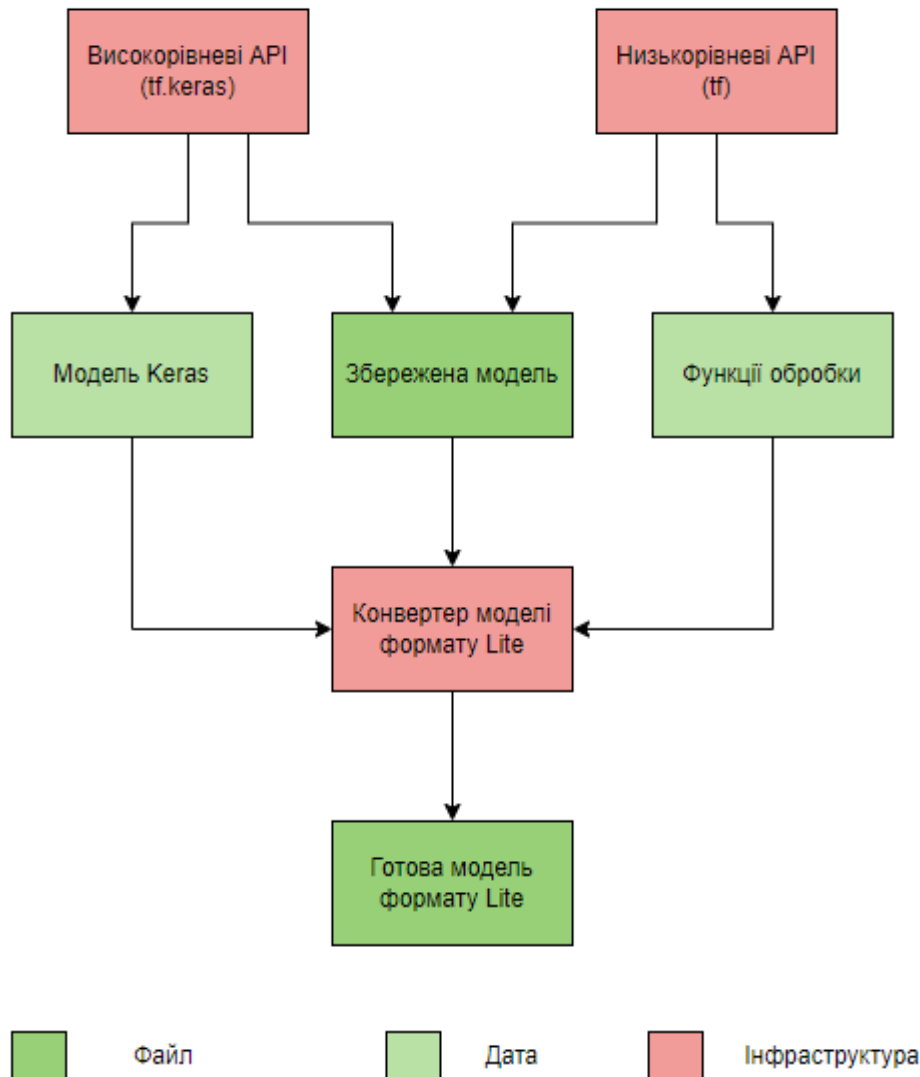


Рисунок 3.20 – Процес перетворення моделі у формат Lite

### 3.3 Висновок до третього розділу

У цьому розділі було виконано значну роботу з розробки моделей машинного навчання за допомогою TensorFlow та TensorFlow Lite для розпізнавання емоцій. Моделі TensorFlow та TensorFlow Lite були розроблені з метою створення ефективних та оптимізованих моделей для розпізнавання емоцій на зображеннях облич людей. TensorFlow Lite, зокрема, був вибраний через його здатність працювати на мобільних пристроях та пристроях з обмеженою продуктивністю, що робить його ідеальним для реального використання в мобільних застосунках.

Датасет FER2013 був використаний як основний датасет для тренування моделей. Цей датасет містить велику кількість зображень облич людей, які виражають різні емоції, що робить його відмінним ресурсом для тренування моделей розпізнавання емоцій.

Крім того, було створено власний датасет з емоціями за допомогою сервісу Roboflow. Roboflow - це потужний інструмент, який дозволяє легко створювати, аугментувати та управляти датасетами для тренування моделей машинного навчання. Використання Roboflow дозволило створити власний датасет, який був спеціально адаптований під потреби проекту, що додатково покращило якість та ефективність тренування моделей.

В цілому, цей розділ показав, як можна розробити ефективні моделі для розпізнавання емоцій за допомогою TensorFlow та TensorFlow Lite, використовуючи як готові датасети, так і власні. Це підкреслює важливість вибору правильних інструментів та датасетів для досягнення найкращих результатів при розробці моделей машинного навчання.

## 4 РЕЗУЛЬТАТИ ТРЕНУВАННЯ МОДЕЛІ

### 4.1 Аналіз результатів тренування моделі

Після завершення усіх етапів, описаних у третьому розділі, у результаті було отримано моделі Tensorflow та Tensorflow Lite. Для цього дослідження необхідним є лише модель Tensorflow Lite, тому аналіз метрик та графів буде проведено саме для цієї моделі.

Найважливішим для моделі є точність, адже саме вона визначає скільки правдивих результатів буде після навчання моделі. Загальна точність моделі зображена на рисунку 4.1.

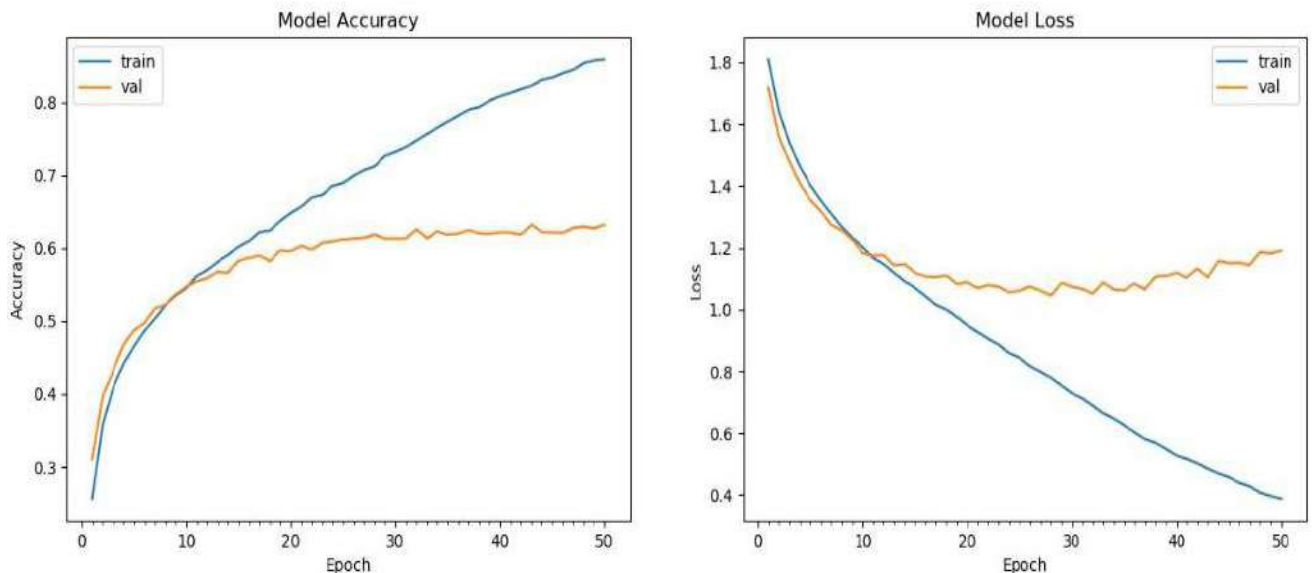


Рисунок 4.1 – Загальна точність моделі Tensorflow Lite

На графіку видно, що модель має середню точність близько 77%, що для моделі, розрахованої на мобільні пристрої є дуже гарним результатом. Також, у результаті тренування мережа має низькі показники помилок, що свідчить про достойні результати тренування. На рисунках 4.2 – 4.5 зображені основні метрики по закінченню навчання моделі. Основні характеристики:

– `train/box_loss`. Це втрати обмежувальної рамки (bounding box) на тренувальному наборі даних. Ця метрика вимірює, наскільки добре модель може визначити положення об'єктів на зображенні;

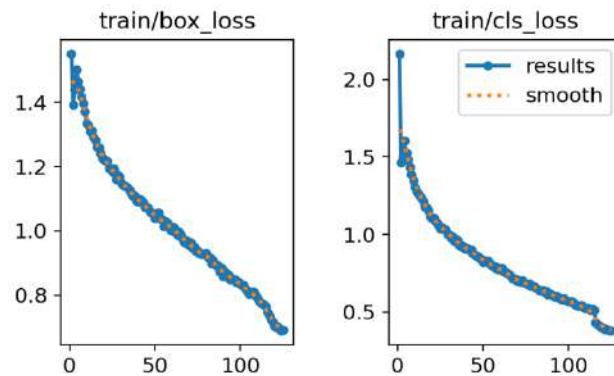


Рисунок 4.2 – Показання `box_loss` та `cls_loss` для моделі TF Lite на тренувальному наборі даних

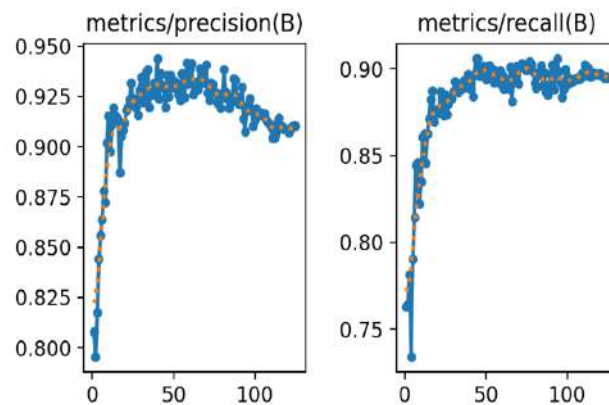


Рисунок 4.3 – Показання `precision` та `recall` для моделі TF Lite на тренувальному датасеті

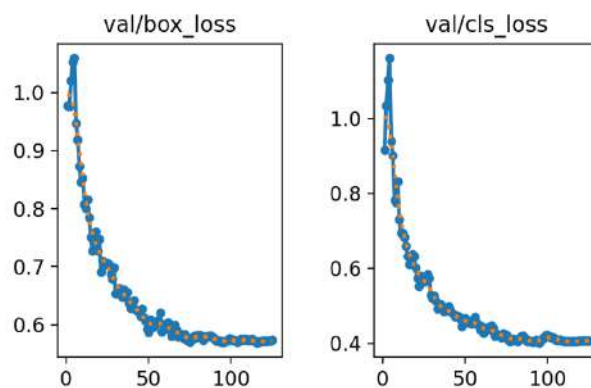


Рисунок 4.4 – Показання `box_loss` та `cls_loss` для моделі TF Lite на тестовому датасеті

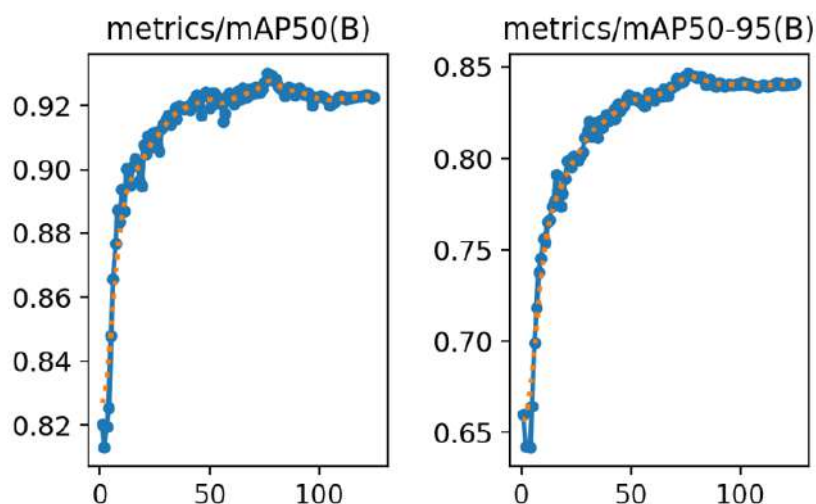


Рисунок 4.5 – Показання mAP50 та mAP50-95 для моделі TF Lite на тестовому датасеті

- `train/cls_loss`. Це втрати класифікації на тренувальному наборі даних. Ця метрика вимірює, наскільки добре модель може визначити клас об'єктів на зображенні;
- `metrics/precision(B)`. Це точність (precision) моделі. Ця метрика вимірює відсоток правильно визначених об'єктів з усіх об'єктів, які модель визначила як певний клас;
- `metrics/recall(B)`. Це повнота (recall) моделі. Ця метрика вимірює відсоток правильно визначених об'єктів з усіх дійсних об'єктів певного класу;
- `metrics/mAP50(B)`. Це середня точність (mean Average Precision) при пороговому значенні IoU (Intersection over Union) 0.50. Ця метрика вимірює якість моделі за допомогою одного числа, враховуючи і точність, і повноту;
- `metrics/mAP50-95(B)`. Це середня точність (mean Average Precision) при порогових значеннях IoU від 0.50 до 0.95 з кроком 0.05. Ця метрика вимірює якість моделі за допомогою одного числа, враховуючи і точність, і повноту при різних порогових значеннях IoU;
- Значення з позначкою `val` аналогічні, але для тестового набору даних. Наступними метриками є матриці помилок (звичайна та нормалізована).

Матриця помилок (Confusion Matrix) - це таблиця, яка використовується для опису професійності моделі класифікації.

Вона допомагає нам зрозуміти, де модель робить помилки. Вона складається з чотирьох компонентів:

- true positives (TP). Це кількість позитивних випадків, які модель правильно класифікувала як позитивні. Якщо модель отримала зображення обличчя людини, що демонструє "радість", і правильно визначила його як "радість", це вважається True Positive. Наприклад, якщо модель аналізує зображення усміхненої людини і класифікує її як "щасливу", то цей випадок додається до TP;

- true negatives (TN). Це кількість негативних випадків, які модель правильно класифікувала як негативні. Якщо людина на фото має нейтральний вираз обличчя, і модель правильно класифікує його як "нейтральний", це True Negative. Тобто модель не помилково приписує якісь емоції;

- false positives (FP). Це кількість негативних випадків, які модель помилково класифікувала як позитивні. Це помилка, коли модель помилково класифікує нейтральний вираз обличчя як "щастя". Наприклад, людина має спокійне обличчя без усмішки, але модель неправильно визначає її як "радісну";

- false negatives (FN). Це кількість позитивних випадків, які модель помилково класифікувала як негативні. Це ще серйозніша помилка: якщо людина справді виражає "злість", але модель не розпізнає це і класифікує її як нейтральну. Така ситуація може вплинути, наприклад, на системи безпеки, де важливо визначити агресивну поведінку.

Нормалізована матриця помилок (Normalized Confusion Matrix) - це просто матриця помилок, де кожне значення було поділено на суму значень у відповідному рядку.

Це допомагає нам краще зрозуміти, як модель виконується для кожного класу, незалежно від того, скільки прикладів кожного класу було в наборі даних. Зазначені матриці зображені на рисунках 4.6 – 4.7.

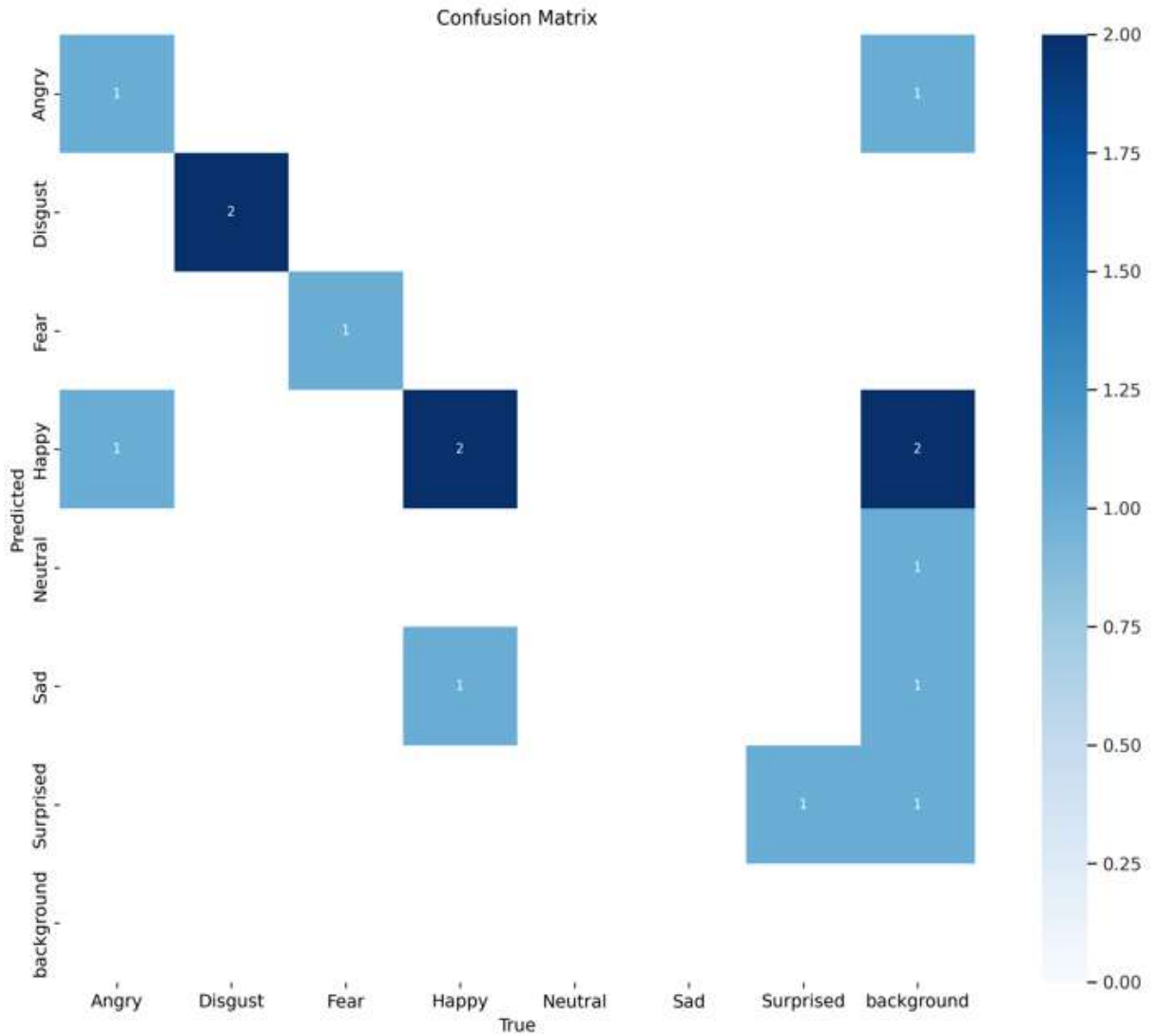


Рисунок 4.6 – Матриця помилок

Згідно усіх метрик можна сказати, що модель є добре навченою та класифікованою, що дозволяє використовувати її у роботі без жодних проблем.

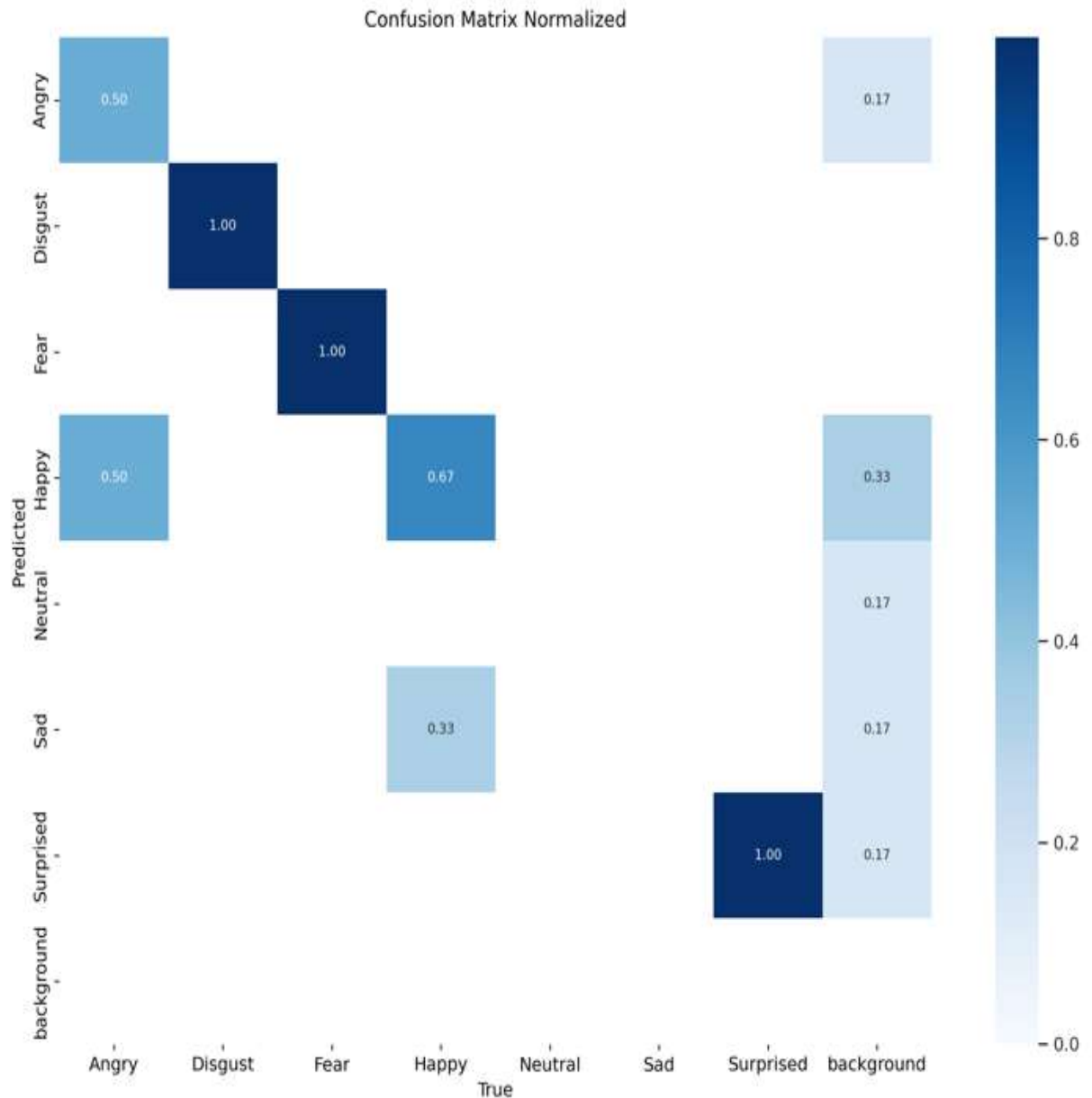


Рисунок 4.7 – Нормалізована матриця помилок

#### 4.2 Перевірка роботи моделі на мобільних пристроях

Щоб перевірити роботу Tensorflow Lite моделі у реальних умовах, необхідно використати мобільних пристрій з обмеженою продуктивністю. Для цього дослідження було використано одноплатний комп'ютер Raspberry pi 5, який має невеликі потужності, що дозволить перевірити роботу нейромережі та правильність її тренування. Raspberry pi 5 зображений на рисунку 4.8, характеристики плати наведені у таблиці 4.1.

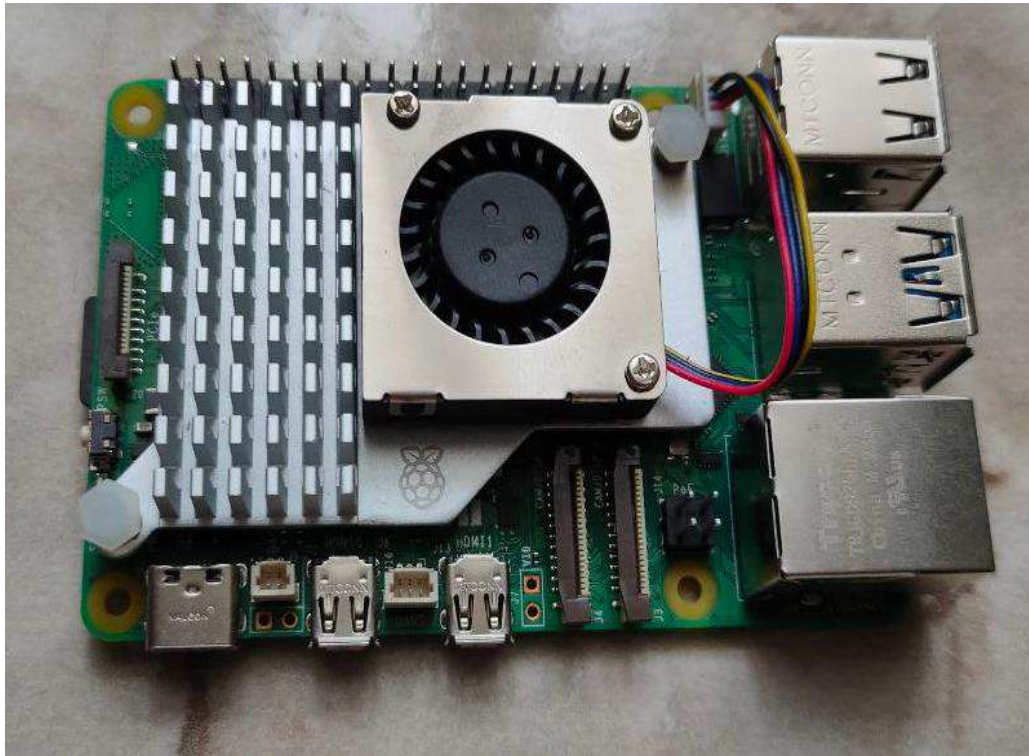


Рисунок 4.8 – Raspberry pi 5

Таблиця 4.1 – Характеристики Raspberry pi 5

Характеристика	Опис
Тип процесора	2.4GHz quad-core, 64-bit Arm Cortex-A76 CPU, з кешем L2 512KB на ядро та спільним кешем L3 2MB
Грфічне ядро	VideoCore VII
Кількість оперативної пам'яті та її вид	LPDDR4X-4267 SDRAM 8GB
Версія Wi-Fi	Двухдіапазонний 802.11ac
Версія Bluetooth	5.0 / Bluetooth Low Energy (BLE)
Кількість USB та їх види	2 x USB 3.0, 2 x USB 2.0
Швидкість Ethernet	Gigabit Ethernet

Кінець таблиці 4.1

Характеристика	Опис
Інтерфейси для камери/дисплея	2 x 4-канальні MIPI
Тип PCIe	PCIe 2.0 x1 інтерфейс для швидких периферійних пристроїв
Живлення для плати	5V/5A DC через USB-C, з підтримкою Power Delivery
GPIO (порти виводу/вводу)	Стандартний 40-контактний роз'єм Raspberry Pi
Годинник реального часу (RTC)	Налаштовується окремо та працює від зовнішньої батареї

Для зручності перевірки роботи неймережі на платі було використано редактор коду Visual Studio Code з усіма необхідними доповненнями для коректної роботи. Результати перевірки зображені на рисунках 4.9 – 4.11.

```
original image is of sad
1/1 [=====] - 0s 55ms/step
model prediction is sad
<matplotlib.image.AxesImage at 0x16abfe14e80>
```

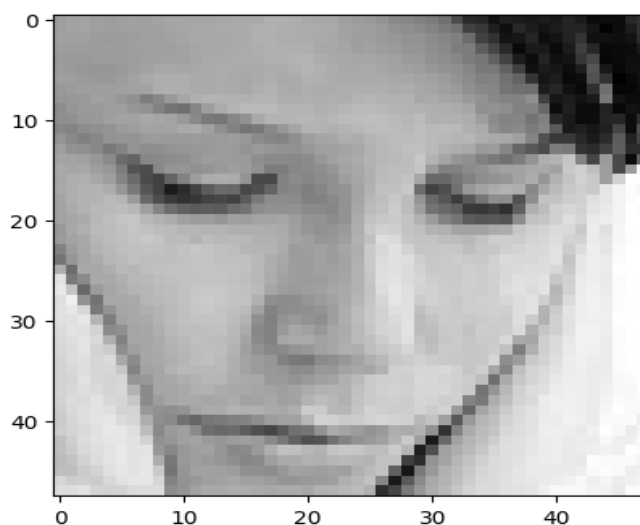


Рисунок 4.9 – Результат передбачення емоції (Сум)

```
original image is of happy  
1/1 [=====] - 0s 42ms/step  
model prediction is happy  
<matplotlib.image.AxesImage at 0x16ac00a8970>
```

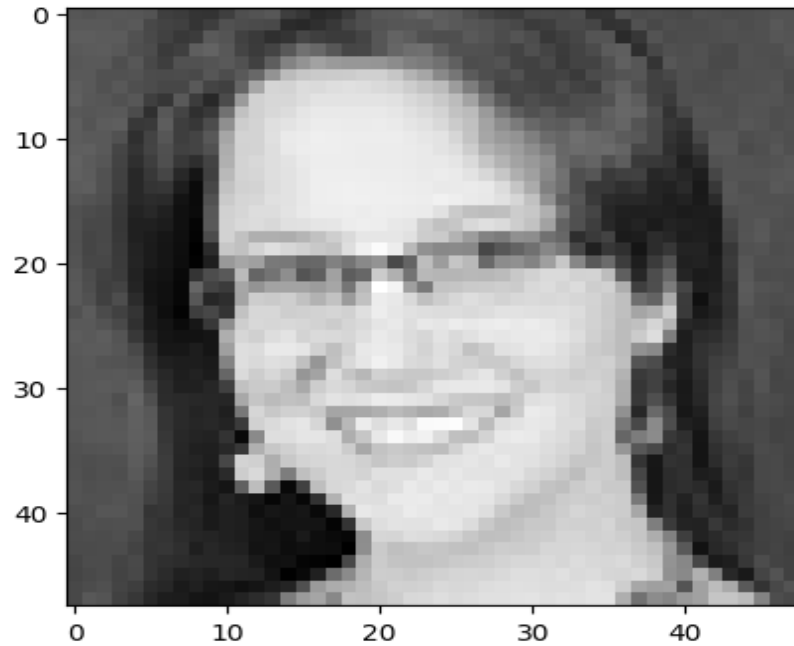


Рисунок 4.10 – Результат передбачення емоції (Радість)

```
original image is of disgust  
1/1 [=====] - 0s 57ms/step  
model prediction is disgust  
<matplotlib.image.AxesImage at 0x16abfef4d90>
```

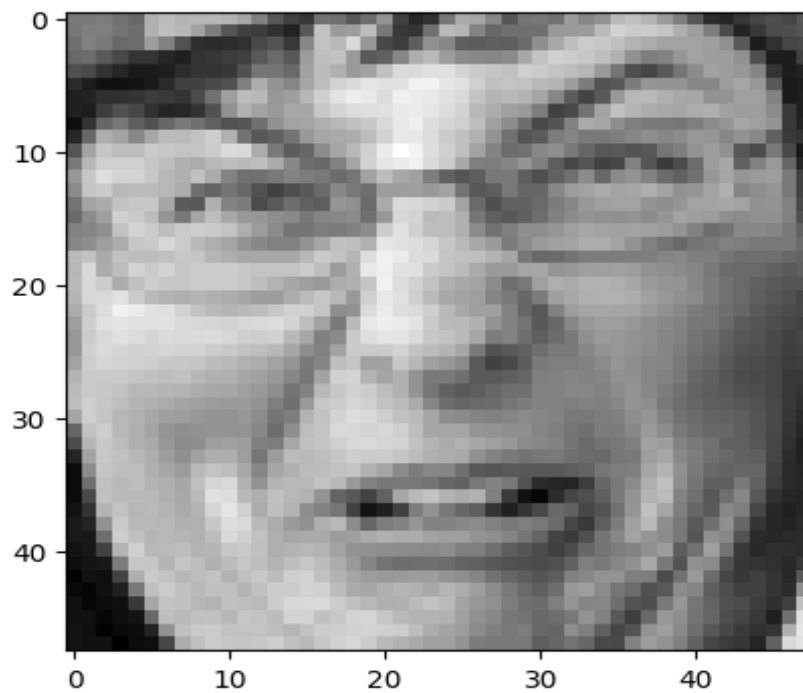


Рисунок 4.11 – Результат передбачення емоції (Відраза)

Для перевірки працездатності моделі у реальному часі, необхідно додатково приєднати веб-камеру до плати. У цьому дослідженні було використано камеру Okey WB280. Характеристики веб камери наведені у таблиці 4.2, зовнішній вигляд камери зображений на рисунку 4.12.

Таблиця 4.2 – Характеристики камери

Роздільна здатність відео камери	FullHD (1920x1080)
Поле огляду, градусів	90
Фокусування	Автофокус
Живлення для камери	USB
Основний роз'єм	USB 3.0/ USB 2.0
Кількість кадрів у секунду	30
Інтерфейс підключення	USB 2.0



Рисунок 4.12 – Зовнішній вигляд камери

Результати перевірки у реальному часі зображені на рисунках 4.13 – 4.15.



Рисунок 4.13 – Перевірка результатів (Neutral)



Рисунок 4.14 – Перевірка результатів (Sad)

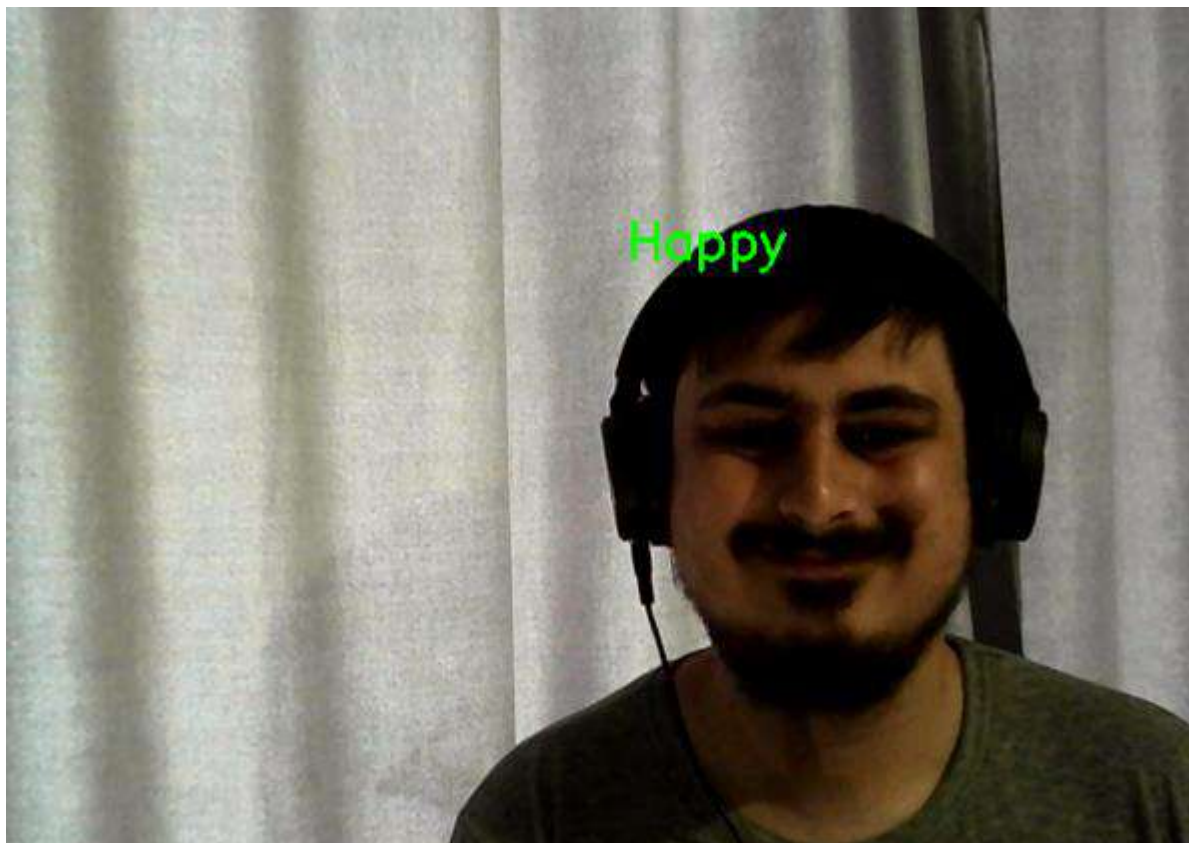


Рисунок 4.15 – Перевірка результатів (Happy)

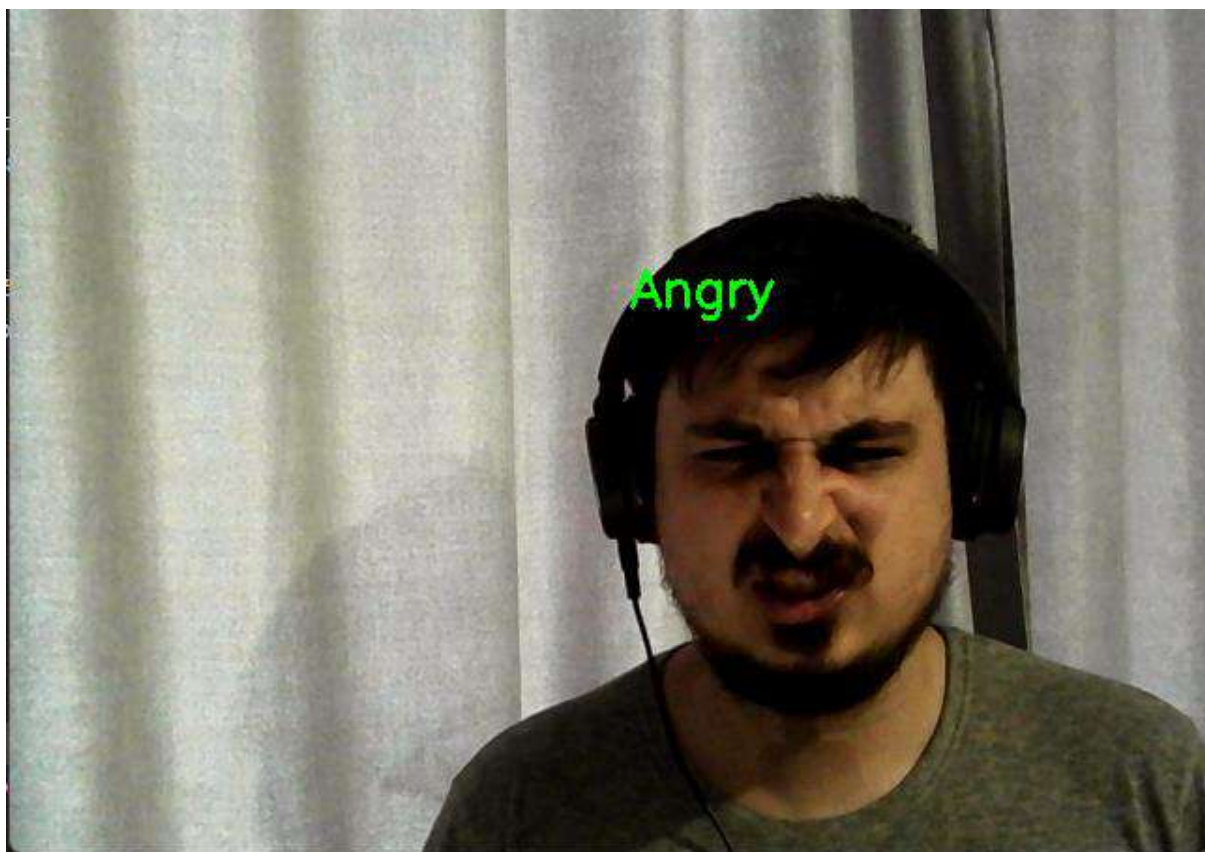


Рисунок 4.16 – Перевірка результатів (Angry)

### 4.3 Можливість покращення результатів

У цьому дослідженні було створено модель Tensorflow Lite для мобільних пристроїв, яку було перевірено на Raspberry pi 5. Щоб покращити результати, було вирішено використати прискорювач Coral, який розроблений для пришвидшення дії моделі в умовах, коли потужностей пристрою не вистачає.

USB-прискорювач Coral додає співпроцесор Edge TPU до системи користувача. Він оснащений роз'ємом USB-C, який можна підключити до хост-комп'ютера для виконання прискореної ML-обробки.

Вбудований Edge TPU - це невелика ASIC, розроблена Google, яка прискорює моделі TensorFlow Lite в енергоефективний він здатний виконувати 4 трильйони операцій в секунду (4 TOPS), використовуючи 2 Вт енергії - тобто 2 TOPS на ват. Наприклад, один Edge TPU може виконувати найсучасніші моделі мобільного зору, такі як MobileNet v2, зі швидкістю майже 400 кадрів в секунду. Така обробка ML на пристрої зменшує затримку, підвищує конфіденційність даних і усуває необхідність постійного підключення до Інтернету. Прискорювач зображений на рисунку 4.17. Для максимальної ефективності, прискорювач необхідно приєднати до роз'єму USB 3 на платі Raspberry pi 5. Необхідні порти USB зображені на рисунку 4.18.

Після під'єднання прискорювача до плати, необхідно його налаштувати. Для початку, потрібно встановити оптимальну версію Python, а саме 3.9.12. Щоб це зробити, нам слід виконати наступні кроки:

1. Завантажити усі оновлення та встановити їх на плату. Використовуючи команди `sudo apt-get update` та `sudo apt-get upgrade`. Виконання цих команд може зайняти тривалий час та після завершення необхідно перезавантажити пристрій для встановлення усіх оновлень.
2. Встановити віртуальне середовище Python. Шляхом виконання команди `curl https://pyenv.run | bash` ми завантажуюмо останню версію середовища для подальшої роботи.
3. Налаштовуємо конфігурацію для встановлення Python.

4. Встановлення залежностей. За допомогою команди `sudo apt-get install -yes libssl-dev zlib1g-dev libbz2-dev libreadline-dev libsqlite3-dev llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev libgdbm-dev lzma lzma-dev tcl-dev libxml2-dev libxmlsec1-dev libffi-dev liblzma-dev wget curl make build-essential openssl` ми довантажуюмо усі необхідні бібліотеки.



Рисунок 4.17 – Зовнішній вигляд Coral USB Accelerator

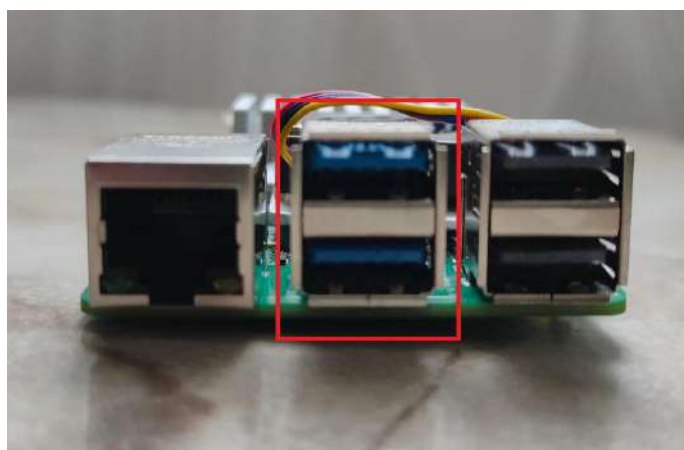


Рисунок 4.18 – Необхідні порти для прискорювача

5. Встановлення версії Python. У цьому дослідженні було встановлено версію 3.9.12, але згідно документації, версія може бути 3.6-3.9.

6. Встановлюємо версію за замовчуванням. Для зручності роботи було встановлено версію 3.9.12 стандартною для всієї системи.

Після завантаження версії Python, виконуємо команду `pip install edge-tpu-`

silva. Ця команда перевіряє налаштування пристрою та прискорювача та довантажує необхідні пакети для подальшої роботи.

Останнім кроком є виконання команди `silvatpu-linux-setup`, яка дозволить використовувати прискорювач разом з пристроєм. Після завершення виконання команди необхідно від'єднати та заново приєднати прискорювач до плати для завершення усіх налаштувань.

Передбачення емоцій з використанням прискорювача зображено на рисунках 4.19 – 4.22.

Після тестування моделі з прискорювачем, було порівняно результати. Прискорювач зменшив навантаження на систему та покращив затримку між розпізнаваннями емоцій у реальному часі. Детальніше про покращення роботи з прискорювачем зображено на рисунках 4.23 – 4.25.

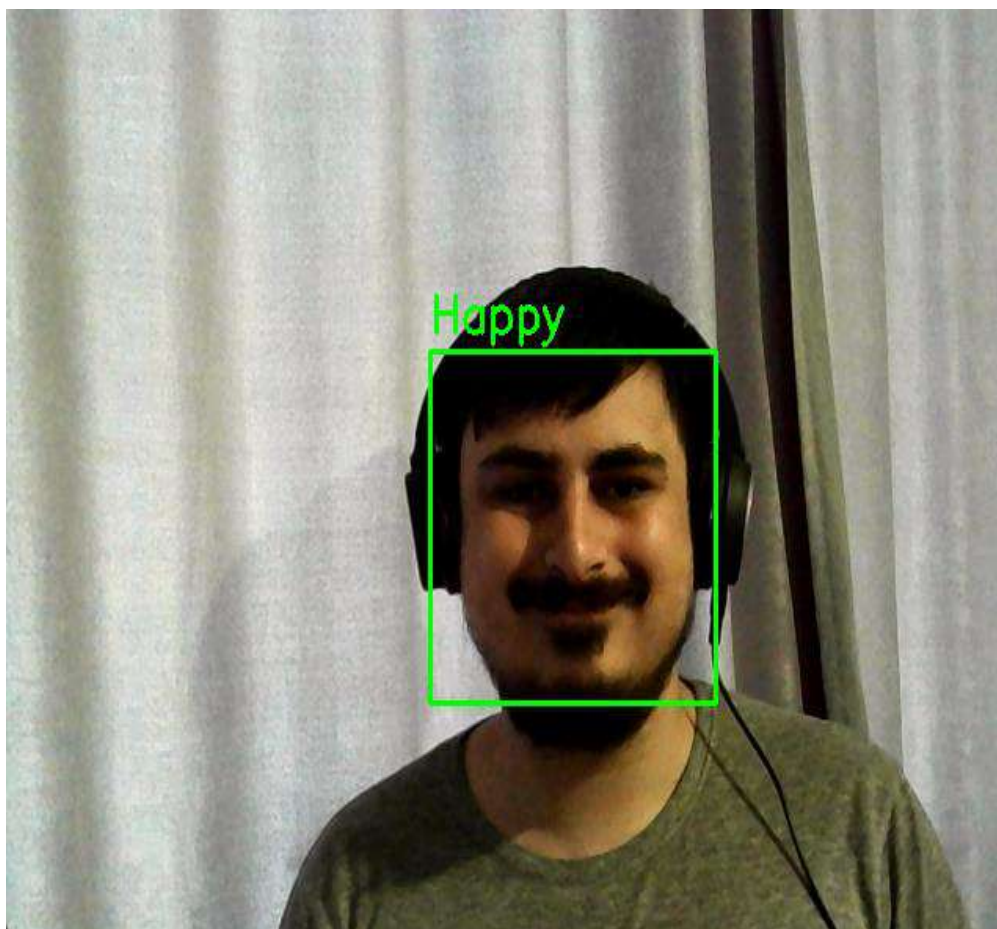


Рисунок 4.19 – Передбачення емоції з прискорювачем (Harry)

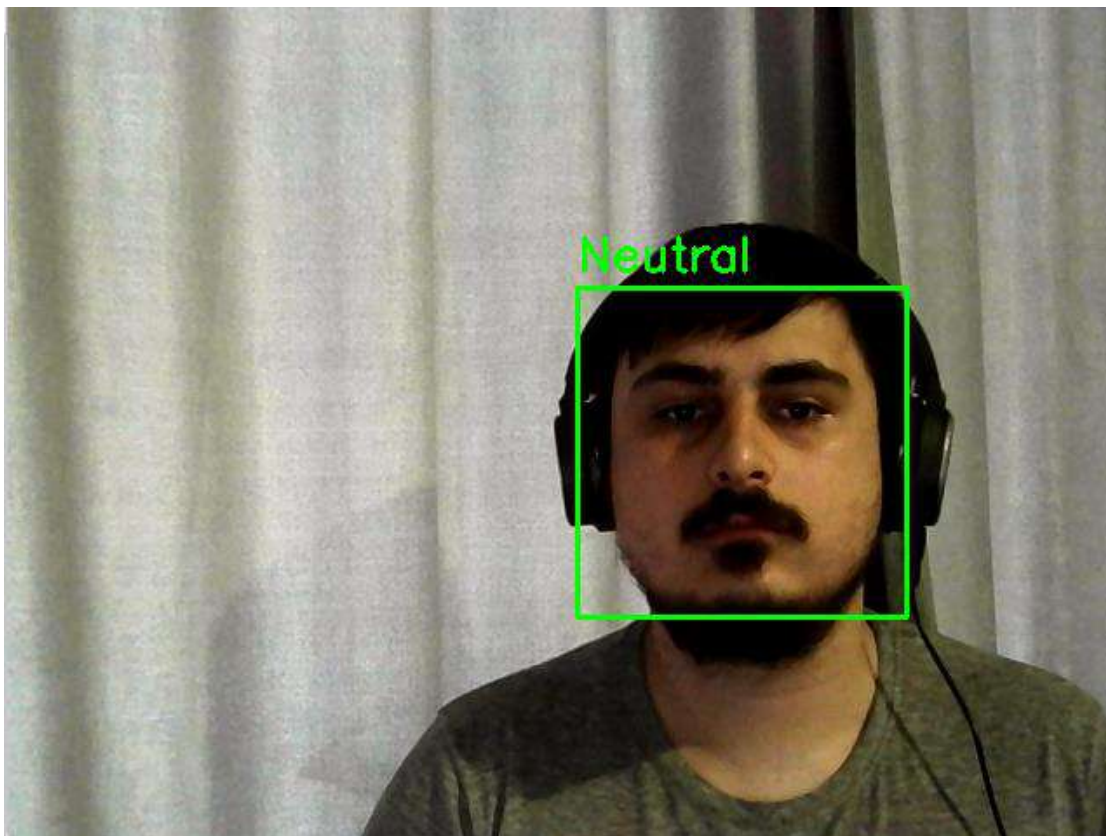


Рисунок 4.20 – Передбачення емоції з прискорювачем (Neutral)

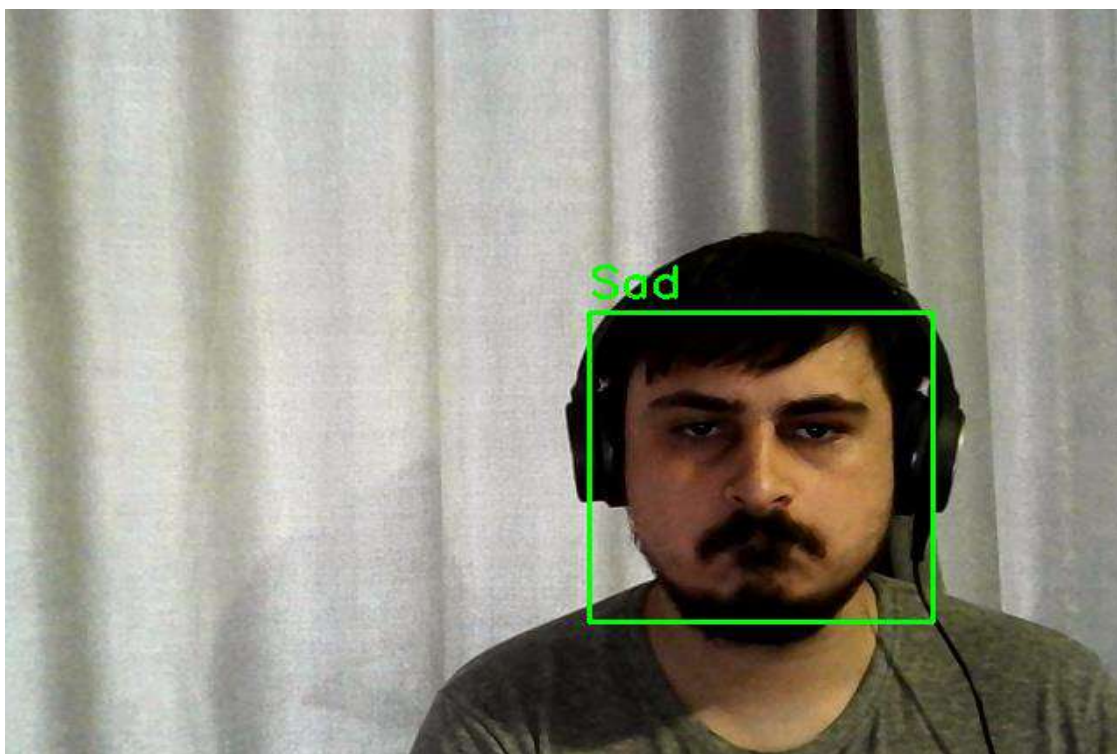


Рисунок 4.21 – Передбачення емоції з прискорювачем (Sad)

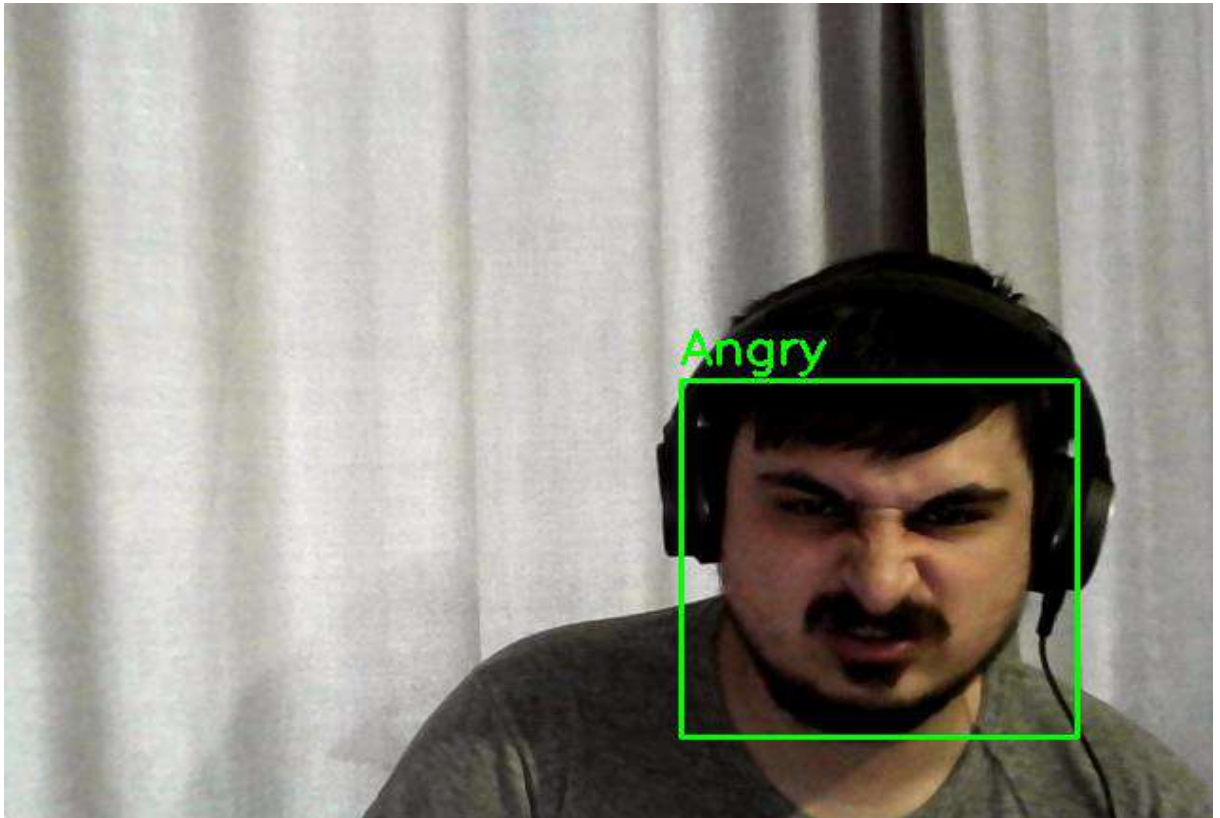


Рисунок 4.22 – Передбачення емоції з прискорювачем (Angry)

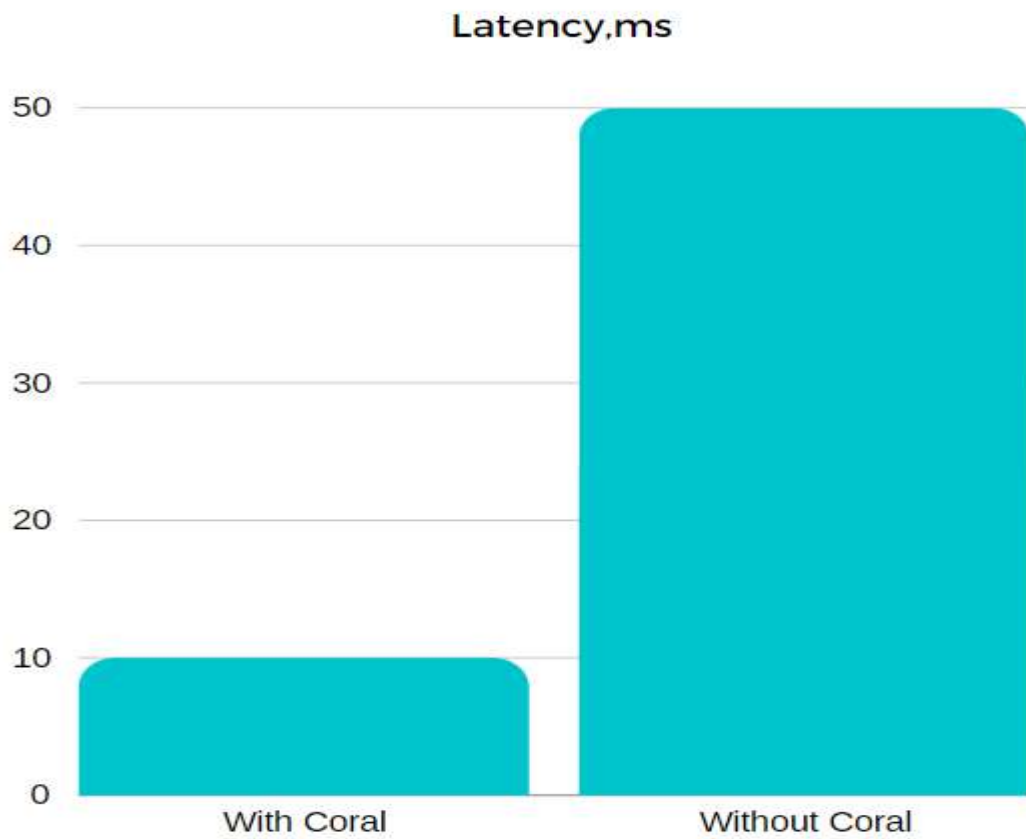


Рисунок 4.23 – Затримка у ms між розпізнаванням емоцій

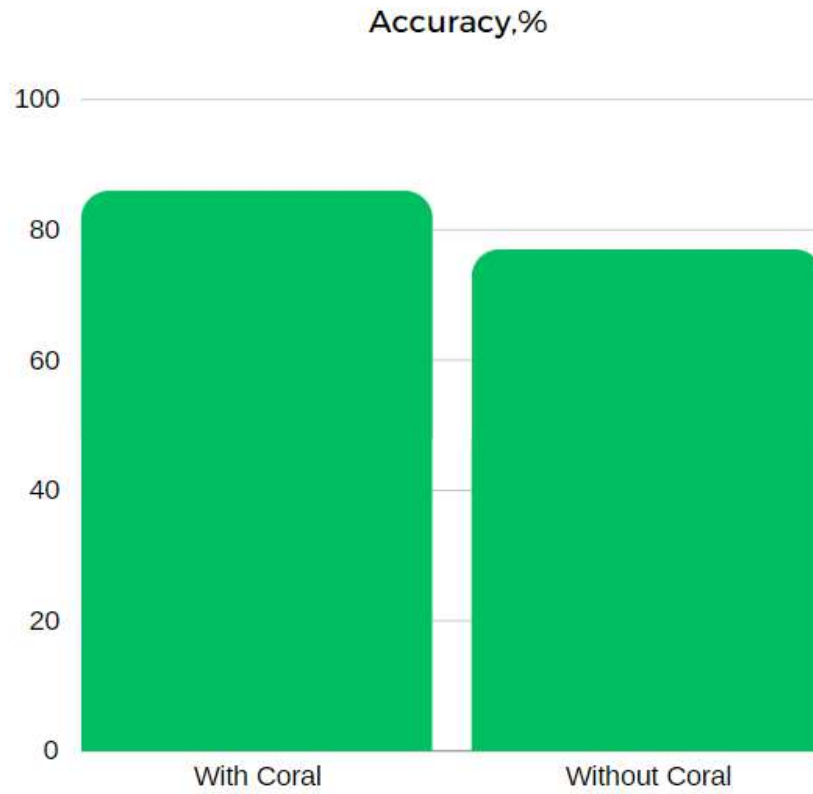


Рисунок 4.24 – Відсоток правильних передбачень емоцій

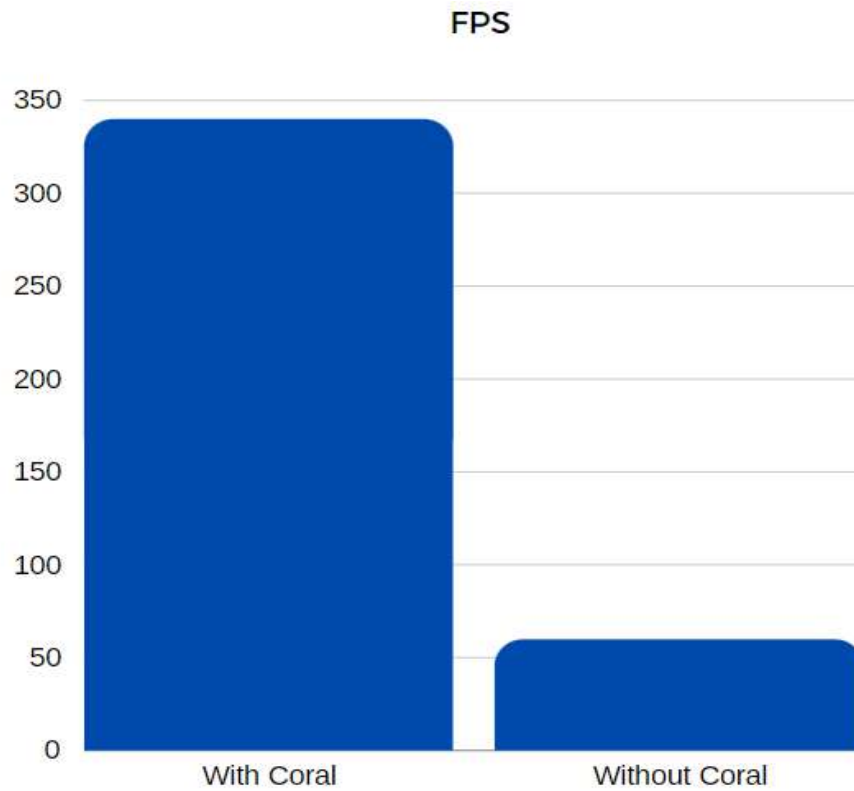


Рисунок 4.25 – Кількість кадрів у секунду під час розпізнавання

#### 4.4 Висновки до четвертого розділу

У цьому розділі було розглянуто та проаналізовано результати тренування моделі, а також можливість покращення результатів моделі шляхом приєднання прискорювача до плати.

В процесі аналізу було створено метрики, що показують продуктивність моделі та її характеристики.

Також, було перевірено результати у реальному часі шляхом підключення камери та налаштування її для роботи з розпізнаванням. У процесі розпізнавання у реальному часі було порівняно роботу моделі з використанням прискорювача та без, та, відповідно, сформовано графіки продуктивності.

## ВИСНОВКИ

У роботі внаслідок виконаних практичних та теоретичних досліджень розроблено метод розпізнавання емоцій у реальному часі за допомогою комп'ютерного зору, який оптимально працює на мобільних пристроях/пристроях з обмеженою продуктивністю.

У першому розділі було розглянуто існуючі рішення у методах розпізнавання емоцій. Було досліджено сучасні методи розпізнавання емоцій з використанням машинного навчання та комп'ютерного зору. Крім того, було проаналізовано найсучасніші тенденції у розвитку методів з використанням штучного інтелекту.

У другому розділі було проаналізовано методи корекції помилок під час розпізнавання емоцій. Також, було більш детально порівняно наявні сучасні моделі, які підходять для цього дослідження та аналіз їх переваг та недоліків у роботі. Було розглянуто усі необхідні компоненти для коректної роботи моделі та підготовка усіх бібліотек для розпізнавання емоцій. Також, було досліджено датасет, який було використано у цьому дослідженні.

У третьому розділі було розроблено та представлено метод та алгоритм тренування нейронної мережі для розпізнавання емоцій у реальному часі. Під час роботи над цим розділом було додатково створено власний датасет з використання персональних фото для покращення методів навчання моделі. Також, було розглянуто усі необхідні бібліотеки для правильної роботи моделі.

У четвертому розділі було проаналізовано отримані у третьому розділі результати шляхом перевірки моделі у точності розпізнавання емоцій з датасету та з використанням комп'ютерного зору. Також, було досліджено спосіб покращення результатів моделі шляхом підключення прискорювача та аналіз отриманих результатів шляхом побудови графів.

За темою дипломної роботи була опубліковані статті:

– у збірнику “Вчені записки” Таврійського національного університету імені В.І. Вернадського, Том 35 (74) №3 Частина 1 (Київ – 2024 – С. 77-85);

- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 15-21);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 37-44).

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Getting started with Raspberry pi 5. URL: <https://www.raspberrypi.com/documentation/computers/getting-started.html> (дата звернення: 10.11.2024).
2. Software for Raspberry pi. URL: <https://www.raspberrypi.com/software/> (дата звернення: 12.11.2024).
3. Raspberry pi Software. URL: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html> (дата звернення: 12.11.2024).
4. Processors for Raspberry pi. URL: <https://www.raspberrypi.com/documentation/computers/processors.html> (дата звернення: 17.11.2024).
5. Configuration for Raspberry pi. URL: [https://www.raspberrypi.com/documentation/computers/config\\_txt.html](https://www.raspberrypi.com/documentation/computers/config_txt.html) (дата звернення: 15.11.2024).
6. Visual representation of Raspberry pi 5. URL: <https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-mechanical-drawing.pdf> (дата звернення: 18.11.2024).
7. Brief information about Raspberry pi 5. URL: [raspberrypi-5-product-brief.pdf \(raspberrypi.com\)](https://datasheets.raspberrypi.com/rpi5/raspberry-pi-5-product-brief.pdf) (дата звернення: 12.11.2024).
8. Case fan for Raspberry pi 5. URL: <https://datasheets.raspberrypi.com/case-fan/case-fan-product-brief.pdf> (дата звернення: 17.11.2024).
9. Debugging process on Raspberry pi. URL: <https://datasheets.raspberrypi.com/debug/raspberry-pi-debug-probe-mechanical-drawing.pdf> (дата звернення: 17.11.2024).
10. Power supply for Raspberry pi 5. URL: <https://datasheets.raspberrypi.com/power-supply/27w-usb-c-power-supply-product-brief.pdf> (дата звернення: 17.11.2024).

11. Огляд Raspberry pi 5. URL: <https://minicomp.com.ua/ua/raspberry-pi-5-v-ukraini> (дата звернення: 18.11.2024).
12. Raspberry pi Camera. URL: <https://www.raspberrypi.com/documentation/accessories/camera.html> (дата звернення: 20.11.2024).
13. Everything about pi 5. URL: <https://www.pcmag.com/reviews/raspberry-pi-5> (дата звернення: 22.11.2024).
14. Get started with Pi Camera Module. URL: <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera> (дата звернення: 22.11.2024).
15. Raspberry Pi OS. URL: <https://www.raspberrypi.com/software/> (дата звернення: 23.11.2024).
16. Introducing to Tensorflow. URL: <https://www.tensorflow.org/learn> (дата звернення: 05.12.2024).
17. Tutorials on Tensorflow. URL: <https://www.tensorflow.org/tutorials> (дата звернення: 05.12.2024).
18. Basic guide on Tensorflow. URL: <https://www.tensorflow.org/guide> (дата звернення: 10.12.2024).
19. How to install Tensorflow. URL: <https://www.tensorflow.org/install> (дата звернення: 10.12.2024).
20. General information about Tensorflow. URL: <https://www.tensorflow.org/about> (дата звернення: 12.12.2024).
21. Get started with Keras. URL: [https://keras.io/getting\\_started/](https://keras.io/getting_started/) (дата звернення: 12.12.2024).
22. Keras API Documentation. URL: <https://keras.io/api/> (дата звернення: 12.12.2024).
23. Guides on Keras. URL: <https://keras.io/guides/> (дата звернення: 15.12.2024).
24. Examples with Keras. URL: <https://keras.io/examples/> (дата звернення: 15.12.2024).

25. About Coral USB Accelerator. URL: <https://coral.ai/products/accelerator> (дата звернення: 20.12.2024).
26. Examples using Coral. URL: <https://coral.ai/examples/> (дата звернення: 20.12.2024).
27. Getting Started with USB Accelerator. URL: <https://coral.ai/docs/accelerator/get-started> (дата звернення: 20.12.2024).
28. Everything about Coral. URL: <https://picockpit.com/raspberry-pi/everything-about-the-google-coral-usb-accelerator/> (дата звернення: 20.12.2024).
29. Harnessing the power of AI with Coral. URL: <https://medium.com/@fernandosrinesh/harnessing-the-power-of-ai-with-the-coral-usb-tpu-a-compact-accelerator-for-edge-computing-57610fcb0a86> (дата звернення: 20.12.2024).
30. Setup Coral on Raspberry pi. URL: <https://www.diyengineers.com/2024/05/18/setup-coral-tpu-usb-accelerator-on-raspberry-pi-5/> (дата звернення: 20.12.2024).
31. Filus K., Domańska J. Software vulnerabilities in TensorFlow-based deep learning applications. *Computers & Security*. 2023. №124. P. 102948–102955.
32. Truong T. X. A new approach based on tensorflow deep neural networks with adam optimizer and gis for spatial prediction of forest fire danger in tropical areas. *Remote Sensing*. 2023. №15(14). P. 3458–3473.
33. Picheny V. Efficiently exploring the depths of black-box functions with TensorFlow. *arXiv preprint arXiv:2302.08436*. 2023. №1. P. 15–20.
34. Contoli C.; Lattanzi E. A study on the application of tensorflow compression techniques to human activity recognition. *IEEE Access*. 2023. №11. P. 48046–48058.
35. Wang H. A novel multi-layer stacking ensemble wind power prediction model under Tensorflow deep learning framework considering feature enhancement and data hierarchy processing. *Energy*. 2024. №286. P. 129409–129415.
36. Niemczynowicz A., Kycia R. KHNNs: hypercomplex neural networks computations via Keras using TensorFlow and PyTorch. *arXiv preprint arXiv:2407.00452*. 2024. №1. P. 15–20.

37. Parra D., Escobar S., Camargo C. A Methodology and Open-Source Tools to Implement Convolutional Neural Networks Quantized with TensorFlow Lite on FPGAs. *Electronics*. 2023. №12(20). P. 4367–4390.
38. Xing L. Deep autoencoder for real-time single-channel EEG cleaning and its smartphone implementation using TensorFlow Lite with hardware/software acceleration. *IEEE Transactions on Biomedical Engineering*. 2024. №1. P. 15–20.
39. Alslaity A. Machine learning techniques for emotion detection and sentiment analysis: current state, challenges, and future directions. *Behaviour & Information Technology*. 2024. №43(1). P. 139–164.
40. Nie W. Long dialogue emotion detection based on commonsense knowledge graph guidance. *IEEE Transactions on Multimedia*. 2023. №26. P. 514–528.
41. Hung L., Alias S. Beyond sentiment analysis: A review of recent trends in text based sentiment analysis and emotion detection. *Journal of Advanced Computational Intelligence and Intelligent Informatics*. 2023. №27(1). P. 84–95.
42. Mamieva D. Multimodal emotion detection via attention-based fusion of extracted facial and speech features. *Sensors*. 2023. №23(12). P. 5475–5490.
43. Anzum F. Emotion detection from micro-blogs using novel input representation. *Ieee Access*. 2023. №11. P. 19512–19522.
44. Gupta D. Image segmentation keras: Implementation of segnet, fcn, unet, pspnet and other models in keras. *arXiv preprint arXiv:2307.13215*. 2023. №1. P. 25–40.
45. Lafta N., Abbood Z. Comprehensive Review and Comparative Analysis of Keras for Deep Learning Applications: A Survey on Face Detection Using Convolutional Neural Networks. *International Journal of Religion*. 2024. №5(11). P. 1203–1213.
46. Kurniawan W. Comparison of the keras-lstm algorithms for classifying autism spectrum disorder using facial images. *2023 International Conference on Data Science and Its Applications (ICoDSA)*. 2023. №41. P. 7–12.
47. Fischer R., Buschjäger S. Stress-Testing USB Accelerators for Efficient Edge Inference. *2024 IEEE*. 2024. №21. P. 150–161.

48. Reidy B. Work in progress: real-time transformer inference on edge AI accelerators: *2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. 2023. №2. P. 341–344.
49. Wess M. Energy Profiling of DNN Accelerators. *2023 26th Euromicro Conference on Digital System Design (DSD)*. 2023. P. 53–60.
50. Fojtik R. Using hardware to support artificial intelligence in education. *INTED2024 Proceedings*. 2024. P. 4519–4524.
51. Karthikeyan, S. A systematic analysis on raspberry pi prototyping: Uses, challenges, benefits, and drawbacks. *IEEE Internet of Things Journal*. 2023. №10(16). P. 14397–14417.
52. Sariyer R., Edwards A. Open hardware for microfluidics: exploiting raspberry Pi singleboard computer and camera systems for customisable laboratory instrumentation. *Biosensors*. 2023. №13(10). P. 948–961.
53. Coppini S. Experiments on real-life emotions challenge Ekman's model. *Scientific reports*. 2023. №13(1). P. 9511–9523.
54. Емоційний інтелект. URL: [https://stud.com.ua/17625/psihologiya/diagnostika\\_emotsiynogo\\_intelektu\\_holl](https://stud.com.ua/17625/psihologiya/diagnostika_emotsiynogo_intelektu_holl) (дата звернення: 20.01.2025).
55. ISRE 2024. URL: <https://www.isre2024.org/> (дата звернення: 20.01.2025).
56. CERE 2023. URL: <https://www.isre.org/mpage/cere> (дата звернення: 20.01.2025).
57. Emotions 2023 Conference. URL: <https://www.tilburguniversity.edu/current/news/more-news/emotions-2023-conference> (дата звернення: 22.01.2025).
58. Police robots. URL: <https://builtin.com/robotics/police-robot-law-enforcement> (дата звернення: 22.01.2025).
59. Everything about robot waiters. URL: <https://backofhouse.io/resources/how-do-robotic-waiters-work-and-are-they-right-for-your-restaurant> (дата звернення: 22.01.2025).

60. Tensorflow Lite Guide. URL: <https://www.tensorflow.org/lite> (дата звернення: 23.01.2025).
61. Python Documentation. URL: <https://www.python.org/> (дата звернення: 23.01.2025).
62. Steps of object detection. URL: [https://www.researchgate.net/figure/Basic-implementation-steps-of-object-detection-and-classification\\_fig2\\_358255344](https://www.researchgate.net/figure/Basic-implementation-steps-of-object-detection-and-classification_fig2_358255344) (дата звернення: 25.01.2025).
63. AlexNet Explained. URL: <https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5> (дата звернення: 25.01.2025).
64. VGGNet Architecture. URL: <https://medium.com/analytics-vidhya/vggnet-architecture-explained-e5c7318aa5b6> (дата звернення: 03.02.2025).
65. ResNet Deep Learning. URL: <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/> (дата звернення: 03.02.2025).
66. Everything about EfficientNet. URL: <https://paperswithcode.com/method/efficientnet> (дата звернення: 03.02.2025).
67. MobileNet Review. URL: <https://keras.io/api/applications/mobilenet/>
68. SqueezeNet Explained. URL: <https://paperswithcode.com/method/squeezenet> (дата звернення: 03.02.2025).
69. DenseNet Explained. URL: <https://paperswithcode.com/method/densenet> (дата звернення: 03.02.2025).
70. FeR2013 Dataset. URL: <https://paperswithcode.com/dataset/fer2013> (дата звернення: 03.02.2025).
71. YOLOv8 Documentation. URL: <https://docs.ultralytics.com/> (дата звернення: 03.02.2025).
72. Everything about yolov9. URL: <https://docs.ultralytics.com/models/yolov9/>
73. Yolov10 Documentation. URL: <https://docs.ultralytics.com/models/yolov10/> (дата звернення: 03.02.2025).
74. OpenCV about page. URL: <https://opencv.org/> (дата звернення: 03.02.2025).

75. Pillow (PIL Fork). URL: <https://pypi.org/project/pillow/> (дата звернення: 05.02.2025).
76. Scikit Guide. URL: <https://scikit-learn.org/stable/index.html> (дата звернення: 05.02.2025).
77. PyTorch Documentation. URL: <https://pytorch.org/> (дата звернення: 05.02.2025).
78. Dini P. Overview of AI-Models and Tools in Embedded IoT Applications. *Electronics*. 2024. №13(12). P. 2322–2330.
79. Roboflow Main page. URL: <https://roboflow.com/> (дата звернення: 07.02.2025).
80. Ісаєв Т. С., Кисіль Т.М. Метод розпізнавання емоцій людини у реальному часі за допомогою машинного зору. *Вчені записки*. 2024. Том 35 (74) № 3. С. 77–85.
81. Isaiev T., Kysil T. Method of creating custom dataset to train convolutional neural network. *Computer systems and information technologies*. 2024. № 4. P. 37–44.
82. Isaiev T., Kysil T. Method for improving the performance of convolutional neural networks using an accelerator. *Computer systems and information technologies*. 2024. № 4. P. 15–21.

**ДОДАТОК А**  
(обов'язковий)

**ЛІСТИНГ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РОЗПІЗНАВАННЯ ЕМОЦІЙ  
ЛЮДИНИ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ КОМП'ЮТЕРНОГО  
ЗОРУ**

Лістинг файлу тренування моделі нейронної мережі Tensorflow

```
from keras.utils import to_categorical
from keras_preprocessing.image import load_img
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten,
MaxPooling2D
import os
import pandas as pd
import numpy as np

TRAIN_DIR = "C:\\Users\\timha\\Mine\\Diplomna\\Images\\train"
TEST_DIR = "C:\\Users\\timha\\Mine\\Diplomna\\Images\\test"

def createdataframe(dir):
    image_paths = []
    labels = []
    for label in os.listdir(dir):
        for imagename in os.listdir(os.path.join(dir, label)):
            image_paths.append(os.path.join(dir, label, imagename))
            labels.append(label)
        print(label, "completed")
    return image_paths, labels

train = pd.DataFrame()
```

```
train['image'], train['label'] = createdataframe(TRAIN_DIR)

print(train)

test = pd.DataFrame()
test['image'], test['label'] = createdataframe(TEST_DIR)

print(test)
print(test['image'])

from tqdm.notebook import tqdm

def extract_features(images):
    features = []
    for image in tqdm(images):
        img = load_img(image, grayscale = True )
        img = np.array(img)
        features.append(img)
    features = np.array(features)
    features = features.reshape(len(features), 48, 48, 1)
    return features

train_features = extract_features(train['image'])

test_features = extract_features(test['image'])

x_train = train_features/255.0
x_test = test_features/255.0

from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
le.fit(train['label'])

y_train = le.transform(train['label'])
y_test = le.transform(test['label'])

y_train = to_categorical(y_train,num_classes = 7)
y_test = to_categorical(y_test,num_classes = 7)

model = Sequential()
# convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu',
input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
# fully connected layers
```

```

model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',
metrics = 'accuracy' )

model.fit(x= x_train,y = y_train, batch_size = 128, epochs = 100,
validation_data = (x_test,y_test))

model_json = model.to_json()
with open("emotiondetector.json",'w') as json_file:
    json_file.write(model_json)
model.save("emotiondetector.h5")

from keras.models import model_from_json

json_file = open("facialemotionmodel.json", "r")
model_json = json_file.read()
json_file.close()
model = model_from_json(model_json)
model.load_weights("facialemotionmodel.h5")

```

### Лістинг файлу перевірки моделі у реальному часі

```

import cv2

import numpy as np

import tensorflow as tf

```

```
import time

# Load TFLite model and allocate tensors.

interpreter = tf.lite.Interpreter(model_path="D:\\Emotion_Detect\\emotion_detection_model-master\\model.tflite")

interpreter.allocate_tensors()

# Get input and output tensors.

input_details = interpreter.get_input_details()

output_details = interpreter.get_output_details()

def process_frame(frame):

    # Initialize fps to avoid UnboundLocalError

    fps = 0

    # Resize frame to match model input size (64x64)

    resized_frame = cv2.resize(frame, (64, 64))

    # Convert frame to array

    img = np.array(resized_frame)

    # Expand dimensions to match model input shape (1, 64, 64, 3)

    img = np.expand_dims(img, axis=0)

    # Convert data type to float32

    img = img.astype('float32')

    # Normalize pixel values

    img /= 255.0

    # Set input tensor

    interpreter.set_tensor(input_details[0]['index'], img)

    # Run inference

    start_time = time.time() # Початковий час для вимірювання часу інференсу
```

```
interpreter.invoke()

end_time = time.time() # Кінцевий час для вимірювання часу інференсу

# Визначення FPS

time_difference = end_time - start_time

if time_difference != 0:

    fps = 1.0 / time_difference

return interpreter.get_tensor(output_details[0]['index']), fps

def main():

    cap = cv2.VideoCapture(0)

    face_cascade = cv2.CascadeClassifier(cv2.data.harcascades + 'haarcascade_frontalface_default.xml')

    while True:

        ret, frame = cap.read()

        # Перевірка, чи вдалося отримати кадр

        if not ret:

            break

        # Ініціалізація fps перед обробкою кадру

        fps = 0

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

        faces = face_cascade.detectMultiScale(gray, 1.3, 5)

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

            face_img = frame[y:y+h, x:x+w]

            # Process each face

            out, current_fps = process_frame(face_img) # Отримання значення fps разом з out

            fps = current_fps # Оновлення значення fps

            emotion_label = 'Unknown'

            if np.argmax(out) == 0:

                emotion_label = 'Sad'
```

```
elif np.argmax(out) == 2:
    emotion_label = 'Angry'
elif np.argmax(out) == 3:
    emotion_label = 'Neutral'
elif np.argmax(out) == 4:
    emotion_label = 'Happy'
elif np.argmax(out) == 6:
    emotion_label = 'Fear'

# Display emotion label
cv2.putText(frame, f"Person {i+1}: {emotion_label}", (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0,
255, 0), 2)

# Draw rectangle around the face
cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

# Відображення FPS
cv2.putText(frame, f"FPS: {int(fps)}", (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)

cv2.imshow('Emotion Recognition', frame)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
cap.release()
cv2.destroyAllWindows()


if __name__ == '__main__':
    main()
```

ДОДАТОК Б  
(обов'язковий)

## Презентаційні матеріали

Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору

Дипломна робота  
Студента групи КІЗм 23-1 Ісаєва Тимура Сергійовича  
Керівник роботи: Кисіль Тетяна Миколаївна



### Мета, Об'єкт та Предмет Дослідження

2

**Метою** кваліфікаційної роботи магістра є створення методу розпізнавання емоцій людини у реальному часі, що працюватиме ефективно на мобільних пристроях.

**Об'єктом** дослідження є підвищення продуктивності роботи моделей штучної нейронної мережі на пристроях з обмеженою потужністю.

**Предметом** дослідження є застосування нейронної мережі для створення нового методу розпізнавання емоцій.

## Наукова Новизна Дослідження

3

Наукова новизна отриманих результатів полягає в удосконаленні методу оптимізації нейронних мереж для задач розпізнавання емоцій, що дозволяє підвищити їхню продуктивність та забезпечити ефективне функціонування на мобільних пристроях з обмеженими ресурсами.

## Задачі Дослідження

4

- створити власний набір даних для підвищення точності розпізнавання емоцій;
- розробити власні методи розпізнавання;
- натренувати нейронну мережу на власному наборі даних;
- покращити результати моделі внаслідок використання прискорювача.

## Актуальність Дослідження

5

Актуальність даної роботи полягає у розробці та імплементуванню методу розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору, який би використовував глибоку нейронну мережу та забезпечував високу швидкість та точність аналізу відеоданих на малопотужних пристроях.

## Публікації

6

За темою дипломної роботи була опубліковані статті:

- у збірнику “Вчені записки” Таврійського національного університету імені В.І. Вернадського, Том 35 (74) №3 Частина 1 (Київ – 2024 – С. 77-85);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 15-21);
- у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (Хмельницький – 2024 – С. 37-44).

## Вибір Платформи/Кінцевого пристрою

7



Raspberry Pi 5

Характеристика	Опис
Тип процесора	2.4GHz quad-core, 64-bit Arm Cortex-A76 CPU, з кешем L2 512KB на ядро та спільним кешем L3 2MB
Грфічне ядро	VideoCore VII
Кількість оперативної пам'яті та її вид	LPDDR4X-4267 SDRAM 8GB
Версія Wi-Fi	Двухдіапазонний 802.11ac
Версія Bluetooth	5.0 / Bluetooth Low Energy (BLE)
Кількість USB та їх види	2 x USB 3.0, 2 x USB 2.0
Швидкість Ethernet	Gigabit Ethernet
Інтерфейс для камери/дисплея	2 x 4-канальні MIPI
Тип PCIe	PCIe 2.0 x1 інтерфейс для швидких периферійних пристроїв
Живлення для плати	5V/5A DC через USB-C, з підтримкою Power Delivery
GPIO (порти виводу/вводу)	Стандартний 40-контактний роз'єм Raspberry Pi

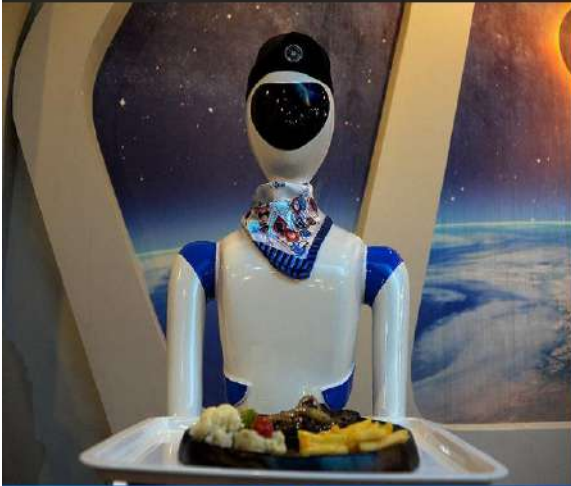
## Розділ 1

8

У першому розділі було розглянуто існуючі рішення у методах розпізнавання емоцій. Було досліджено сучасні методи розпізнавання емоцій з використанням машинного навчання та комп'ютерного зору. Крім того, було проаналізовано найсучасніші тенденції у розвитку методів з використанням штучного інтелекту.

## Розділ 1

9



Робот Офіціант



Робот Поліцейський

## Розділ 2

10

У другому розділі було проаналізовано методи корекції помилок під час розпізнавання емоцій. Також, було більш детально порівняно наявні сучасні моделі, які підходять для цього дослідження та аналіз їх плюсів та мінусів у роботі. Було розглянуто усі необхідні компоненти для коректної роботи моделі та підготовка усіх бібліотек для розпізнавання емоцій. Також, було досліджено датасет, який було використано у цьому дослідженні.

## Розділ 2

11



FeR2013

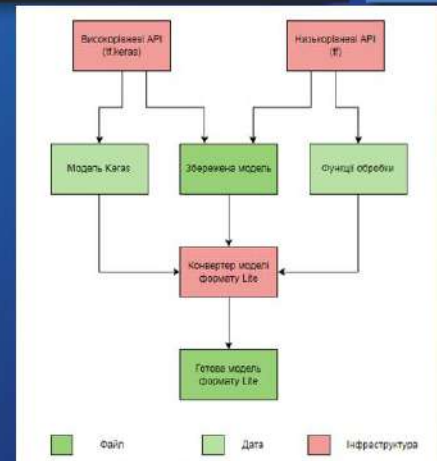
Кількість зображень: Набір даних містить приблизно 30,000 зображень обличч.  
 Розмір зображень: Кожне зображення має розмір 48×48 пікселів.  
 Типи виразів обличчя: Набір даних містить 7 типів виразів обличчя (0 = Злий, 1 = Відраза, 2 = Страх, 3 = Щасливий, 4 = Сумний, 5 = Здивований, 6 = Нейтральний)  
 Розподіл зображень: Вираз обличчя “Відраза” має мінімальну кількість зображень - 600, тоді як інші мітки мають майже 5,000 зразків кожна.

## Розділ 2

12



Процес тренування Tensorflow моделі



Процес перетворення Tensorflow у TF Lite

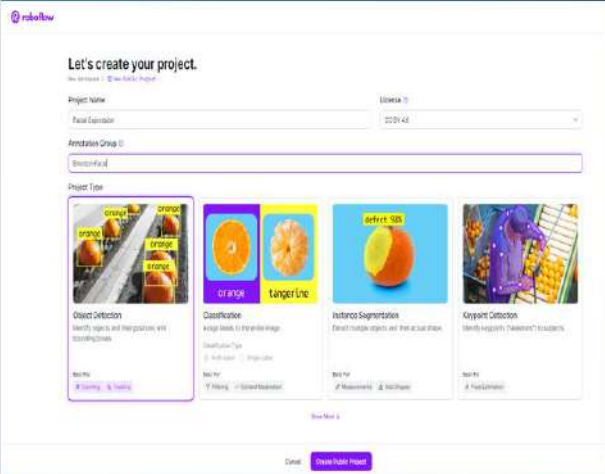
# Розділ 3

13

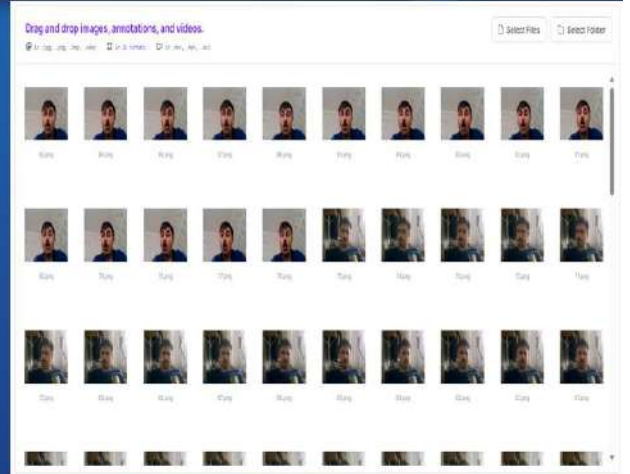
У третьому розділі було розроблено та представлено метод та алгоритм тренування нейронної мережі для розпізнавання емоцій у реальному часі. Під час роботи над цим розділом було додатково створено власний датасет з використання персональних фото для покращення методів навчання моделі. Також, було розглянуто усі необхідні бібліотеки для правильної роботи моделі.

# Розділ 3

14



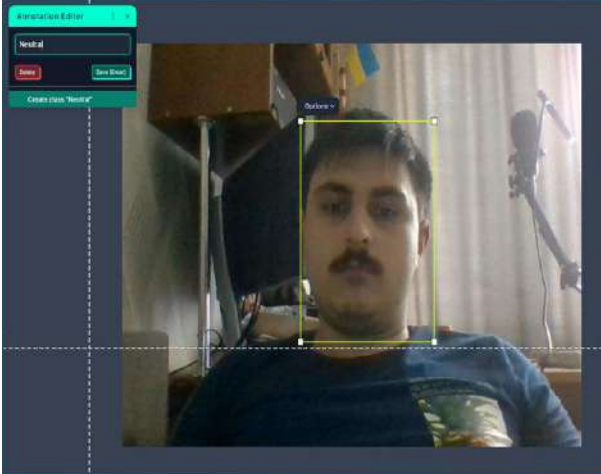
Процес створення власного набору даних



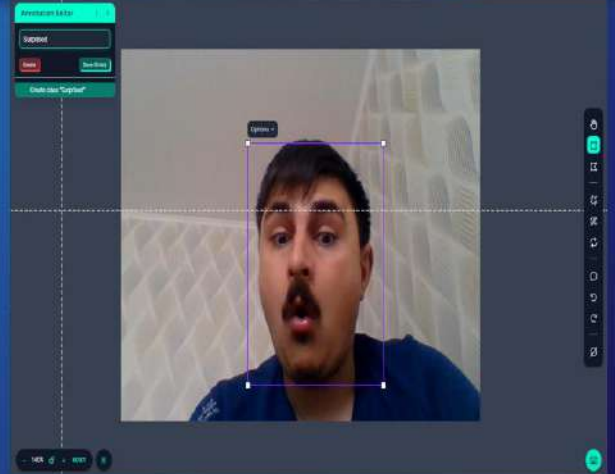
Завантаження зображень

# Розділ 3

15



Процес маркування фото (Neutral)



Процес маркування фото (Surprised)

# Розділ 3

16

```

model = Sequential()
# convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
# fully connected layers
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))

```

Налаштування моделі

```

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = 'accuracy')

model.fit(x=x_train,y=y_train, batch_size = 128, epochs = 100, validation_data = (x_test,y_test))

```

Налаштування для тренування моделі

```

model_json = model.to_json()
with open("emotiondetector.json", 'w') as json_file:
    json_file.write(model_json)
model.save("emotiondetector.h5")

```

```

from keras.models import model_from_json

```

```

json_file = open("facialemotionmodel.json", "r")
model_json = json_file.read()
json_file.close()
model = model_from_json(model_json)
model.load_weights("facialemotionmodel.h5")

```

Збереження моделі

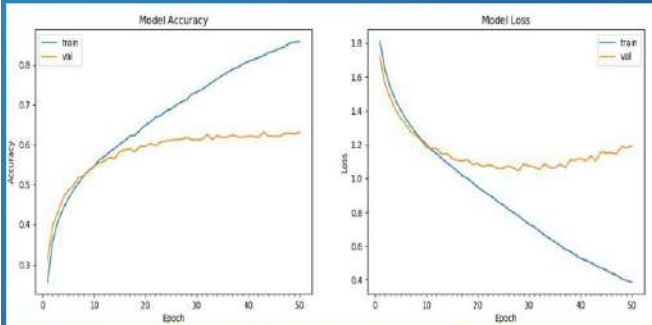
# Розділ 4

17

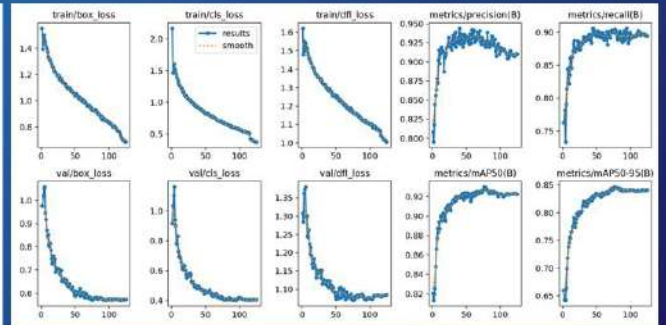
У четвертому розділі було проаналізовано отримані у третьому розділі результати шляхом перевірки моделі у точності розпізнавання емоцій з набору даних та з використанням комп'ютерного зору. Також, було досліджено спосіб покращення результатів моделі шляхом підключення прискорювача та аналіз отриманих результатів шляхом побудови графів.

# Розділ 4

18



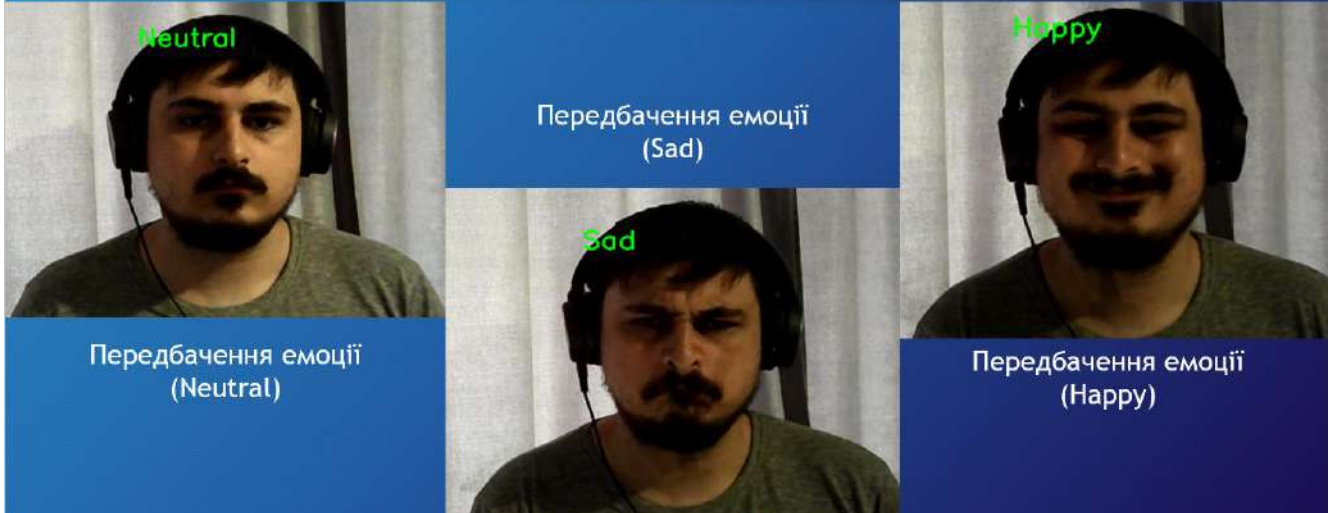
Точність моделі Tensorflow Lite



Набір діаграм з результатами тренування

## Розділ 4

19

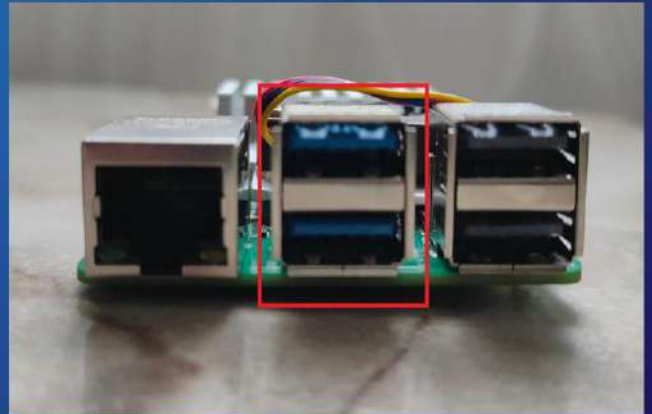


## Розділ 4

20



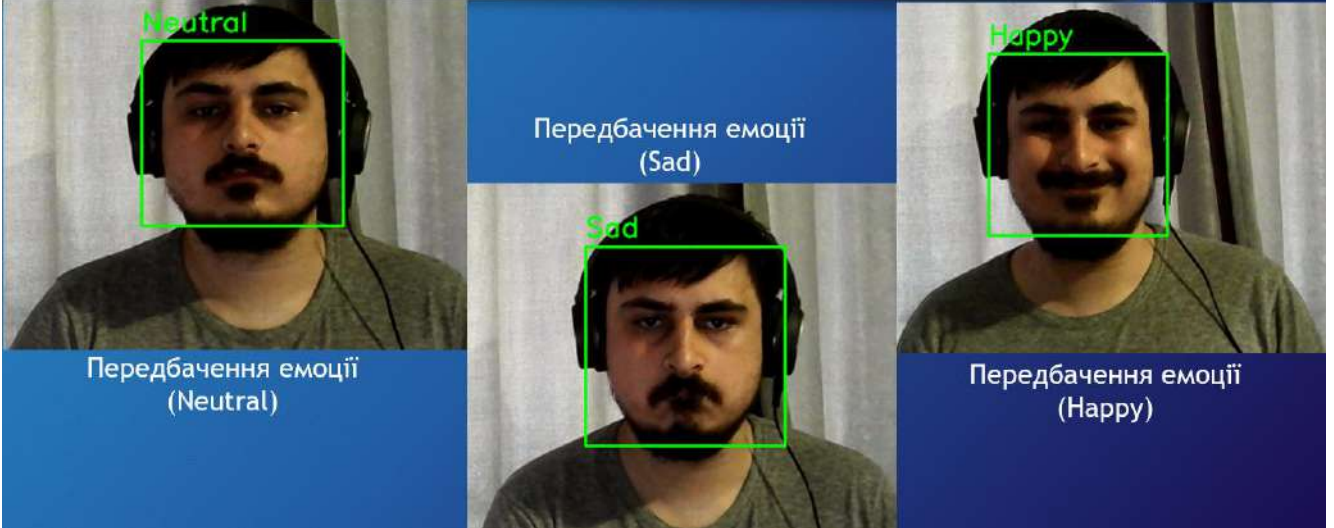
Google Coral USB Accelerator



Порти для прискорювача

# Розділ 4

21



# Розділ 4

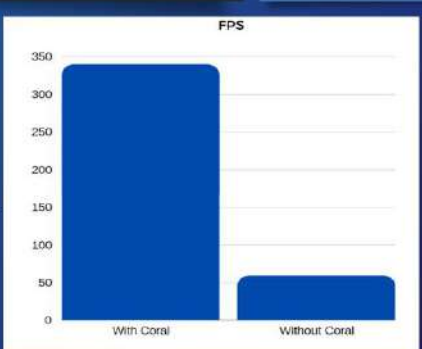
22



Точність моделі з прискорювачем та без



Затримка між розпізнаваннями з прискорювачем та без



Кількість кадрів у секунду з прискорювачем та без

## Висновок

23

У роботі внаслідок виконаних практичних та теоретичних досліджень розроблено метод розпізнавання емоцій у реальному часі за допомогою комп'ютерного зору, який оптимально працює на мобільних пристроях/пристроях з обмеженою продуктивністю.

24

Дякую за увагу!

## ДОДАТОК В (обов'язковий) КОПІ СТАТЕЙ

Стаття у збірнику “Вчені записки” Таврійського національного університету  
імені В.І. Вернадського, Том 35 (74) №3 Частина 1

Інформатика, обчислювальна техніка та автоматизація

УДК 004.932

*Ісачев Т.С.*

Хмельницький національний університет

*Кисіль Т.М.*

Хмельницький національний університет

### МЕТОД РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ МАШИННОГО ЗОРУ

*У наш час існують різні методи і різні ознаки, які можуть бути використані для розпізнавання обличчя і емоцій, таких як обличчя, текст, мова і так далі. Вираз обличчя серед них є найкращою. Виявлення емоцій також широко використовується у багатьох сферах, таких як розуміння поведінки людини, виявлення психічних розладів, виявлення емоцій людей у натовпі тощо. У запропонованій концепції розпізнавання емоцій обличчя можна розділити на три етапи. На першому етапі відбувається виявлення людського обличчя на зображенні з камери, потім на другому етапі вхідні дані аналізуються на основі ознак за допомогою моделі згорткових нейронних мереж (CNN). На останньому етапі відбувається класифікація людських емоцій на 7 основних категорій: страх, гнів, відраза, сум, здивування, нейтральний та радість. Таким чином, запропонована модель розділена на три основні цілі: виявлення обличчя, розпізнавання обличчя та класифікація емоцій. Розпізнавання обличчя має багато важливих застосувань у різних сферах, таких як верифікація особи, безпека, біометрія, смарт-картки, системи спостереження тощо. Міміка передає невербальні сигнали, які відіграють важливу роль у міжособистісних відносинах. Система Cognitive Emotion AI – це процес ідентифікації емоційного стану людини. Основною метою нашого дослідження є розробка надійної системи, яка може виявляти та розпізнавати людські емоції з прямих трансляцій. Методологія цієї системи базується на двох етапах – розпізнавання обличчя здійснюється шляхом вилучення каскаду Хаара та розпізнавання емоцій за допомогою алгоритму Віоли-Джонса, а потім відбувається перевірка та розпізнавання емоцій за допомогою методів штучного інтелекту. Система приймає зображення або кадр в якості вхідних даних, і шляхом аналізу зображення моделі, виконується попередня обробка та виділення рис обличчя, після чого система прогнозує емоційний стан. У даній роботі було зроблено згорткову нейронну мережу для класифікації людських емоцій за динамічною мімікою обличчя в реальному часі. Також, було проведено навчання нашої системи з використанням моделі Tensorflow, яка була навчена класифікувати людські емоції. Було використано відкриту базу даних із зображеннями для навчання моделі, а також власні фото для покращення результату. Для відображення результатів було підключено живий відеопотік. Отримані результати демонструють можливість застосування нейронних мереж в реальному часі для виявлення людських емоцій.*

**Ключові слова:** розпізнавання емоцій, аналітика, тренування моделі, реальний час, машинне навчання, штучний інтелект.

**Постановка проблеми.** Емоції часто опосередковують і полегшують взаємодію між людьми. Таким чином, розуміння емоцій часто вносить контекст у, здавалося б, химерну та/або складну соціальну комунікацію. Емоції можна розпізнати за допомогою цілої низки засобів, як-от: голос, інтонація, мова тіла та більш складними методами, такими як електроенцефалографія (ЕЕГ) [11]. Однак, більш простим і практичним методом є дослідження міміки. Існує сім типів людських емоцій, які, як було доведено, є універсальними для всіх людей [2]: гнів, відраза, страх, щастя, смуток, здивування та нейтральний вираз. Таким чином, утиліта, яка розпізнає емоції за виразом

обличчя, могла б знайти широке застосування. Завдання розпізнавання емоцій є особливо складним з двох причин: 1) не існує великої бази даних навчальних зображень і 2) класифікація емоцій може бути складною в залежності від того, чи є вхідне зображення статичним, чи зображення рухається. Більшість програм розпізнавання емоцій досліджують статичні зображення виразу обличчя. Дослідження застосування згорткових нейронних мереж (ЗНМ) для розпізнавання емоцій у реальному часі за допомогою вхідного відеопотоку є дуже важливими на сьогодні. Враховуючи обчислювальні вимоги та складність штучної моделі, оптимізація мережі для ефективних

обчислень для покадрової покадрової класифікації є необхідною. Крім того, необхідно враховувати варіації освітлення та положення об'єктів. Було розроблено систему для виявлення людських емоцій у різних сценах, ракурсах та умовах освітлення в режимі реального часу.

**Аналіз останніх досліджень і публікацій.** За останні два десятиліття дослідники значно просулися в розпізнаванні людських емоцій на обличчі за допомогою комп'ютерного зору. Історично склалося так, що існувало багато підходів до вирішення цієї проблеми, включаючи використання пірамідальних гістограм градієнтів (PHOG) [12], AU, що враховує риси обличчя, покращені LBP-дескриптори та RNN [13]. Однак нещодавні найкращі роботи, подані на конкурс «Емоції в дикій природі» 2020 року [18] використовували глибокі згорткові нейронні мережі (CNN), що дають до 62 % точності тесту.

Нещодавня розробка G. Levi et. al [14] показала значне покращення розпізнавання емоцій на обличчі за допомогою штучного інтелекту. Головною проблемою для дослідників є: 1) мала кількість даних, доступних для навчання глибоких нейронних мереж, та 2) варіації зовнішнього вигляду, які зазвичай спричинені варіаціями освітлення. Вони використовували локальні

бінарні патерни (LBP) для перетворення зображення в інваріантний до освітлення тривимірний простір, який може слугувати входними даними для CNN.

Ця спеціальна обробка даних була застосована до різних загальнодоступних модулів, таких як VGG S [15]. Фінальні результати показали точність тесту до 54,56 %. Існує багато підходів до навчання моделі, але вони не мають аналізу точності розпізнавання емоцій, що є великим недоліком тому, що аналіз точності моделі у реальному часі допоможе покращити саму модель та краще зрозуміти які фактори впливають на її роботу та що потрібно для отримання оптимального результату щоб використовувати модель більш досконало.

**Постановка завдання.** Метою роботи є створення моделі, що дозволить аналізувати емоційний стан людини у реальному часі.

**Виклад основного матеріалу.** Створення та дослідження моделі із застосуванням машинного та штучного інтелекту відрізняється від розробки додатків, програм та ігор у тому, що процес моделювання включає постійні дослідження та оновлення даних, здійснюючи експерименти, щоб навчити нейронну мережу розрізняти об'єкти та емоції людини. У машинного навчання власний життєвий цикл, що означає проходження даних через певні етапи розробки та впровадження системи прогнозування. В порівнянні з циклом розробки програмного забезпечення, створення моделей машинного навчання включає етап експериментування з наборами даних для досягнення поставлених цілей і завдань, при цьому використовуються нові дані після проходження етапу навчання.

#### Початок роботи із моделлю та її тренування

На рис. 1 показано структуру запропонованої в цьому дослідженні системи. Як показано на рис. 1, було використано два підходи для виявлення емоцій суб'єкта: виявлення емоцій за допомогою мімічних орієнтирів та виявлення емоцій за допомогою сигналів ЕЕГ [11].

Для тренування моделі необхідно було обрати відкриту базу даних, яка містить картинки людей з різними емоціями та надасть змогу натренувати систему досить гарно. Було обрано базу даних Fer2013 [16]. Ця база даних знаходиться у відкритому доступі, а отже кожен бажаючий без проблем зможе використати картинку без ризику. Вона містить досить багато картинок людей різного віку, які показують різні емоції. Саму систему баз даних із картинками зображено на рис. 2.

У цьому дослідженні було використано HD-камеру для зйомки обличчя суб'єктів та створення

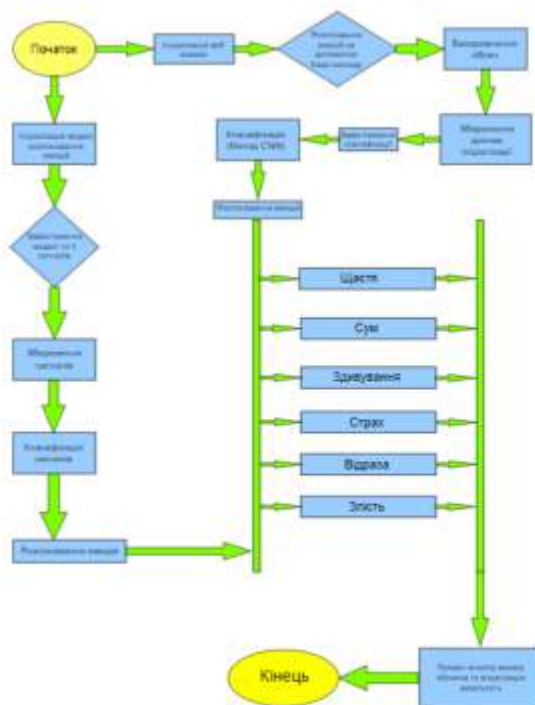


Рис. 1. Процес ініціалізації програми

зображення у відтінках сірого (додатково до наявних зображень у системі Fer2013). Це спростило процес розпізнавання виразу обличчя при розпізнаванні міміки. Потім, використовуючи зображення у відтінках сірого, визначаються очі суб'єкта, і десять віртуальних маркерів (одиниць дії) розміщуються на обличчі суб'єкта у визначених місцях за допомогою математичної моделі.

Алгоритм читання емоцій із зображення ґрунтується на алгоритмі їхнього розпізнавання, який використовується, наприклад, у FaceID. Для вирішення цих завдань із зображень витягуються спеціальні дескриптори обличчя, або вектори ознак. Спочатку мережа глибокого навчання тренується розпізнавати риси обличчя, роблячи це за допомогою карти точок навколо основних його частин (очі, брови, ніс, губи). Відстань між цими точками і є вектор ознак. Подібно до відбитка пальця, вектор ознак буде індивідуальний для кожної людини. Хоча риси обличчя й індивідуальні, вираз одних і тих самих емоцій однаковий у більшості людей. Для їхнього розпізнавання використовують групи точок із вектора ознак. Так, наприклад, підйом групи точок біля куточків рота свідчить про посмішку, зведення групи точок на бровах – про насупленість тощо. Процес розпізнавання зображень на рис. 3.

Алгоритм оптичного потоку Лукаса-Канде використовується для передачі положення кожного віртуального маркера, щоб відстежити його положення під час емоційного вираження суб'єкта. Десять ознак виводяться як відстань між кожним маркером і точкою. У поточному дослідженні всі дані про відстані обчислюються за допомогою теореми Піфагора. Потім вони зберігаються у форматі CSV під час процесу збору даних для подальшої обробки.

Основна функція цього методу – це дослідження відстаней між очима та носом, носом та ротом тощо. Це дає змогу зрозуміти почуття людини внаслідок дослідження її міміки та особливостей виразу обличчя. Звісно, у кожної людини свій власний вираз обличчя, але за допомогою цього методу можна дослідити різні емоції та навчити модель розпізнавати усі ознаки міміки та виразів обличчя людини. Цей процес є досить складним, але внаслідок розвитку технологій, цю систему можна досить



Рис. 2. Фрагмент системи баз даних FER2013[16]



Рис. 3. Процес розпізнавання емоцій [2]

легко імплементувати за допомогою штучного інтелекту та машинного навчання. Увесь процес розпізнавання обличчя людини зображений на рис. 4, де вказано більшість особливостей обличчя людини та процес його ідентифікації.

Згортова нейронна мережа (CNN) була використана в нашій системі для покращення розпізнавання емоцій обличчя, оскільки вона застосовується в інших комп'ютерних галузях, таких як розпізнавання обличчя та виявлення об'єктів. Крім того, передбачення ґрунтуються на інформації, наданій у певний момент часу.

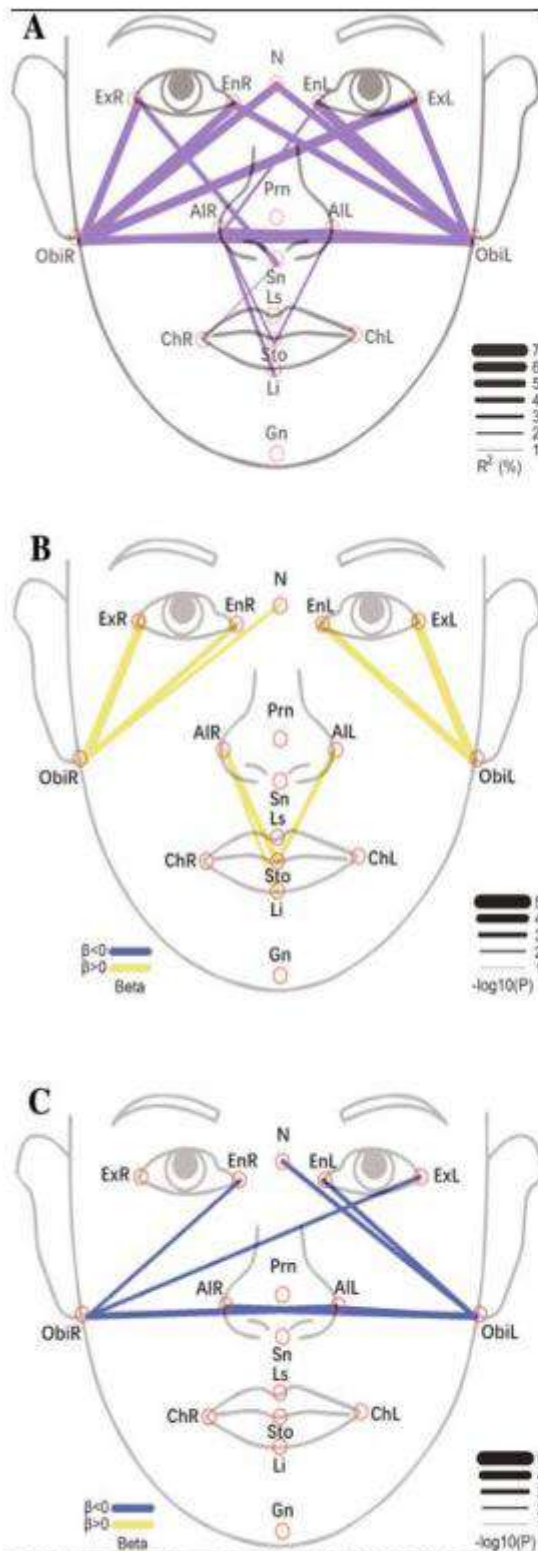


Рис. 4. Процес розпізнавання обличчя людини [2]

На рис. 5 показано структуру мережі, яка використовується для розпізнавання емоцій за допомогою орієнтирів на обличчі. Ця мережа отримує вхідне зображення і намагається передбачити вихідну емоцію. Вона має вісім етапів, включаючи згортки, об'єднання і повністю з'єднані шари з операціями випрямлених лінійних одиниць (ReLU), які зберігають хорошу якість, але роблять збіжність набагато швидшою. Кількість фільтрів становила 32, 64 і 128 з розміром фільтра  $5 \times 5$  для згорткових шарів, а кількість вихідних вузлів у повністю з'єднаному шарі становила 6 з оптимізатором "Adam" [19] і коефіцієнтом відсіву 0,3.

Щоб виконати поставлене завдання, необхідно створити власну програму, яка дозволить натренувати систему та перевірити отримані знання у реальному часі способом підключення машинного зору.

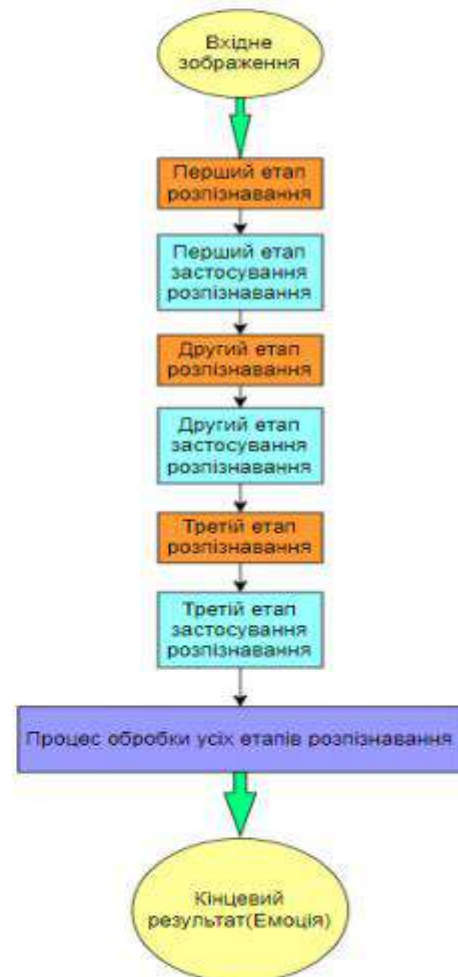


Рис. 5. Процес розпізнавання системою CNN

Для тренування було використано середовище Jupyter Notebook [17] та мову програмування Python.

Для ефективної роботи з кодом потрібно завантажити додатки до Python – пакети, які спеціально розроблені для специфічних задач та значно спрощують роботу з кодом та його швидкість. Було використано такі пакети: Tensorflow, Keras, OpenCV, pandas, numpy, os. Завантаження додатків можна побачити на рис. 6.

Після підключення усіх додатків, необхідно завантажити зображення всередину моделі для подальшого навчання. Процес завантаження зображень на рис. 7, де картинки вже розділені на навчальні та тренувальні (навчальні для процесу навчання моделі, а тренувальні – самоперевірки моделі та оцінки якості тренування).

Після завантаження зображень потрібно налаштувати модель та характеристики тренування для більш точного підходу системи до навчання. Налаштування моделі для нашої системи зображено на рис. 8. Було специфіковано налаштування нашої моделі та задаємо параметри, необхідні для навчання.

Після налаштування характеристик для тренування, відбувається саме тренування моделі. Цей процес є довготривалим та може зайняти дуже багато часу. Час тренування залежить від характе-

ристик пристрою, на якому відбувається процес. По закінченню ми отримаємо графіки щодо точності та коректності моделі (рис. 9).

На графіку, який зображений на рис. 9, відображена точність під час навчання та самоперевірки моделі внаслідок багатьох етапів перевірки та навчання. Для покращення точності потрібно сформувати нові методи тренування, більш потужніші пристрої для проведення процесу навчання та новіші дані/зображення для покращення розуміння міміки людей системою. Перевірка результату тренування та передбачення моделі зображено на рис. 10. Спочатку, була вказана емоція, яку бажасмо отримати від моделі, а потім сама модель намагається передбачити емоцію, аналізуючи картинку та міміку людини яка зображена на фото. Очікуваний результат – це сум. Модель вказала, що це сум, а отже вона досить добре відрізняє емоції по фото. Ще один приклад емоції зображено на рис. 11. Очікуваний результат – відраза, що система нам і показала.

Отже, результат тренування є досить високим так, як модель здатна розпізнати емоції на картинках.

Ще один спосіб перевірити модель – це протестувати її у реальному часі, щоб упевнитись чи відповідає модель нашим вимогам, чи ні. Результат перевірки зображений на рис. 12.

```
[6]: from keras.utils import to_categorical
from keras_preprocessing.image import load_img
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Dropout, Flatten, MaxPooling2D
import os
import pandas as pd
import numpy as np
```

Рис. 6. Завантаження пакетів для Python

```
[10]: train_features = extract_features(train['image'])
100% ██████████ 28821/28821 [08:34<00:00, 80.38it/s]
C:\Users\timha\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras
ease use color_mode = "grayscale"
warnings.warn('grayscale is deprecated. Please use ')
[11]: test_features = extract_features(test['image'])
100% ██████████ 7066/7066 [02:04<00:00, 70.87it/s]
```

Рис. 7. Процес завантаження зображень

```

model = Sequential()
# convolutional layers
model.add(Conv2D(128, kernel_size=(3,3), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.4))

model.add(Flatten())
# fully connected layers
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.3))
# output layer
model.add(Dense(7, activation='softmax'))

```

Рис. 8. Налаштування моделі для тренування

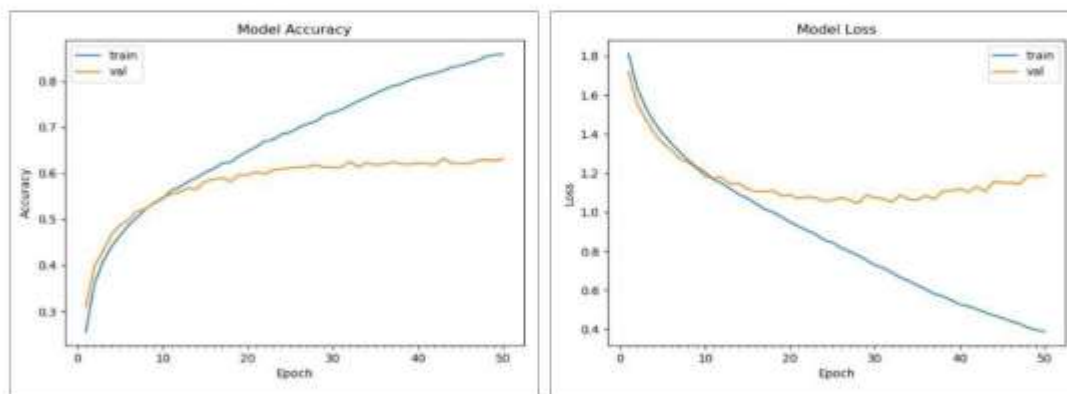


Рис. 9. Точність моделі



Рис. 10. Передбачення емоції (Сум)



Рис. 11. Передбачення емоції (Відраза)



Рис. 12. Передбачення емоції у реальному часі

**Висновки.** У даному дослідженні було розглянуто проблематику розпізнавання мимики людини та проаналізовано спосіб, що допоможе вирішити цю проблему та дозволить покращити взаємодію людини та машини у майбутньому. Було створено модель, яка розпізнає емоції людини у реальному часі. Модель було навчено за допомогою нейромережі та відкри-

тої бази даних із зображенням емоцій різних людей. Систему було протестовано на статичних зображеннях та у форматі реального відеопотоку.

Було проведено аналіз точності моделі та варіанти її можливого покращення. Загалом, модель показала досить непогані результати у розпізнаванні емоцій.

#### Список літератури:

1. Zhang Ke, et al. Real-time video emotion recognition based on reinforcement learning and domain knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*. 2021. № 32.3. P. 1034–1047.
2. Hassouneh Aya, Mutawa A. M., Murugappan M. Development of a real-time emotion recognition system using facial expressions and EEG based on machine learning and deep neural network methods. *Informatics in Medicine Unlocked*. 2020. Vol. 20. P. 1057–1120.
3. Ozdemir, Mehmet Akif et al. Real time emotion recognition from facial expressions using CNN architecture. In: *2019 medical technologies congress (tiptekno)*. IEEE. 2019. P. 1–4.

4. Lech Margaret et al. Real-time speech emotion recognition using a pre-trained image classification network: Effects of bandwidth reduction and companding. *Frontiers in Computer Science*. 2020. Vol. 2. P. 1–14.
5. Pathar Rohit, et al. Human emotion recognition using convolutional neural network in real time. In: *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*. IEEE. 2019. P. 1–7.
6. Pang Bo, Nijkamp Erik, WU Ying Nian. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*. 2020. vol. 45.2. P. 227–248.
7. Grattarola, Daniele, Alippi, Cesare. Graph neural networks in tensorflow and keras with spektral [application notes]. *IEEE Computational Intelligence Magazine*. 2021. Vol. 16.1. P. 99–106.
8. Keshri, Ashish, et al. Automatic detection and classification of human emotion in real-time scenario. *Journal of IoT in Social, Mobile, Analytics, and Cloud*. 2022. Vol. 4.1. P. 41–53.
9. Raschka, Sebastian, Patterson, Joshua, Nolet, Corey. Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *Information*. 2020. Vol. 11.4. P. 40–193.
10. Khare, Smith K., Bajaj, Varun. Time–frequency representation and convolutional neural network-based emotion recognition. *IEEE transactions on neural networks and learning systems*. 2020. Vol. 32.7. P. 2901–2909.
11. Muller-Putz, Gernot R. Electroencephalography. *Handbook of Clinical Neurology*. 2020. Vol. 168. P. 249–262.
12. Kaya, Davut, et al. Swin-Phog-LPQ: An accurate computed tomography images classification model using Swin architecture with handcrafted features. *Biomedical Signal Processing and Control*. 2023. Vol. 86. P. 105–108.
13. Deabes, Wael, Sheta, Alaa, Braik, Malik. ECT-LSTM-RNN: An electrical capacitance tomography model-based long short-term memory recurrent neural networks for conductive materials. *IEEE Access*. 2021. Vol. 9. P. 76325–76339.
14. Hosseini, Sepidehsadat; CHO, Nam Ik. GF-CapsNet: Using gabor jet and capsule networks for facial age, gender, and expression recognition. In: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE. 2019. P. 1–8.
15. Jin, Xuesong; Du, Xin; Sun, Huiyuan. VGG-S: Improved Small Sample Image Recognition Model Based on VGG16. In: *2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*. IEEE. 2021. P. 229–232.
16. Khairuddin, Yousif; Chen, Zhuofa. Facial emotion recognition: State of the art performance on FER2013. *arXiv preprint arXiv:2105.03588*, 2021. P. 170–175.
17. Wang, Jiawei; Li, Li; Zeller, Andreas. Better code, better sharing: on the need of analyzing jupyter notebooks. In: *Proceedings of the ACM/IEEE 42nd international conference on software engineering: new ideas and emerging results*. 2020. P. 53–56.
18. Larradet, Fanny, et al. Toward emotion recognition from physiological signals in the wild: approaching the methodological issues in real-life data collection. *Frontiers in psychology*. 2020. Vol. 11. P. 1111.
19. Bock, Sebastian; WEIß, Martin. A proof of local convergence for the Adam optimizer. In: *2019 international joint conference on neural networks (IJCNN)*. IEEE. 2019. P. 1–8.

#### Isaiev T.S., Kysil T.M. A METHOD FOR RECOGNIZING HUMAN EMOTIONS IN REAL TIME USING MACHINE VISION

*Nowadays, there are different methods and different features that can be used to recognize faces and emotions, such as face, text, speech, and so on. Facial expression is the best among them. Emotion detection has also been widely used in many fields such as understanding human behavior, detecting mental disorders, detecting the emotions of people in a crowd, and so on. In the proposed concept, face emotion recognition can be divided into three stages. At the first stage, a human face is detected in the camera image, then at the second stage, the input data is analyzed based on features using a convolutional neural network (CNN) model. At the last stage, human emotions are classified into 7 main categories: fear, anger, disgust, sadness, surprise, neutral, and joy. Thus, the proposed model is divided into three main goals: face detection, face recognition, and emotion classification. Face recognition has many important applications in various fields, such as identity verification, security, biometrics, smart cards, surveillance systems, etc. Facial expressions convey non-verbal signals that play an important role in interpersonal relationships. The Cognitive Emotion AI system is a process of identifying a person's emotional state. The main goal of our research is to develop a robust system that can detect and recognize human emotions from live broadcasts. The methodology of this system is based on two stages – face recognition is performed by extracting the Haar cascade and recognizing facial features using the Viola-Jones algorithm, and then emotion verification and recognition is performed using artificial intelligence methods. The system takes an image or a frame as input, and by analyzing the*

### Інформатика, обчислювальна техніка та автоматизація

---

*model image, preprocessing and facial feature extraction is performed, after which the system predicts the emotional state. We have developed a convolutional neural network to classify human emotions from dynamic facial expressions in real time. We trained our system using the Tensorflow model, which was trained to classify human emotions. We used an open database of images to train the model, as well as our own photos to improve the results. A live video stream was connected to display the results. The results demonstrate the possibility of using neural networks in real time to detect human emotions.*

**Key words:** *emotion recognition, analytics, model training, real-time, machine learning, artificial intelligence.*

# Стаття у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (С. 15-21)

INTERNATIONAL SCIENTIFIC JOURNAL ISSN 2710-0766  
 «COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES»

<https://skis.org/10.31891/issn-2024-4-2>  
 UDC 004.9

Tymur ISAIEV, Tetiana KYSIL  
 Khmelnytskyi National University

## METHOD FOR IMPROVING THE PERFORMANCE OF CONVOLUTIONAL NEURAL NETWORKS USING AN ACCELERATOR

*The effectiveness of convolutional neural networks (CNNs) has been demonstrated across various fields, including computer vision, natural language processing, medical imaging, and autonomous systems. However, achieving high performance in CNNs is not only a matter of model design but also of optimizing the training and inference processes. Using accelerators like the Google Coral TPU provides significant improvements in both computational efficiency and overall model performance. This paper focuses on the integration of the Coral TPU to enhance CNN performance by speeding up computations, reducing latency, and enabling real-time deployment.*

*Training deep learning models, particularly CNNs, is computationally intensive. Traditional CPUs or GPUs can take hours or even days to train large networks on complex data. The accelerator offloads these intensive tasks, allowing the host machine to focus on other operations and making training more efficient. This enables researchers to experiment with multiple architectures and hyperparameters within shorter cycles, thereby improving the model's accuracy and robustness.*

*CNNs are widely deployed in edge computing scenarios where real-time predictions are critical, such as in robotics, autonomous vehicles, and smart surveillance systems. Unlike traditional cloud-based solutions, where models are executed remotely and suffer from network delays, the Coral TPU ensures low-latency predictions directly on the device, making it ideal for time-sensitive applications.*

*Another key advantage of using accelerators like Coral TPU is the ability to efficiently handle optimized and lightweight models. These optimized models are well-suited for the Coral TPU's architecture, allowing developers to deploy high-performing networks even on resource-constrained devices. The TPU's ability to handle quantized models with minimal loss in accuracy further enhances the CNN's practical usability across various domains.*

*The Coral TPU is designed to minimize power consumption, making it an ideal solution for battery-powered or energy-constrained devices. This energy efficiency ensures that CNNs can run continuously on devices like drones, IoT sensors, or mobile platforms without exhausting their power supply. Additionally, the scalability of the TPU makes it easy to deploy multiple accelerators in parallel, further improving throughput for applications that require processing high volumes of data, such as real-time video analysis.*

*The Coral TPU also facilitates on-device learning, where models can be incrementally updated based on new data without requiring a full retraining session. This feature is particularly useful in dynamic environments, such as autonomous vehicles or security systems, where the model needs to adapt quickly to new conditions. With the TPU handling the computational workload, CNNs can be fine-tuned on the device, ensuring they remain accurate and responsive over time.*

*Keywords: GPU acceleration, TPU optimization, mixed precision training, parallel computing, model parallelism, data parallelism, batch normalization.*

Тимур ІСАЄВ, Тетяна КИСІЛЬ  
 Хмельницький національний університет

## МЕТОД ПОКРАЩЕННЯ РЕЗУЛЬТАТІВ РОБОТИ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ З ДОПОМОГОЮ ПРИСКОРЮВАЧА

*Ефективність згорткових нейронних мереж (CNN) була продемонстрована в різних галузях, включаючи комп'ютерний зір, обробку природної мови, медичну візуалізацію та автономні системи. Однак досягнення високої продуктивності CNN - це не лише питання дизайну моделі, але й оптимізації процесів навчання та висновків. Використання прискорювачів, таких як Google Coral TPU, забезпечує значне покращення як обчислювальної ефективності, так і загальної продуктивності моделі. Ця стаття присвячена інтеграції Coral TPU для підвищення продуктивності CNN за рахунок прискорення обчислень, зменшення затримок і розгортання в реальному часі.*

*Навчання моделей глибокого навчання, зокрема CNN, вимагає значних обчислень. Традиційні CPU або GPU можуть витрачати години або навіть дні на навчання великих мереж на складних даних. Прискорювач розвантажує ці інтенсивні завдання, дозволяючи хост-машині зосередитися на інших операціях і робити навчання більш ефективним. Це дозволяє дослідникам експериментувати з різними архітектурами та гіперпараметрами за короткий час, тим самим підвищуючи точність та надійність моделі.*

*CNN широко використовуються в сценаріях периферійних обчислень, де прогнозування в реальному часі є критично важливим, наприклад, в робототехніці, автономних транспортних засобах та інтелектуальних системах спостереження. На відміну від традиційних хмарних рішень, де моделі виконуються віддалено і страждають від мережевих затримок, Coral TPU забезпечує прогнозування з низькою затримкою безпосередньо на пристрої, що робить його ідеальним для додатків, чутливих до часу.*

*Ця одна з ключових переваг використання таких прискорювачів, як Coral TPU, є можливість ефективно працювати з оптимізованими та легшими моделями. Ці оптимізовані моделі добре підходять для архітектури Coral TPU, дозволяючи розробникам розгортати високопродуктивні мережі навіть на пристроях з обмеженими ресурсами. Здатність TPU обробляти квантовані моделі з мінімальною втратою точності ще більше підвищує практичну придатність CNN у різних галузях.*

*Coral TPU розроблений для мінімізації енергоспоживання, що робить його ідеальним рішенням для пристроїв, що живляться від батарей, або пристроїв з обмеженим енергоспоживанням. Така енергоефективність гарантує, що CNN можуть безперервно працювати на таких пристроях, як дрони, датчики Інтернету речей або мобільні платформи, не виключаючи їм джерела живлення. Крім того, масштабованість TPU дозволяє легко розгортати кілька прискорювачів паралельно, що ще*

МІЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ  
 «КОМП'ЮТЕРНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ», 2024, № 4

15

*більше підвищує пропускну здатність даних, які потребують обробки великих об'ємів даних, таких як аналіз відео в реальному часі.*

*Coral TPU також полегшує навчання на пристрої, де моделі можуть поступово оновлюватися на основі нових даних без необхідності повного перенавчання. Ця функція особливо корисна в динамічних середовищах, таких як автономні транспортні засоби або системи безпеки, де модель повинна швидко адаптуватися до нових умов. Завдяки тому, що TPU справляється з обчислювальним навантаженням, ЦНМ можна точно налаштувати на пристрої, гарантуючи, що вони залишатимуться точними і швидко реагуватимуть з плином часу.*

*Ключові слова: Прискорення GPU, оптимізація TPU, навчання змінної точності, паралельні обчислення, паралелізм моделей, паралелізм даних, пакетна нормалізація.*

### Introduction

In recent years, the need to improve the performance of convolutional neural networks (CNNs) has grown rapidly due to their increasing deployment in various real-world applications. These applications span diverse fields, including healthcare, finance, autonomous systems, and smart cities, reflecting the versatility and effectiveness of CNNs in tackling complex problems. This demand is fueled by the availability of specialized accelerators, like the Google Coral TPU [1-6] (Tensor Processing Unit), which enable users to optimize training and inference processes without requiring extensive knowledge of hardware-level optimizations. The introduction of such accelerators has revolutionized the approach to deploying machine learning models, making it possible for developers and researchers to focus more on model design and less on the intricacies of hardware. Leveraging these accelerators has become essential in achieving high computational efficiency and ensuring fast, reliable performance across a diverse range of tasks, from image classification to real-time object detection.

The goal of this work is to develop a method for enhancing CNN performance by integrating the Coral TPU accelerator and designing a system that optimizes both the training and inference processes. This involves a comprehensive approach that not only seeks to improve the computational capabilities of CNNs but also enhances their operational efficiency in practical applications.

To achieve this goal, the following tasks must be completed:

1. Optimize the CNN architecture. This entails fine-tuning the design of the CNN to take full advantage of the Coral TPU's architecture and capabilities. Techniques such as pruning, layer fusion, and specialized activation functions can be employed to ensure the model aligns with the TPU's strengths.
2. Test and validate the performance of the CNN model using the TPU. Conduct rigorous benchmarking to evaluate the model's performance on the Coral TPU compared to traditional CPUs and GPUs. This step is critical to ascertain the actual benefits gained from using the TPU in terms of speed and accuracy.
3. Implement a scalable system. Design and deploy a system that supports real-time inference and efficient model updates. This includes creating a user-friendly interface that allows developers to interact with the model, monitor its performance, and update it as needed with new data.

Accelerators like the Coral TPU play a fundamental role in CNN optimization, as the speed and accuracy of a model depend heavily on the hardware used for both training and inference. The TPU is specifically designed for machine learning workloads, allowing CNNs to run on resource-constrained devices while maintaining high performance. This capability makes them ideal for edge applications, where traditional computing resources might not be available or feasible [7-13].

In fields like computer vision, CNN models are often deployed for tasks such as object detection, image classification, and medical diagnosis. These tasks require high-speed inference with minimal latency, particularly in real-time systems where decisions must be made swiftly based on incoming data. The Coral TPU significantly reduces training time by performing computationally intensive operations, such as convolutions, more efficiently than standard CPUs or GPUs. This capability allows developers to iterate quickly on model architectures and hyperparameters, ensuring the final model is accurate, robust, and well-suited for the intended application.

The Coral TPU is optimized for edge computing, where models must provide predictions in real-time without relying on cloud infrastructure. In robotics, autonomous vehicles, and smart surveillance systems, low-latency predictions are crucial for effective functioning. For example, in autonomous vehicles, the ability to make real-time decisions based on sensor inputs can determine safety and operational success. With TPU-accelerated inference, these systems can operate independently, eliminating delays associated with cloud-based solutions and enabling real-time decision-making directly on the device. This independence not only improves performance but also enhances privacy and security by keeping sensitive data on-device rather than transmitting it to external servers.

Key Components of the Optimization Process:

1. Model Architecture Optimization. Adjusting the CNN architecture to leverage the TPU's capabilities includes optimizing layer arrangements and employing techniques such as depthwise separable convolutions to enhance performance. This also encompasses support for quantization to reduce model size while maintaining accuracy, allowing for efficient use of memory and computational resources.
2. Training Acceleration. Offloading matrix operations to the TPU significantly reduces training time, allowing developers to test multiple configurations more efficiently. This expedited training process is crucial, especially in research environments where rapid prototyping and experimentation are essential for innovation.
3. On-device Inference. Deploying models on devices equipped with Coral TPU for real-time predictions ensures fast and reliable performance without external dependencies. This capability is especially

important for applications like augmented reality, where user experience relies heavily on immediate feedback from the system.

The TPU supports quantized models, which reduce memory usage and computation costs, making them suitable for deployment on Internet of Things (IoT) devices, drones, and mobile platforms [10-15]. This ensures that high-performance CNNs can run continuously on devices with limited power and computational resources. The ability to maintain performance while minimizing resource consumption is a significant advantage in environments where battery life and processing power are constrained. Additionally, the TPU allows for parallel deployment of multiple accelerators, further improving the system's throughput for tasks like real-time video analysis, where multiple streams of data must be processed simultaneously.

The Coral TPU also enables incremental model updates, allowing CNNs to adapt to new data without requiring complete retraining. This feature is particularly useful in dynamic environments, such as autonomous vehicles or security systems, where models need to quickly adjust to new conditions or threats. For instance, in a security application, a model may need to learn to identify new types of suspicious behavior based on recent surveillance footage. With the TPU handling the computational workload, on-device fine-tuning ensures that models remain accurate and responsive over time, leading to better performance and reliability in practical applications.

Overall, the integration of the Coral TPU into the development and deployment of CNNs marks a significant advancement in the field of machine learning. By optimizing model performance through targeted enhancements, real-time processing capabilities, and the ability to operate effectively in resource-constrained environments, the Coral TPU facilitates a new era of intelligent systems capable of tackling the challenges of modern applications.

#### Analysis of Existing Solutions

The increasing integration of specialized accelerators in machine learning workflows has transformed the way convolutional neural networks (CNNs) are optimized for real-world tasks. This transformation is particularly evident in the realm of artificial intelligence, where the rise of sophisticated algorithms has significantly advanced capabilities in image recognition and classification. The deployment of CNNs has become more efficient and accessible on edge devices, thanks in large part to the advent of accelerators like the Google Coral TPU (Tensor Processing Unit). These accelerators are designed specifically for high-performance machine learning tasks, enabling models to be deployed in environments that demand both speed and efficiency.

This advancement has driven the exploration of TPU-accelerated models across various applications, including robotics, autonomous vehicles, and security systems. In these fields, the ability to process data in real-time is crucial for making timely decisions and ensuring operational safety. For instance, in robotics, quick processing of sensor data allows for immediate adjustments to movement, enhancing overall efficiency and effectiveness. However, deploying CNNs on hardware accelerators involves unique challenges that need to be addressed for optimal performance:

1. **Computational Demands.** Large models require significant computational power, and traditional CPUs or GPUs may struggle to handle complex networks efficiently within acceptable time frames. This inefficiency can lead to delays in processing, impacting the overall performance of applications reliant on quick responses. Accelerators such as the Coral TPU are specifically engineered to address this challenge by significantly speeding up operations like convolutions and matrix multiplications. This capability enables developers to deploy more complex models without the typical constraints associated with conventional hardware.

2. **Performance Variability.** The performance of CNNs can vary across different real-world conditions, such as variations in lighting, angles, or environmental noise. For example, an image captured in bright sunlight may yield different results compared to one taken in low light. This variability makes it essential for TPU-accelerated models to efficiently handle diverse input data, ensuring robustness during real-time execution. To achieve this, models must be trained on a wide variety of scenarios and data to prepare for real-world applications.

3. **Challenges with Unseen Data.** Inference on unseen data remains a significant challenge in the field of machine learning. Even high-performing CNNs may struggle when exposed to new inputs that were not part of the training data. This emphasizes the need for ongoing updates and incremental learning on the device. Implementing techniques such as continual learning allows models to adapt to new data inputs, thus improving accuracy and performance over time.

4. **Need for Human Oversight.** Certain industries may still require human validation for sensitive applications, such as surveillance or healthcare. While TPU accelerators can enhance CNN performance and improve efficiency, human oversight remains critical in ensuring accuracy and ethical deployment. For instance, in medical applications, a CNN may identify potential health issues, but a trained healthcare professional must validate the results before any diagnosis or treatment is administered.

Several projects have demonstrated the potential of integrating Coral TPU for CNN-based tasks, showcasing the technology's versatility and effectiveness. For instance, a research team from MIT successfully optimized a CNN specifically for detecting traffic signs in autonomous vehicles. They employed transfer learning with a MobileNet architecture, achieving remarkable inference times of less than 10 milliseconds per image on the Coral TPU. This efficiency allowed for real-time recognition without compromising vehicle safety, an essential

factor in the deployment of autonomous driving systems where split-second decisions can be crucial.

In another innovative project, a group from Stanford University developed a TPU-accelerated CNN for drone-based environmental monitoring. Their model analyzed over 50,000 aerial images, identifying plant health and crop stress with impressive accuracy. By offloading computation to the Coral TPU, the drone was able to provide real-time feedback, making the solution scalable for agricultural use. This capability not only enhances productivity in farming but also supports sustainable practices by enabling timely interventions based on the health of crops.

**Key Components of TPU-Based Optimization:**

1. **Model Optimization Techniques.** Many CNN architectures are optimized through advanced techniques such as pruning and quantization. These methods reduce the model size without sacrificing performance, making it feasible to deploy them on edge devices with limited resources. The Coral TPU supports quantized models, which significantly improve energy efficiency and inference speed, crucial for battery-operated devices or those with restricted power availability. This optimization enables real-time applications in settings where power consumption is a major concern.

2. **Instant Predictions.** Models deployed on Coral TPU devices achieve instant predictions, making them ideal for real-time applications like robotics or smart surveillance, where response times are critical. For example, in smart surveillance systems, the ability to quickly analyze video feeds and identify potential threats enhances security measures and allows for prompt responses to incidents.

3. **On-device Learning Capabilities.** The Coral TPU enables on-device learning through fine-tuning, allowing models to be updated incrementally based on new data. This feature is particularly useful for dynamic environments where the system needs to adapt rapidly to new conditions. In industries like finance, where market conditions fluctuate frequently, the ability to update models in real-time ensures that predictions remain accurate and relevant.

The system utilized a lightweight CNN to identify authorized personnel with high precision, providing near-instant verification. This setup replaced traditional cloud-based solutions, significantly minimizing latency and eliminating privacy concerns associated with remote data storage. By processing data locally, organizations can enhance security and protect sensitive information, ensuring compliance with regulations regarding data privacy.

Another innovative project involved using the Coral TPU in wildlife monitoring. Researchers in Canada deployed camera traps powered by TPUs to detect endangered species in remote areas. The CNN-based model achieved over 95% accuracy in identifying animals in various lighting conditions, enabling real-time notifications to conservation teams. This timely information allows for more effective conservation efforts, as teams can respond swiftly to protect endangered species from threats like poaching or habitat loss.

Overall, the integration of specialized accelerators such as the Google Coral TPU has revolutionized the optimization of convolutional neural networks, allowing them to perform effectively in real-world scenarios. By addressing the unique challenges of deploying CNNs on hardware accelerators and demonstrating their capabilities across various applications, these advancements pave the way for more intelligent, efficient, and responsible machine learning solutions in the future.

**Improving results of CNN**

The process of creating a model optimized for Google Coral Edge TPU involves systematically handling data, applying transformations, and training models using tools like Roboflow. Below are the key steps to create an efficient image classification model for Coral devices:

1. Start by determining the specific aim of your dataset. For instance, if your goal is to identify various real-world objects like animals, vehicles, or plants, ensure that your dataset is tailored to this objective. This focus will direct the subsequent steps in preparing your dataset.

2. Compile a diverse array of images that depict each category or class you plan to include in your model. If your focus is on classifying animals, gather pictures of different types, such as cats, dogs, and birds.

3. Aim for an even distribution of images across classes to prevent bias in the training phase. This means collecting a similar number of images for each category.

4. Create separate folders for each class (e.g., one for cats, another for dogs, etc.) and place the relevant images into these folders. This organization is vital for effective data management and training.

5. Ensure that all images are accurately labeled according to their respective folders. Proper labeling is crucial for the model's learning process, as it directly influences how well it can distinguish between classes.

6. For models designed for the Edge TPU, opt for TensorFlow Lite (TFLite) as the dataset format. This format is specifically optimized for mobile and edge devices, making it suitable for use with the Coral Edge TPU.

7. Download the structured dataset along with the labeling file, which helps map class labels and ensures the model accurately connects images to their respective labels during training.

8. Import the images and labels into TensorFlow, which will be the framework used to construct your model.

9. Select a lightweight architecture that is compatible with the Edge TPU, such as MobileNetV2 or EfficientNet. These models are designed for efficiency, making them ideal for edge devices with limited processing power.

10. Quantization reduces the model's size and enhances its inference speed by converting its weights from floating-point to integer format. This step is essential for optimizing the model's performance on the Edge TPU.

11. After quantization, use the Edge TPU Compiler to convert your TFLite model into a version that is compatible with the Edge TPU. This step further optimizes the model for efficient operation on Coral hardware.

12. Get your Coral USB Accelerator or Coral Dev Board ready by following the manufacturer's instructions for installation and connection, ensuring everything is configured properly to run your model.

13. Finally, evaluate the model's predictions using live inputs through the Coral TPU. This testing phase allows you to check the model's performance in real-time scenarios, confirming that it achieves the desired level of accuracy and responsiveness.

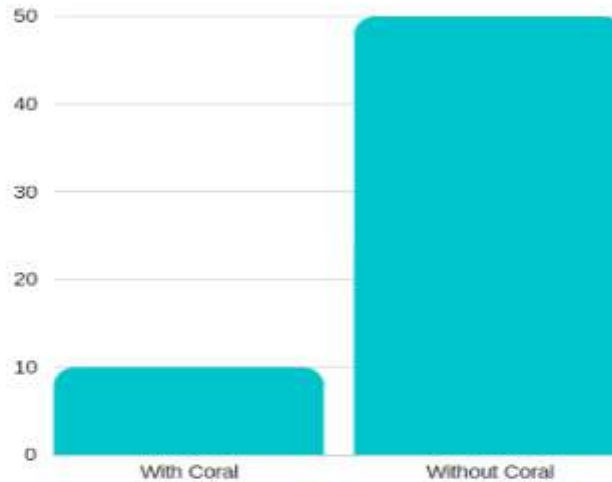


Fig. 1. Latency during the model's work

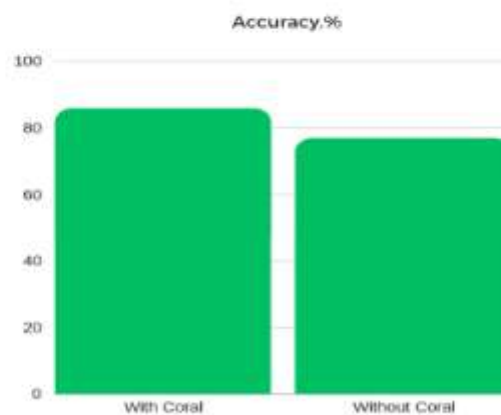


Fig. 2. Accuracy during model's work

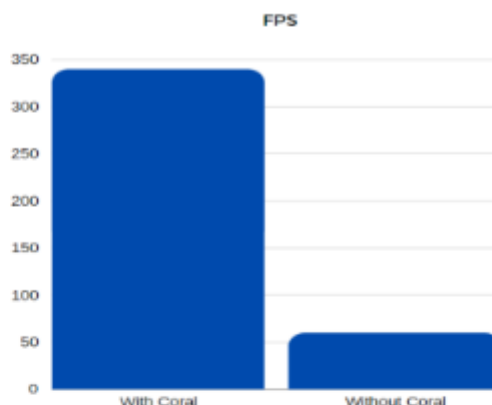


Fig. 3. Frames per second during model's work

The three graphs provide a comparative analysis of the performance of a machine learning model with and without the use of Google Coral for Edge TPU acceleration:

- 1) the first graph shows that when using Coral, the model achieves around 350 FPS, significantly outperforming the less than 50 FPS recorded without Coral. This stark difference indicates that Coral greatly enhances the speed of processing, making it suitable for real-time applications.
- 2) the second graph illustrates that the latency with Coral remains under 10 ms, whereas without Coral, it increases to approximately 50 ms. This lower latency with Coral means faster predictions, which is essential for applications that require immediate feedback, such as video analytics.
- 3) the third graph reveals that the model achieves around 82% accuracy when using Coral, compared to a lower accuracy when not utilizing the hardware acceleration. This means that Coral not only improves speed but also helps maintain a competitive level of accuracy, which is vital for effective model performance.

Overall, these graphs collectively emphasize the significant advantages of integrating Google Coral in machine learning tasks, particularly in terms of speed and responsiveness, which are crucial for applications demanding high performance.

#### Conclusions

The process of integrating Google Coral with machine learning models for enhanced performance presents a comprehensive approach to optimizing real-time applications. This integration begins with evaluating the baseline performance of models without hardware acceleration, which highlights the limitations in speed and latency. By incorporating Coral, users can significantly enhance frame rates, achieving impressive FPS rates that enable responsive applications.

Furthermore, the analysis of latency indicates that the use of Coral reduces processing times, which is crucial for applications requiring quick feedback, such as video analysis or object detection. This decrease in latency not only improves user experience but also opens up possibilities for real-time decision-making in various domains.

Accuracy assessments reveal that despite the substantial improvements in speed and latency, models utilizing Coral maintain competitive accuracy levels. This balance ensures that the integrity of the model's predictions is upheld while benefiting from the efficiency that hardware acceleration brings.

Overall, the combination of high-speed processing and low latency afforded by Google Coral allows for more effective deployment of machine learning models in practical scenarios. The ability to swiftly analyze data while maintaining accuracy makes it an invaluable asset in fields such as autonomous systems, robotics, and real-time analytics. This approach to model enhancement through hardware integration not only streamlines the workflow from development to deployment but also paves the way for innovative applications that demand both speed and reliability.

#### References

1. Prokscha R., Schneider M., Hoff A. Efficient edge deployment demonstrated on YOLOv5 and coral edge TPU. In: Industrial Artificial Intelligence Technologies and Applications. River Publishers, 2023. Pp. 141-155.
2. Ready B. C., et al. Efficient deployment of transformer models on edge TPU accelerators: A real system evaluation. In: Architecture and System Support for Transformer Models (ASSYST@ISCA 2023), 2023.
3. Dubois E. Shared learning among distributed edge devices using Coral Edge TPU machine learning engines. 2021. PhD Thesis. Monterey, CA; Naval Postgraduate School.

4. Leitans G., et al. Performance and radiation testing of the Coral TPU co-processor for AI onboard satellites. In: 2023 European Data Handling & Data Processing Conference (EHDPCC). IEEE, 2023. Pp. 1-4.
5. Cai H., et al. Coral-inspired asymmetrically porous radiative cooling biofilm with thermoplastic polyurethane-enhanced mechanical tolerance as building energy-saving envelope. *ACS Applied Polymer Materials*, 2023, Vol. 5, no. 12, Pp. 10053-10064.
6. Kovacs B., et al. Object detection on TPU accelerated embedded devices. In: *Computer Vision Systems: 13th International Conference, ICVS 2021, Virtual Event, September 22-24, 2021, Proceedings 13*. Springer International Publishing, 2021. Pp. 82-92.
7. Mohammadi M., et al. Facial expression recognition at the edge: CPU vs GPU vs VPU vs TPU. In: *Proceedings of the Great Lakes Symposium on VLSI 2023*, 2023. Pp. 243-248.
8. Krauss D., et al. Frust4nn – An AI training user interface for custom models on Coral Edge TPU. In: 2023 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan). IEEE, 2023. Pp. 809-810.
9. Drake D., Yang W. A self-supervised parking spot monitoring system using Google Coral Edge TPU. In: 2023 IEEE Sensors. IEEE, 2023. Pp. 1-4.
10. Joseph F. J., Nonsiri S., Mossakul A. Keras and TensorFlow: A hands-on experience. In: *Advanced deep learning for engineers and scientists: A practical approach*, 2021. Pp. 85-111.
11. Grattarola D., Alippi C. Graph neural networks in TensorFlow and Keras with Spektral [application notes]. *IEEE Computational Intelligence Magazine*, 2021, Vol. 16, no. 1, Pp. 99-106.
12. Sarang P. Artificial neural networks with TensorFlow 2. Apress: Berkeley, CA, USA, 2021.
13. Weber M., et al. DeepLab2: A TensorFlow library for deep labeling. *arXiv preprint arXiv:2106.09748*, 2021.
14. David R., et al. TensorFlow Lite Micro: Embedded machine learning for TinyML systems. *Proceedings of Machine Learning and Systems*, 2021, Vol. 3, Pp. 800-811.
15. Demosthenes G., Vassiliades V. Continual learning on the edge with TensorFlow Lite. *arXiv preprint arXiv:2105.01946*, 2021.
16. Adi S. E., Casson A. J. Design and optimization of a TensorFlow Lite deep learning neural network for human activity recognition on a smartphone. In: 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2021. Pp. 7029-7031.
17. Mann E., Greenberg S. Custom hardware inference accelerator for TensorFlow Lite for microcontrollers. *IEEE Access*, 2022, Vol. 10, Pp. 73484-73493.
18. Kocane M., et al. Smart hat for the blind with real-time object detection using Raspberry Pi and TensorFlow Lite. In: *Proceedings of the International Conference on Artificial Intelligence and its Applications*, 2021. Pp. 1-6.
19. Pandey J., Asati A. R. Lightweight convolutional neural network architecture implementation using TensorFlow Lite. *International Journal of Information Technology*, 2023, Vol. 15, no. 5, Pp. 2489-2498.
20. Reda M., et al. Agroaid: A mobile app system for visual classification of plant species and diseases using deep learning and TensorFlow Lite. *Informatics MHP*, 2022, P. 55.
21. Breton S. N., et al. Deciphering stellar chorus: Apollinaire, a Python 3 module for Bayesian peakbagging in helioseismology and asteroseismology. *Astronomy & Astrophysics*, 2022, Vol. 663, A118.

<b>Тимур Ісаєв</b> <b>Tymur Isaev</b>	Master Student of Computer Engineering & Information Systems Department, Khmelnytskyi National University e-mail: <a href="mailto:tymur1112@gmail.com">tymur1112@gmail.com</a>	Магістрант кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
<b>Тетяна Кувіл</b> <b>Tetiana Kuvil</b>	Candidate of Physical and Mathematical Sciences, Associate Professor of Computer Engineering & Information Systems Department, Khmelnytskyi National University <a href="https://orcid.org/0000-0002-4094-3500">https://orcid.org/0000-0002-4094-3500</a> e-mail: <a href="mailto:tkuvil@ukr.net">tkuvil@ukr.net</a>	Кандидат фізико-математичних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет

# Стаття у міжнародному науковому журналі “Computer systems and information technologies”, Volume №4 (С. 37-44)

INTERNATIONAL SCIENTIFIC JOURNAL ISSN 2710-0766  
«COMPUTER SYSTEMS AND INFORMATION TECHNOLOGIES»

<https://doi.org/10.11891/issn-2024-4-5>  
UDC 004.9

Tymur ISAIEV, Tetiana KYSEL  
Khmelnytskyi National University

## METHOD OF CREATING CUSTOM DATASET TO TRAIN CONVOLUTIONAL NEURAL NETWORK

*The task of creating and developing custom datasets for training convolutional neural networks (CNNs) is essential due to the increasing adoption of deep learning across industries. CNNs have become fundamental tools for various applications, including computer vision, natural language processing, medical imaging, and autonomous systems. However, the success of a CNN depends heavily on the quality and relevance of the data it is trained on. The datasets used to train these models must be diverse, representative of the task at hand, and of sufficient quality to capture the underlying patterns that the CNN needs to learn. Thus, building custom datasets that align with the specific objectives of a neural network plays a critical role in enhancing the performance and generalization capability of the trained model.*

*This paper focuses on developing a method and subsystem for generating high-quality custom datasets tailored to CNNs. The aim is to provide a framework that automates and streamlines the processes involved in data collection, preprocessing, augmentation, annotation, and validation. Moreover, the method integrates tools that allow the dataset to evolve over time, incorporating new data to adapt to changing requirements or environments, making the system flexible and scalable.*

*The process of creating a dataset begins with the acquisition of raw data. The data can come from various sources such as images from cameras, videos, sensor feeds, open data repositories, or proprietary datasets. A key consideration during data collection is ensuring that the samples cover the full range of conditions or classes the CNN will encounter in production. For example, in an object recognition task, it is essential to collect images from diverse environments, lighting conditions, and angles to train the model effectively. Ensuring variability in the dataset increases the model's ability to generalize, reducing the risk of poor performance on unseen data.*

*Data augmentation is a critical step in building a robust dataset, particularly when the size of the dataset is limited. Augmentation techniques introduce variability into the dataset by artificially modifying the existing samples, thereby simulating a wider range of conditions. This helps the CNN generalize better and prevents overfitting. In essence, it allows the model to experience different perspectives and distortions of the same data, strengthening its adaptability to real-world scenarios.*

*Annotation involves labeling the data samples with the correct class or category information. Depending on the task, annotations may include bounding boxes for object detection, segmentation masks for semantic segmentation, or class labels for classification tasks. The importance of well-annotated data cannot be overstated, as CNNs rely on this labeled information to understand the relationships between input data and the desired output predictions.*

*Keywords: CNN, dataset, neural network, Roboflow, data preprocessing, data augmentation, labeling.*

Тимур ІСАЄВ, Тетяна КИСІЛЬ  
Хмельницький національний університет

## МЕТОД СТВОРЕННЯ СПЕЦІАЛІЗОВАНОГО НАБОРУ ДАНИХ ДЛЯ ТРЕНУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

*Завдання зі створення та розробки спеціальних наборів даних для навчання згорткових нейронних мереж (CNN) є надзвичайно важливим через зростаюче використання глибокого навчання в різних галузях. CNN стали основними інструментами для багатьох застосувань, включаючи комп'ютерний зір, обробку природної мови, медичну візуалізацію та автономні системи. Однак успіх CNN значною мірою залежить від якості та релевантності даних, на яких вона навчається. Набори даних для навчання цих моделей повинні бути різноманітними, відповідати специфіці завдання та мати достатню якість, щоб заохотити приховані патерни, які модель має спанувати. Таким чином, створення спеціальних наборів даних, що відповідають конкретним цілям нейронної мережі, відіграє ключову роль у покращенні ефективності та здатності моделі до узагальнення. Ця робота зосереджена на розробці методу та підсистеми для створення високісних спеціалізованих наборів даних для CNN. Метою є надання структури, яка автоматизує та спрощує процес збору даних, їх попередньої обробки, аугментації, анотації та валідації. Більше того, цей метод включає інструменти, які дозволяють датасету еволюціонувати з часом, інтегруючи нові дані для адаптації до змінних вимог або умов середовища, що робить систему гнучкою та масштабованою.*

*Процес створення набору даних починається з отримання сирих даних. Дані можуть надходити з різних джерел, таких як зображення з камер, відео, сенсорні потоки, відкриті репозиторії даних або власні корпоративні датасети. Важливим аспектом під час збору є забезпечення того, щоб вибірка охоплювала весь спектр умов або класів, з якими CNN зустрічатиметься під час експлуатації. Наприклад, у завданні розпізнавання об'єктів важливо зібрати зображення з різних середовищ, при різному освітленні та під різними кутами для ефективного навчання моделі. Забезпечення різноманітності у датасеті підвищує здатність моделі до узагальнення та знижує ризик поганого результату на невідомих даних.*

*Аугментація даних є критичним кроком у створенні надійного датасету, особливо коли його розмір обмежений. Техніки аугментації вносять різноманітність у вибірку, штучно модифікуючи наявні зображення, імітуючи ширший діапазон умов. Це допомагає CNN краще узагальнювати та запобігти перенавчанню. По суті, це дозволяє моделі виробувувати різні перспективи й спотворення одних і тих самих даних, звичайно ж здатність адаптуватися до реальних сценаріїв.*

*Анотації передбачає призначення міток зразкам даних із правильною класовою або категорійною інформацією. Залежно від завдання, анотації можуть включати рамки для вивілення об'єктів, маски сегментації для семантичної сегментації або класові мітки для класифікації. Важливість якісно анотованих даних важко переоцінити, адже CNN покладаються на ці мітки для розуміння взаємозв'язків між відносами даними та очікуваними прогнозами.*

*Ключові слова: CNN, набір даних, нейронна мережа, Roboflow, попередня обробка даних, доповнення даних, маркування.*

МЕЖНАРОДНИЙ НАУКОВИЙ ЖУРНАЛ  
«КОМП'ЮТЕРНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ», 2024, № 4

37

### Introduction

In recent years, there has been a rapid increase in the need to create custom datasets for training convolutional neural networks (CNNs) to solve practical tasks in various domains. This growth is largely driven by the availability of frameworks that allow users to design and train neural networks without requiring deep mathematical expertise [1-3]. Therefore, building custom datasets to meet specific objectives has become an essential task in the development of neural network models, enabling these systems to perform accurately across diverse applications.

The goal of this work is to develop a method for creating custom datasets and to design a system that optimizes the training process of convolutional neural networks.

To achieve this goal, the following tasks must be done:

- create a training dataset tailored to the specific neural network task;
- design and test a CNN model using the custom dataset;
- implement the dataset creation pipeline in a scalable and automated system.

Custom datasets are fundamental in training neural networks, as the quality of a model's performance depends heavily on the data it learns from. Each dataset must contain samples that represent the real-world conditions the model will encounter. Depending on the domain, these datasets may include images, videos, or sensor data tailored to tasks such as object recognition, sentiment analysis, or medical diagnosis.

In the field of computer vision, datasets often consist of labeled images that help neural networks learn to detect patterns. Creating such datasets involves not only data collection but also careful preprocessing, augmentation, and annotation to ensure the samples are both varied and representative. This ensures that the trained model generalizes well and performs accurately when exposed to new, unseen data.

One widely-used tool for building datasets is Roboflow, a platform that simplifies the process of dataset creation, management, and deployment. Roboflow allows developers to easily collect, organize, and label data for neural networks. It also offers features for data augmentation, such as adjusting brightness, contrast, or rotation, which helps increase dataset diversity and prevents overfitting during model training.

Key Components of the Dataset Creation Method:

- 1) data collection: Collecting raw data from relevant sources such as cameras, public repositories, or proprietary databases. This step ensures that the dataset reflects the conditions and scenarios the neural network will encounter;
- 2) preprocessing and augmentation: Applying transformations to improve data quality and simulate different real-world conditions. With Roboflow's tools, images can be scaled, rotated, flipped, or have noise added, making the dataset more robust and varied;
- 3) annotation and labeling: Tagging each sample with the appropriate class labels. For tasks like object detection or emotion recognition, this could involve drawing bounding boxes or assigning specific categories to images. Accurate labeling is crucial for the CNN to learn meaningful relationships between inputs and outputs.

A custom dataset must be diverse and well-balanced to avoid bias. For example, if one class of images is overrepresented, the CNN may learn to favor that class, resulting in poor performance on other categories. Roboflow helps address this issue by tracking class distribution and suggesting ways to balance the dataset. This ensures the model generalizes well across all categories [7-8].

Creating custom datasets tailored to specific CNN tasks plays a vital role in ensuring the success of neural network models. By using Roboflow, developers can streamline the data collection, preprocessing, and annotation processes, leading to more efficient and scalable workflows. This method not only ensures higher accuracy in model predictions but also enhances the network's ability to generalize across different environments and tasks. As deep learning technologies continue to evolve, the demand for high-quality, task-specific datasets will remain a critical factor in the development of intelligent systems.

One of the key advantages of Convolutional Neural Networks (CNNs) is their ability to automatically extract relevant features from data. This means that instead of requiring a manual selection of features or a detailed analysis of how the input variables are related, CNNs learn to identify the most important patterns and relationships by themselves during training.

For example, in traditional machine learning approaches, a lot of time and effort might be spent analyzing data to understand which characteristics (or features) are most important. In image classification, this could mean manually deciding if color, shape, or texture should be prioritized. With CNNs, this step is not necessary. The network uses its layers to progressively learn which features matter most for the task at hand.

Because of this capability, CNNs reduce the need for a pre-analysis of data correlations. Instead of manually identifying what might be important, the network "learns" by itself through exposure to the training data. This automatic feature extraction saves time, minimizes human error, and often leads to better model performance since CNNs can uncover subtle patterns that may not be immediately obvious to a person.

#### Analysis of Existing Solutions

The increasing integration of technology in various fields has led to the emergence of numerous online platforms for compiling and sharing diverse datasets, greatly enhancing accessibility and participation for researchers globally. The rise of artificial intelligence, particularly in the domain of image recognition, has made it feasible to automate processes such as the classification of objects, including ancient artifacts. This advancement has prompted the exploration of machine learning techniques for various applications, including the study of historical items, artworks, and even natural specimens. Currently, automating the classification of ancient artifacts often serves more as an academic pursuit than a practical solution due to several key factors:

- 1) many existing datasets are limited in size, with most models requiring thousands of samples to achieve reliable accuracy. Datasets with fewer than 1,000 images often lead to overfitting, where the model learns noise rather than the underlying patterns;
- 2) no single method guarantees complete accuracy in classification. This uncertainty is compounded by the variability in the quality of images, including differences in lighting, angles, and backgrounds, which can adversely affect the model's performance;
- 3) many models struggle with recognizing items that were not included in their training sets. This limitation highlights the importance of diverse training data that encompasses various types, conditions, and contexts in which artifacts may appear;
- 4) the need for such systems is not commercially driven; collectors or museums often prefer expert analysis over machine learning solutions that still require human verification. This reliance on expertise underscores the complexities involved in understanding the historical context and significance of each item, which cannot always be captured by algorithms.

Despite these challenges, several innovative systems have been developed for classifying various artifacts. For example, a research group from the University of Tokyo designed a CNN specifically for categorizing ancient pottery shards. Their dataset comprised over 15,000 images of pottery, organized into multiple classes based on historical significance, material, and decorative styles. The team employed the VGG-16 architecture, utilizing transfer learning to leverage pre-trained weights. After 150 epochs of training, they achieved an impressive classification accuracy of 94%, demonstrating the effectiveness of their approach and contributing valuable insights into the pottery's historical context [21].

In another instance, researchers from Stanford University created a classification model for identifying different species of ancient coins, combining state-of-the-art machine learning with archaeology. They compiled an extensive dataset of 20,000 high-resolution images of coins from various eras, meticulously categorized by denomination, material, and design features. This dataset formed the backbone of their project, enabling the team to train their model using the EfficientNet architecture, chosen for its optimal balance between computational efficiency and accuracy. EfficientNet, with its scalable design, allowed the researchers to efficiently process and analyze the intricate details of each coin, such as inscriptions and fine engravings. Their model achieved a classification accuracy of up to 90%, showcasing how advanced deep learning techniques can provide valuable tools for traditional fields like archaeology. This breakthrough not only aids historians and numismatists in identifying and cataloging ancient coins but also serves as a template for similar studies in cultural preservation, facilitating the digitization and analysis of other historical artifacts [22].

Furthermore, a team of data scientists in Canada demonstrated the practicality of convolutional neural networks (CNNs) in modern financial systems. Their project focused on identifying and classifying contemporary currency notes, addressing challenges like fraud detection and authenticity verification. Using a dataset of 10,000 images of various currency denominations, they employed techniques such as data augmentation, normalization, and the generation of synthetic data through computer graphics to enhance the dataset's diversity and robustness. Data augmentation introduced variations in lighting, angles, and backgrounds, simulating real-world conditions. As a result, their CNN achieved an impressive 97% accuracy on validation sets, highlighting the potential of deep learning to improve reliability and efficiency in applications requiring high precision, such as banking and security [23].

Similarly, a notable project on the collaborative platform Medium focused on developing a mobile application aimed at helping collectors identify rare and valuable coins. This project adopted a user-centric approach, relying on community contributions to build a dynamic dataset. Users uploaded images of their collections, which were then annotated to create a growing dataset tailored to the recognition of rare editions and minting errors. The lightweight CNN architecture powering the app was designed to operate efficiently on mobile devices, balancing computational constraints with robust performance. With a preliminary classification accuracy of 92%, the application not only provides immediate value to collectors but also encourages active user participation, enriching the dataset over time and ensuring the model remains relevant and effective.

Another innovative approach gaining momentum is the use of generative adversarial networks (GANs) to augment limited datasets. GANs, with their ability to generate realistic synthetic data, are proving to be a game-changer for artifact classification tasks. Researchers have employed GANs to create high-quality images of artifacts that mimic real-world conditions, such as varying levels of wear and lighting. This synthetic data complements existing datasets, allowing models to learn from a broader range of inputs and reducing the risks of overfitting.

Preliminary studies using GAN-augmented datasets have reported significant improvements in classification accuracy, as models trained on these datasets demonstrate enhanced generalization to unseen data. This method has the potential to revolutionize the field, particularly for niche applications where acquiring large volumes of real-world data is challenging.

In summary, the field of artifact classification through deep learning is experiencing rapid advancements, but challenges remain. Current research underscores the importance of tailored approaches, with researchers leveraging standard CNN architectures, efficient preprocessing techniques, and cutting-edge innovations like GANs to achieve high performance. However, the lack of a universal solution highlights the need for continued experimentation and refinement. The integration of diverse, high-quality datasets and the exploration of novel methodologies will play a crucial role in overcoming existing limitations. As these technologies mature, their application promises to not only automate classification tasks but also deepen our understanding of cultural artifacts, creating opportunities to preserve and appreciate the richness of human history.

#### Creating Dataset using Roboflow to train custom model

Creating a custom dataset using Roboflow for an emotions dataset involves a series of methodical steps, from data collection to model training:

- 1) define objectives;
- 2) data collection;
- 3) organize images;
- 4) create or log into Roboflow account;
- 5) set up a new project;
- 6) click on "New Project," provide a project name, and select the appropriate project type ;
- 7) upload images;
- 8) label images;
- 9) apply data augmentation;
- 10) export the dataset;
- 11) download additional annotation files, such as CSVs, for better class training

Initially, first step is to define the objectives of the project. It is necessary to determine the purpose of the emotions dataset, specifically whether to classify emotions in images, videos, or text. For the purposes of this example, it was decided to focus on images representing various emotions, such as happiness, sadness, anger, and surprise.

Next, step is to gather a collection of images that depict these different emotions. During this project it was decided to use authors' own images by taking screenshots from webcam to gather various images. It is essential to ensure that there is a balanced number of images for each emotion class to avoid bias in the model.

Once the step of collecting images is done, it is essential to organize them into folders, each labeled with the corresponding emotion. For instance, creating folders named "happy," "sad," "angry," "surprised," and "neutral." This systematic organization facilitates the subsequent labeling process.

After organizing the images, we need to create a Roboflow account by visiting their website and signing up, if we do not already possess an account. Once logged in, click on the "New Project" button, choosing a project name and setting the type of project to "Image Classification," as this aligns with our current project on emotions dataset (Fig. 1).

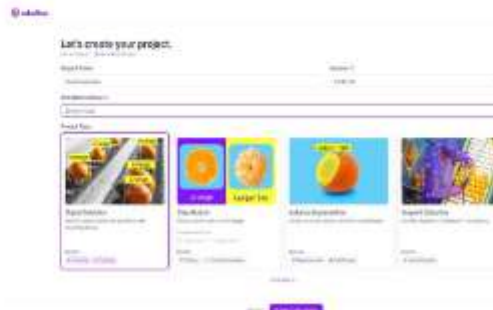


Fig. 1. Settings for creating dataset in Roboflow service

In the project dashboard, find the option to upload images. We can either drag and drop my image folders or use the upload button to select the folders. Roboflow supports bulk uploads, allowing us to upload all the images simultaneously (Fig. 2).

Once the images have been uploaded, proceed to the labeling phase. Roboflow provides tools for labeling images, enabling users to assign labels to each image based on the emotion it represents. This step is critical, as it defines the categories from which the model will learn (Fig. 3).

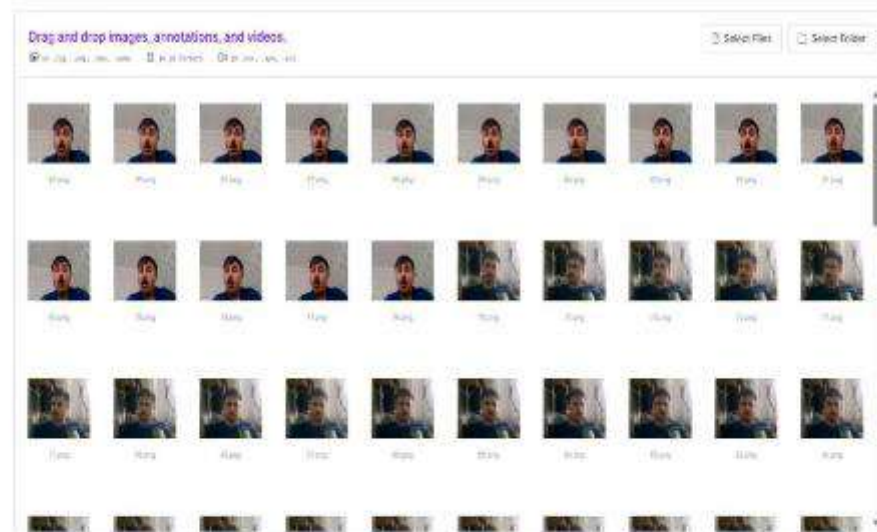


Fig. 2. Process of downloading images to Roboflow



Fig. 3. Process of labeling images

After all images are labeled, advance to the dataset augmentation phase. Roboflow offers various augmentation techniques, such as rotation, flipping, and color adjustments. These techniques enhance the robustness of model, used in this paper, by increasing the diversity of the training data (Fig. 4).

Once we have applied the desired augmentations, we can export the dataset. Roboflow allows users to choose the format for exporting, such as TensorFlow, PyTorch, or YOLO. Users can select the format that is compatible with the framework they intend to use for training custom model model (Fig. 5).

Additionally we can download marking file to help model better determine classes and help it to train more accurately (Fig. 6).

With the dataset exported, the training process begins. Users can import the dataset into their machine learning framework, set up the model architecture, and configure the training parameters. During training, the model learns to classify emotions based on the labeled images that was created using Roboflow and custom method.

Finally, after completing the training, next step is to evaluate the model's performance on a separate validation set to ensure it generalizes well to unseen data. Depending on the results, users may need to fine-tune the model or gather additional data to improve its accuracy. This process of creating a custom dataset using Roboflow

provides a streamlined approach to preparing data for machine learning tasks focused on emotion recognition. Additionally, it is very easy to add new images to the dataset using Roboflow platform.

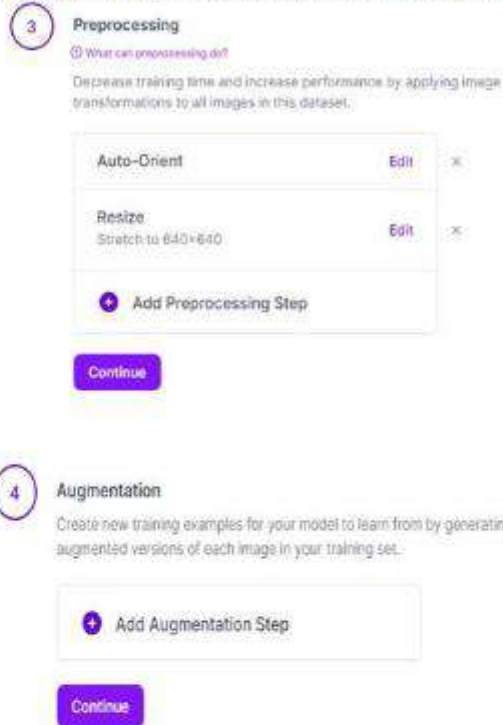


Fig. 4. Augmentation settings of Dataset



Fig. 5. Process of choosing model to download custom dataset



5. Taberidoost H. Data collection methods and tools for research: a step-by-step guide to choose data collection technique for academic and business research projects. *International Journal of Academic Research in Management (IJARM)*. 2021. Vol. 10, no. 1. Pp. 30–38.
6. Ribeiro-Navarrete S., Saura J. R., Palacios-Marqués D. Towards a new era of mass data collection: Assessing pandemic surveillance technologies to preserve user privacy. *Technological Forecasting and Social Change*. 2021. Vol. 167. P. 120681.
7. Olaniyi O. O., Okunleye O. J., Olobunji S. O. Advancing data-driven decision-making in smart cities through big data analytics: A comprehensive review of existing literature. *Current Journal of Applied Science and Technology*. 2023. Vol. 42, no. 25. Pp. 10–18.
8. Sun Z., Ma W., Yang Y., Li H., Zhang L. A Novel Efficient Data Gathering Algorithm for Disconnected Sensor Networks Based on Mobile Edge Computing. *Wireless Communications and Mobile Computing*. 2022. Vol. 2022. P. 4763153.
9. Sharma N. K. Instruments used in the collection of data in research. Available at SSRN 4138751, 2022.
10. Röbiger P., Gurevich M., Scintua F., Duran A. Two contrasting data annotation paradigms for subjective NLP tasks. arXiv preprint arXiv:2112.07475, 2021.
11. Greenwald N. F., B. K. A., B. J. W., K. H., G. M., N. L., M. C. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature Biotechnology*. 2022. Vol. 40, no. 4. Pp. 555–565.
12. Wang X., Yu Y., Wu Y., Zhang L., Li Y. A Structure-Guided Molecular Network Strategy for Global Untargeted Metabolomics Data Annotation. *Analytical Chemistry*. 2023. Vol. 95, no. 31. Pp. 11603–11612.
13. Mamit N., Alnubaidi M., Abdul Hamid S., Ibrahim N., Rahman A. Enhancing image annotation technique of fruit classification using a deep learning approach. *Sustainability*. 2023. Vol. 15, no. 2. P. 901.
14. Wang S., Cheng X., Zhang H., Shen J., Yu Y. Annotation-efficient deep learning for automatic medical image segmentation. *Nature Communications*. 2021. Vol. 12, no. 1. P. 5915.
15. Sereñ A., Turban C., Sengul G. A Hybrid Approach for Semantic Image Annotation. *IJEE Access*. 2021. Vol. 9. Pp. 131977–131984.
16. Reiß S., O. M., L. H., R. H. Every annotation counts: Multi-label deep supervision for medical image segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. Pp. 9532–9542.
17. Sager C., Janiesch C., Zschech P. A survey of image labelling for computer vision applications. *Journal of Business Analytics*. 2021. Vol. 4, no. 2. Pp. 91–110.
18. Niu C., Shan H., Wang G. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*. 2022. Vol. 31. Pp. 7264–7278.
19. Arzi M., Latif S., H. P., J. H. LABKIT: labeling and segmentation toolkit for big image data. *Frontiers in Computer Science*. 2022. Vol. 4. P. 777728.
20. Roboflow main page. URL: <https://roboflow.com>
21. VGG16 Project. URL: <https://neurohive.io/en/popular-networks/vgg16/>
22. Stanford University project. URL: <https://cr231n.stanford.edu>

<b>Тимур Ісаєв</b> <b>Tymur Isayev</b>	Master Student of Computer Engineering & Information Systems Department, Khmelnytskyi National University E-mail: <a href="mailto:tymur1112@gmail.com">tymur1112@gmail.com</a>	Магістрант кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет
<b>Тетяна Кувіл</b> <b>Tetyana Kuvil</b>	Candidate of Physical and Mathematical Sciences, Associate Professor of Computer Engineering & Information Systems Department, Khmelnytskyi National University <a href="https://orcid.org/0000-0002-4094-3500">https://orcid.org/0000-0002-4094-3500</a> e-mail: <a href="mailto:kuvil_taty@ukr.net">kuvil_taty@ukr.net</a>	Кандидат фізико-математичних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем, Хмельницький національний університет

### Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Тимур ІСАЄВ

Співавтор:

Назва: ІСАЄВ\_Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:3%

Коефіцієнт подібності 2:0.6%

Мікропробіли: 33

Заміна букв: 1

Інтервали: 0

Блі знаки: 1

Дата створення звіту: 2025-04-16 18:01:38.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-04-16

Доцент Андрій Нічепорук

Дата

експерт

## Anti-Plagiarism v-15.260 Educational

Максимальне співпадіння з одним документом 24.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 13%

ID: 231960 Назва: МКР Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору Додано в БД: 2025-04-16 Автора: Тимур ІСАЄВ Керівники: Тетяна КИСІЛЬ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	88009	804	21478 (24%)	206 (26%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
187516	Назва: Звіт ПДП Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору Додано в БД: 2025-03-05 Автора: Т. С. Ісаєва Керівники: Кисіль Т.М. Консультанти: Опоненти:	20958 (24.0%)	196 (24.0%)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Тимур ІСАЄВ

Тема: Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 84

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод розпізнавання емоцій людини у реальному часі, який буде працювати на малопотужних пристроях

2. Висновок про відповідність роботи дипломному завданню \_\_\_\_\_  
Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі було розглянуто існуючі рішення у методах розпізнавання емоцій. Було досліджено сучасні методи розпізнавання емоцій з використанням машинного навчання та комп'ютерного зору. Крім того, було проаналізовано найсучасніші тенденції у розвитку методів з використанням штучного інтелекту.

У другому розділі було проаналізовано методи корекції помилок під час розпізнавання емоцій. Також, було більш детально порівняно наявні сучасні моделі, які підходять для цього дослідження та аналіз їх переваг та недоліків у роботі. Було розглянуто усі необхідні компоненти для коректної роботи моделі та підготовка усіх бібліотек для розпізнавання емоцій. Також, було досліджено датасет, який було використано у цьому дослідженні.

У третьому розділі було розроблено та представлено метод та алгоритм тренування нейронної мережі для розпізнавання емоцій у реальному часі. Під час роботи над цим розділом було додатково створено власний датасет з використання

персональних фото для покращення методів навчання моделі. Також, було розглянуто усі необхідні бібліотеки для правильної роботи моделі.

У четвертому розділі було проаналізовано отримані у третьому розділі результати шляхом перевірки моделі у точності розпізнавання емоцій з датасету та з використанням комп'ютерного зору. Також, було досліджено спосіб покращення результатів моделі шляхом підключення прискорювача та аналіз отриманих результатів шляхом побудови графів.

4. Позитивні сторони роботи: Розроблений метод розпізнавання емоцій у реальному часі дозволяє швидко та ефективно визначати емоційний стан людини. Це корисно для медицини, освіти та безпеки. Оптимізація моделі для мобільних пристроїв забезпечує її ефективну роботу навіть на пристроях з обмеженою продуктивністю. Створений власний набір даних дозволяє покращити точність розпізнавання емоцій та адаптувати модель до різних умов. Інтеграція моделі на платформі Raspberry Pi підтвердила ефективність методу та можливість його застосування на пристроях із обмеженими потужностями.

5. Негативні сторони роботи: Відсутні.

6. Оцінка графічного оформлення та пояснювальної записки роботи: =

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: =

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «відмінно» 5.00 (A)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)

Мартинюк Валерій Володимирович, зав. каф.  
АКІТ-ЯР, ХНУ



Завідувачу кафедри КПС  
доктору філософії, доценту  
Ользі ПАВЛОВІЙ

Ісаєва Тимура Сергійовича

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІЗм-23-1

#### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (StrikePlagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.04.2025р

**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору

Автор: Ісаєв Тимур Сергійович

Спеціальність: 123 – Комп'ютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Кисіль Тетяна Миколаївна, к.ф.м.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) запозичення розміщені в розділах є збігом зі звітом з науково-дослідної практики автора Ісаєва Тимура Сергійовича "Метод розпізнавання емоцій людини у реальному часі за допомогою комп'ютерного зору", який було додано в репозитрій ХНУ 5 березня 2025 року;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживими фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі крайномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 3% і адресується до 30 першоджерел; та системою Anti-Plagiarism складає 24%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи \_\_\_\_\_

Тетяна КИСІЛЬ

Гарант ОП \_\_\_\_\_

Олег САВЕНКО

Завідувач кафедри КІС \_\_\_\_\_

Ольга ПАВЛОВА