

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

*«Розробка програмного забезпечення інтелектуальної
системи організації роботи підприємства»*

Студентки 2м курсу, бз групи,
спеціальності 121 «Інженерія
програмного забезпечення»,
спеціалізації «Інженерія
програмного забезпечення»

підпис студента

Опенько
Дар'ї Павлівни

Науковий керівник
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

Гарант освітньої програми
доктор технічних наук,
професор кафедри інженерії
програмного забезпечення та
кібербезпеки

підпис керівника

Криворучко Олена
Володимирівна

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь магістр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

8 листопада 2019 р.

Завдання на випускний кваліфікаційний проект студентіві

Опенько Дар'ї Павлівні
(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проекту: «Розробка програмного
забезпечення інтелектуальної системи організації роботи підприємства»

Затверджена наказом ректора від "24" грудня 2019 р. № 4440

2. Строк здачі студентом закінченого проекту 01 грудня 2020 р.

3. Цільова установка та вихідні дані до проекту

Мета проекту: дослідження розвитку сучасних методів ідентифікації осіб та їх
ефективного застосування у створенні власного програмного рішення.

Об'єкт дослідження: інтелектуальна система.

Предмет дослідження: точність та ефективність алгоритмів комп'ютерного
бачення для ідентифікації особи по фото та відео.

4. Консультанти проекту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)
ВСТУП

РОЗДІЛ 1. ОСНОВИ РОБОТИ БІОМЕТРИЧНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ

1.1. Огляд інформаційного середовища

1.2. Змістовний опис і аналіз предметної області

1.3. Аналіз успішних програмних рішень

1.4. Формування технічного завдання

1.5. Висновки до розділу 1

РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Розробка структури та загальної концепції побудови

2.2. Конструювання програмного забезпечення

2.2.1. Методологія та підходи до розробки програмного забезпечення

2.2.2. Використані технології

2.3. Висновки до розділу 2

РОЗДІЛ 3. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

3.1. Узагальнена архітектура мережі

3.2. Модель розпізнавання та ідентифікації

3.2.1. Ідентифікація обличчя

3.2.2. Розпізнавання статі та віку

3.2.3. Розпізнавання емоційного стану

3.3. Реалізація зберігання даних

3.4. Висновки до розділу 3

РОЗДІЛ 4. АНАЛІЗ ТА РОЗГОРТАННЯ ПРОГРАМНОГО ПРОДУКТУ

4.1. Перевірка ефективності розробки

4.2. Розгортання програмного продукту

4.3 Висновки до розділу 4

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ТЕХНІЧНЕ ЗАВДАННЯ

ДОДАТКИ

6. Календарний план виконання проекту

№ пор.	Назва етапів випускного кваліфікаційного проекту	Строк виконання етапів проекту	
		за планом	фактично
1.	Вибір теми випускної кваліфікаційної роботи	20.09.2019	20.09.2019
2.	Розробка та затвердження завдання на проект магістра	08.11.2019	08.11.2019
3.	Вступ та перелік літературних джерел	24.01.2020	24.01.2020
4.	Наукова стаття	01.09.2020	01.09.2020
5.	Технічне завдання (ТЗ)	28.02.2020	28.02.2020
6.	Розділ 1. Основи роботи біометричної системи розпізнавання	25.06.2020	25.06.2020
7.	Розділ 2. Моделювання та конструювання програмного забезпечення	07.09.2020	07.09.2020
8.	Розділ 3. Архітектура нейронних мереж	19.10.2020	19.10.2020
9.	Розділ 4. Аналіз та розгортання програмного забезпечення	20.10.2020	20.10.2020
10.	Програма та методика тестування	21.10.2020	21.10.2020
11.	Керівництво користувача	23.10.2020	23.10.2020
12.	Висновки та пропозиції	30.10.2020	30.10.2020
13.	Здача випускного кваліфікаційного проекту на кафедрі	05.11.2020	05.11.2020
14.	Підготовка автореферату та презентації доповіді	05.11.2020	05.11.2020
15.	Попередній захист випускного кваліфікаційного проекту	25.11.2020-27.11.2020	25.11.2020-27.11.2020
16.	Зовнішнє рецензування випускного кваліфікаційного проекту	01.12.2020	01.12.2020
17.	Підготовка до публічного захисту випускного кваліфікаційного проекту	10.12.2020-11.12.2020	

7. Дата видачі завдання «8» листопада 2019 р.

8. Науковий керівник випускного кваліфікаційного проекту

Криворучко О. В.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми

Криворучко О. В.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент

Опенько Д. П.

(прізвище, ініціали, підпис)

11. Відгук керівника випускного кваліфікаційного проекту

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Науковий керівник випускного кваліфікаційного проекту _____
(підпис, дата)

Відмітка про попередній захист _____ Криворучко О. В.
(ПІБ, підпис, дата)

12. Висновок про випускний кваліфікаційний проект

Випускний кваліфікаційний проект студента Опенько Д. П.
(прізвище, ініціали)

може бути допущена до захисту екзаменаційній комісії.

Гарант освітньої програми Криворучко О. В.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« » 20 p.

Анотація

Опенько Д. П. «Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства»

У випускній кваліфікаційній роботі розглядаються принципи та особливості розробки та створення системи для організації роботи підприємства, якою виступає система ідентифікації та трекінгу особи, етапи її розробки, формування технічного завдання та проектування, а також програмування на мові високого рівня Python та використання баз даних для отримання інформації.

Ключові слова: КОМП'ЮТЕРНЕ БАЧЕННЯ, АЛГОРИТМ, РОЗПІЗНАВАННЯ, КЛАСИФІКАЦІЯ, НЕЙРОННІ МЕРЕЖІ, МАШИННЕ НАВЧАННЯ.

Annotation

Openko D.P. " Software development of an intelligent system for organizing the work of the enterprise "

The final qualification work considers the principles and features of development and creation of a system for organizing the work of the enterprise, which is a system of identification and tracking of a person, stages of its development, formation of terms of reference and design, as well as programming in high-level Python and use of databases.

Keywords: COMPUTER VISION, ALGORITHM, RECOGNITION, CLASSIFICATION, NEURAL NETWORKS, MACHINE LEARNING.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ПЗ – програмне забезпечення.

Машинне навчання (англ. machine learning) – це підгалузь інформатики (зокрема, м'яких та гранульованих обчислень), яка еволюціювала з дослідження розпізнавання образів та теорії обчислювального навчання в галузі штучного інтелекту.

CNN – Convolutional Neural Network(s).

DCNN – Deep Convolutional Neural Network.

Object detection – розпізнавання образів.

Data Mining – виявлення прихованих закономірностей або взаємозв'язків між змінними у великих масивах необроблених даних. Зазвичай поділяють на задачі класифікації, моделювання та прогнозування.

Бекбоун – функціональна будова та структура зв'язків між шарами нейронної мережі.

API – набір процедур та функцій, що дозволяють створювати програмні продукти з доступом до внутрішнього функціоналу операційної системи чи пристрою.

Розпізнавання образів (pattern recognition) – це розділ теорії штучного інтелекту (artificial intelligence), що вивчає методи класифікації об'єктів.

					КНТЕУ 121 063-09.МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства	Стадія	Арку	Аркушів
Зав. каф.	Криворучко О.В.			19.10.20		ПС	2	49
Керівник	Криворучко О.В.			19.10.20		Факультет інформаційних технологій 2м курс, 6з група		
Гарант	Криворучко О.В.			19.10.20				
Розробив	Опенько Д.П.			19.10.20				
					Перелік умовних скорочень			

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ОСНОВИ РОБОТИ БІОМЕТРИЧНОЇ СИСТЕМИ	
РОЗПІЗНАВАННЯ.....	7
1.1. Огляд інформаційного середовища	7
1.2. Змістовний опис і аналіз предметної області	10
1.3. Аналіз успішних програмних рішень	13
1.4. Формування технічного завдання	Error! Bookmark not defined.
1.5. Висновки до розділу 1	Error! Bookmark not defined.
РОЗДІЛ 2. МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ.....	19
2.1. Розробка структури та загальної концепції побудови	19
2.2. Конструювання програмного забезпечення.....	23
2.2.1. Методологія та підходи до розробки програмного забезпечення	23
2.2.2. Використані технології	24
2.3. Висновки до розділу 2	25
РОЗДІЛ 3. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ	26
3.1. Узагальнена архітектура мережі	26
3.2. Модель розпізнавання та ідентифікації.....	28
3.2.1. Ідентифікація обличчя.....	30
3.2.2. Розпізнавання статі та віку.....	30
3.2.3. Розпізнавання емоційного стану	32
3.3. Реалізація зберігання даних.....	33
3.4. Висновки до розділу 3	35
РОЗДІЛ 4. АНАЛІЗ ТА РОЗГОРТАННЯ ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ.....	36
4.1. Перевірка ефективності розробки.....	36
4.2. Розгортання програмного продукту	40
4.3. Висновки до розділу 4	44
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	Error! Bookmark not defined.
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	Error! Bookmark not defined.
ТЕХНІЧЕ ЗАВДАННЯ	Error! Bookmark not defined.
ДОДАТКИ.....	58

					<i>КНТЕУ 121 063-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			19.10.20	Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства	Стадія	Арку	Аркуш
Керівник	Криворучко О.В.			19.10.20		Зміст	3	49
Гарант	Криворучко О.В.			19.10.20		Факультет інформаційних технологій 2м курс, бз група		
Розробив	Опенько Д.П.			19.10.20				
					Зміст			

ВСТУП

Актуальність. В умовах глобалізації ми все більше використовуємо технічні та програмні засоби для реалізації власних потреб. Іде поступова заміна понять та перерозподіл ролей відповідальності людей та машин.

Звичайно, якою би чудовою не була програмна розробка, у більшості випадків, все ще потрібний нагляд людини за коректністю роботи та проведення аналізу отриманих результатів, розробки подальшого плану дій.

Програми та системи вже вміють самі навчатися на власному досвіді з набутих даних, проте людський мозок є більш досконалішою системою.

Тому, головною цінністю будь-якої кампанії є люди, які там працюють.

Аби мати змогу гарно організувати робочий процес на підприємстві та контролювати його, потрібно було би гарно мати зручно дані про працівників, як от: їх автоматична ідентифікація у просторі. При звірці з часом, це дозволить:

- контролювати виконання робочих завдань;
- аналізувати робочий час людей, чи працівник належно ставиться до своїх обов'язків (працює), чи займається іншими справами не у свою перерву;
- мати контроль доступу до спеціальних приміщень аби не було змоги потрапити, наприклад, до складу чи кабінету начальника непоміченим та можна було встановити особу.

Це дозволить вчасно прийняти міри у випадку необхідності, навчить бути більш обізнаним у своїх працівниках та їх звичках, та краще контролювати робочий процес.

					<i>КНТЕУ 121 063-09.МР</i>			
					<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	<i>Стадія</i>	<i>Арку</i>	<i>Аркуш</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>В</i>	<i>4</i>	<i>49</i>
<i>Зав. каф.</i>	<i>Криворучко О.В.</i>			<i>19.10.20</i>		<i>Факультет інформаційних технологій 2м курс, бз група</i>		
<i>Керівник</i>	<i>Криворучко О.В.</i>			<i>19.10.20</i>				
<i>Гарант</i>	<i>Криворучко О.В.</i>			<i>19.10.20</i>				
<i>Розробив</i>	<i>Опенько Д.П.</i>			<i>19.10.20</i>	<i>Вступ</i>			

Розпізнавання осіб та їх ідентифікація були завжди потрібними, при невеликій кількості існуючих аналогів. Дана тематика є дуже актуальною у наш час, коли ще більше зростають вимоги до виконання роботи у більш сжаті терміни, проте на тому ж рівні, через кризу у всьому світі. Аби компанія могла залишитись на плаву та не втратити все.

Метою роботи є дослідження розвитку сучасних методів ідентифікації осіб та їх ефективного застосування у створенні власного успішного програмного рішення.

Об'єкт дослідження: інтелектуальна система.

Предмет дослідження: точність та ефективність алгоритмів комп'ютерного бачення для ідентифікації особи по фото та відео.

Для реалізації поставленої мети, було сформовано такі *задачі дослідження:*

- огляд існуючих способів розпізнавання осіб у громадській сфері;
- дослідження існуючих технічних способів ідентифікації осіб;
- підбір архітектури для побудови нейронної мережі моделі розпізнавання;
- підбір навчальних датасетів для тренування моделі розпізнавання;
- розробка програмного забезпечення, що буде використовувати спроектовану модель розпізнавання;

Наукова новизна дослідження: у результаті проведеної роботи було реалізовано унікальний програмний модуль для ідентифікації осіб та отримання їх біометричного шаблону за допомогою сучасних алгоритмів комп'ютерного бачення.

Методи дослідження : дослідження, аналіз, експеримент.

Практичне значення дослідження: реалізація наведеного в роботі програмного модулю та практичне впровадження його в систему організації

					КНТЕУ 121 063-09.МР	Аркуш
						5
Зм.	Аркуш	№ докум	Підпис	Дата		

роботи виробництва дозволить забезпечити своєчасне виявлення потенційно небезпечних осіб, в тому числі емоційно-психологічного стану робітників.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

РОЗДІЛ 1.

ОСНОВИ РОБОТИ БІОМЕТРИЧНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ

1.1. Огляд інформаційного середовища

У наш час найбільш поширеним методом перевірки особистості людини - є зіставлення рис обличчя і фото на документах (паспорт, водійські права) для встановлення особи. Порівняння особи на фото та вживу – це досить важке завдання, адже навіть натреновані у цьому люди помиляються досить часто. Відсоток "промаху" під час виконання ідентифікації по обличчю людиною є чималим: навіть в ідеальних умовах, де немає інших відволікаючих чинників, відсоток помилок склав 10-20% – і це при порівнянні фото, що зроблені в один день в умовах лабораторії .

Станом же на квітень 2020 року, кращий програмний алгоритм ідентифікації осіб має коефіцієнт похибки 0,08%. Проте існує різниця між розпізнаванням і порівнянням осіб (Face Comparison), незважаючи на те, що обидва підходи використовують одну і ту ж технологію. Розпізнавання осіб (Facial Recognition) має на увазі пошук збіги із зображенням по великій базі даних, його широко використовують в правоохоронних органах (для пошуку фотографій злочинців), в системах безпеки (наприклад, для верифікації співробітників), в банківській сфері (для запобігання шахрайських операцій), в споживчому секторі (для здійснення платежів в реальному часі). У разі програм для порівняння осіб, два фото одного і того ж людини порівнюються один з одним.

В основу роботи біометричної системи розпізнавання осіб закладений принцип побудови і порівняння математичних моделей особи, що дозволяє

					<i>КНТЕУ 121 06з-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		25.06.20		РІ	7	49
Керівник		Криворучко О.В.		25.06.20		Факультет інформаційних технологій 2м курс, 6з група		
Гарант		Криворучко О.В.		25.06.20				
Розробив		Опенько Д.П.		25.06.20	<i>Основи роботи біометричної системи розпізнавання</i>			

з високою точністю ідентифікувати або верифікувати особистість. Ці системи будують гіпотезу, чи збігається особа на одній фотографії з особою на іншій, незалежно від його вираження і інших атрибутів, таких як вік, волосся на обличчі, аксесуари і так далі.

До того ж, камери здатні захопити зображення на відстані, що ідеально підходить для побудови систем моніторингу та прихованої ідентифікації (безконтактна біометрія).

Незалежно від технології, що використовується, програми для розпізнавання осіб працюють в кілька етапів:

Етап 1. Виявлення особи. Система автоматично визначає обличчя людини на зображенні (скан фотографії або кадр відеопотоку). Виявлення (детекція) особи є вкрай важливим етапом розпізнавання, без якого неможлива подальша ідентифікація.

Етап 2. Аналіз особи. Особа людини містить 80 нодальних точок. Програма для розпізнавання осіб може зчитувати геометрію цих орієнтирів, наприклад, відстань між очима, форму носа і т.д. Цифрове розпізнавання осіб покладається в основному на 2D-зображення, бо їх легше порівняти з тими, що містяться в базі даних.

Етап 3. Цифровий відбиток особи. Система розпізнавання сприймає людину, як дані: результати аналізу особи перетворюються в цифровий код, що є унікальним «підписом лиця» кожної людини.

Етап 4. Пошук збігів. Відбувається порівняння системою відбитку особи з базою даних до тих пір, поки не буде знайдено збіг, і далі повертається результат із прикріпленою інформацією – ім'я та стать, наприклад (рис. 1.1.).

Все це можливо завдяки штучному інтелекту та нейронним мережам, які можуть класифікувати об'єкти.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		8

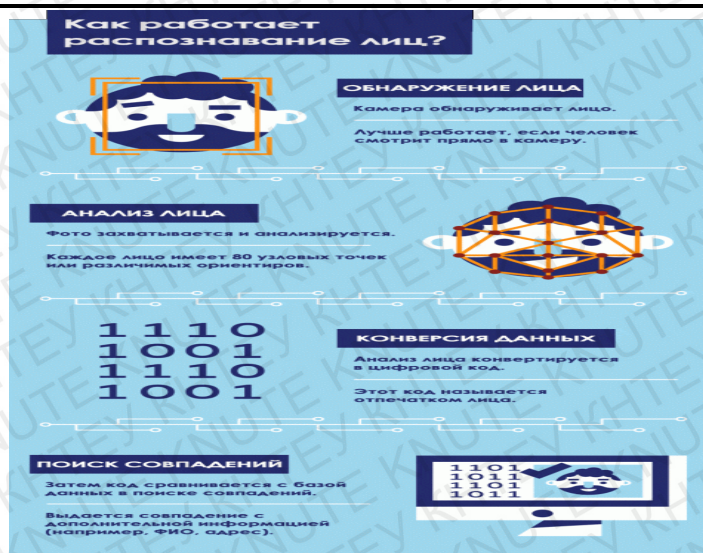


Рис. 1.1. Схема ідентифікації обличчя

За традицією об'єкт, що піддається класифікації, називається образом. Образом може бути як цифровий файл (розпізнавання зображень), буква або цифра (розпізнавання символів), запис мови.

Розпізнавання образів також включається в більш широку наукову дисципліну – теорію машинного навчання, в межах теорії штучного інтелекту, її метою є розробка методів побудови алгоритмів, що здатні навчатися.

Нейронні мережі – це клас моделей, заснованих на біологічній аналогії з мозком людини та призначених для рішення різноманітних задач аналізу даних після проходження етапу навчання на наявних даних.

При застосуванні даних методів дослідник може самостійно підбирати кількість шарів – глибину, та кількість «нейронів» у кожному з них.

Непростим є також вибір архітектури, бо на початковому етапі аналізу природа досліджуваного явища може бути відома погано, тому процес підбору пов'язаний із довготривалими «намаганнями та помилками». Тож саме для рішення подібних проблем останнім часом у комплексних задачах почали застосовувати методи штучного інтелекту.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

У процесі машинного навчання нейронні мережі ітеративно обробляють вхідні дані та корегують ваги всередині мереж аби та виконувала підгонку даних найефективнішим способом. Далі на тренувальній вибірці мережа є готовою до використання та для перевірки прогнозів.

Для тренування будь-якого алгоритму або нейронної мережі потрібна мати велику кількість наявних даних. Процес виявлення закономірностей у даних складається з декількох етапів:

- огляд та формалізація задачі аналізу;
- підбір та калібрування даних для авто-аналізу (препроцесинг);
- застосування методів Data Mining та побудова моделей;
- перевірка ефективності отриманих моделей;
- інтерпретація моделей людиною.

1.2. Змістовний опис та аналіз предметної області

Предметною областю цього дослідження є громадська сфера та її наступні характеристики: людина або групи людей, що зосереджені в одному місці з певною метою, потрібно розпізнавати та ідентифікувати їх у просторі та часі, аналізувати отримані дані для належної організації роботи та також стежити за безпекою. Адже, у компанії, це дозволить:

- контролювати виконання робочих завдань;
- аналізувати робочий час людей, чи працівник належно ставиться до своїх обов'язків (працює), чи займається іншими справами не у свою перерву;
- мати контроль доступу до спеціальних приміщень аби не було змоги потрапити, наприклад, до складу чи кабінету начальника непоміченим

					КНТЕУ 121 063-09.МР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

та можна було встановити особину застосування методів Data Mining та побудова моделей.

Це дозволить вчасно прийняти міри у випадку необхідності, навчить бути більш обізнаним у своїх працівниках та їх звичках, та краще контролювати робочий процес.

Нище наведена схема задач дослідження (рис. 1.2):

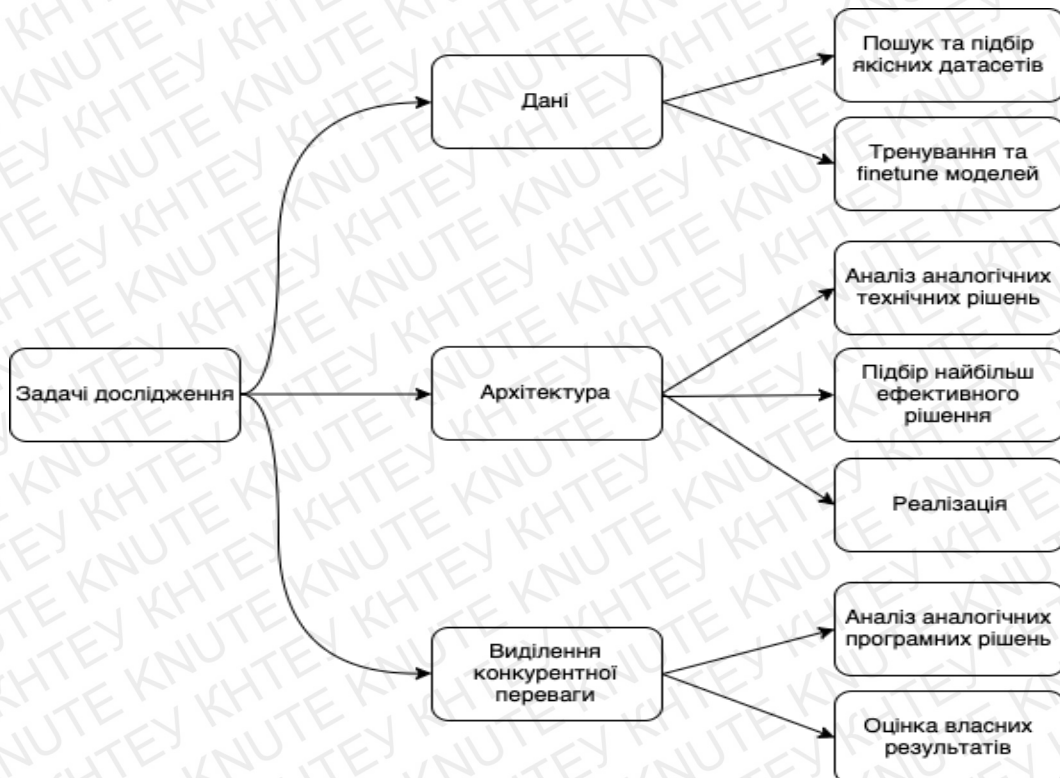


Рис. 1.2. Схема задач дослідження

Під час огляду предметної області, було визначено компоненти, що необхідні для реалізації:

- алгоритм для моделей розпізнання обличчя людини, рис обличчя та класифікації емоційного стану, віку, статі;
- навчальні вибірки (для розпізнавання віку, статі, емоцій);

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

- тестовий датасет;
- валідаційний датасет.

Модель для розпізнавання реалізує алгоритм комп'ютерного бачення та є універсальною. Тобто її можна перетренувати для розпізнавання образів інших предметних областей, завдяки коректно спроектованій архітектурі, навіть не дивлячись на те, що створена модель була для розпізнавання зовнішніх рис обличчя людини.

Зв'язки системи з користувачем та зв'язки між підсистемами наведені на діаграмі діяльності (рис. 1.3).

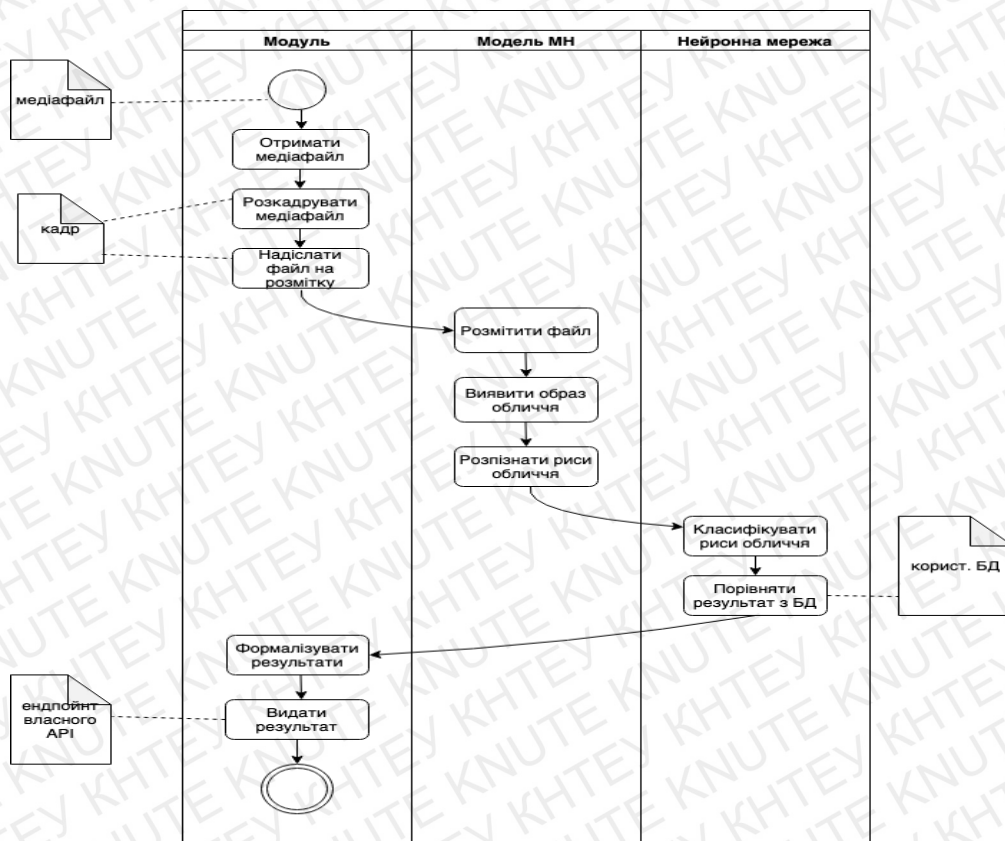


Рисунок 1.3. Діаграма діяльності

1.3. Аналіз успішних програмних рішень

Аби досконало підійти до розробки майбутнього проекту, потрібно провести комплексний огляд існуючих на ринку аналогів.

FindFace SDK

FindFace SDK – це бібліотека на мові С, яка надає доступ до передових технологій розпізнавання осіб на основі нейронних мереж. Швидко і точно вирішує 3 ключові завдання в сфері розпізнавання осіб: розпізнавання фрагментів з обличчями, розпізнавання характеристик зовнішності(стать, вік, настрій, наявність окулярів чи бороди), пошук по соціальній мережі ВКонтакте за фото. Користування платне, є демо-версія на 24 години зі скороченим функціоналом. Система розроблена стартапом NTechLab у 2015 році, приклад роботи наведений на рис. 1.4.

У 2018 році рішення FindFace увійшло до трійки лідерів в конкурсі з детектування пішоходів і велосипедистів на дорогах – Wider Pedestrian Challenge, організованого Amazon і SenseTime.

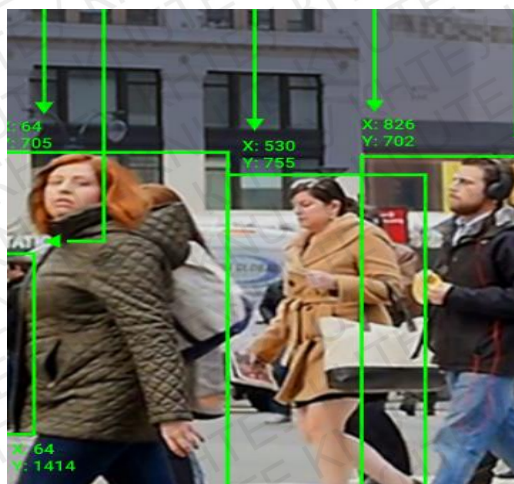


Рис. 1.4. Приклад роботи FindFace

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

SearchFace

Використовується для пошуку людей за фото у соціальній мережі ВК на базі алгоритму порівняння відстаней (пропорцій). Застосунок видає дуже швидкий результат. Розробка відбувалася російськими колегами, де отримання сім-карти дозволено лише по паспорту, тож власники порталу мають більш інформацію про те хто та кого шукає. Використання програми є платним та відбувається по номеру телефону. Рішення для комерційних клієнтів відсутнє. Приклад роботи наведений на рис. 1.5.

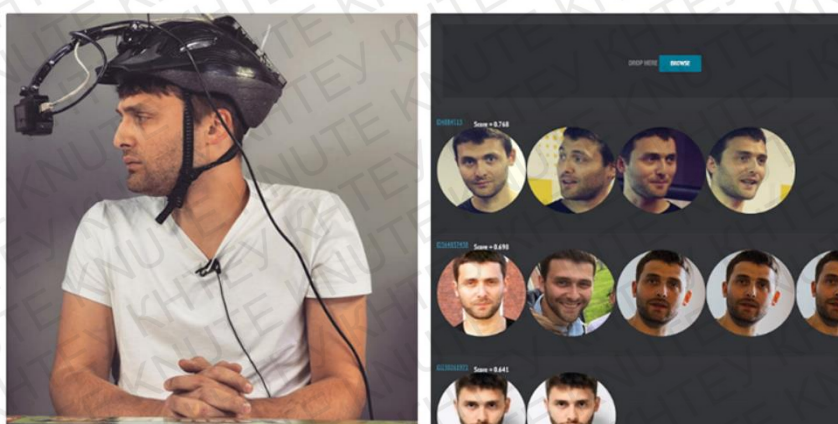


Рис. 1.5. Приклад роботи SearchFace

1.4. Формування технічного завдання

Завдяки проведеному аналізу характеристик аналогічних програмних рішень, було визначено основні варіанти використання програмного забезпечення згідно його функціоналу.

Головним діячем, актором є система; розробка не потребує адміністративних чи користувацьких функцій.

Визначимо основний функціонал модулю:

					<i>КНТЕУ 121 063-09.МР</i>	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

- отримання медіафайлу через API підключеного пристрою ;
- розпізнавання з медіафайлу образу людини;
- розпізнавання з образу характеристик зовнішності особи;
- класифікація характеристик зовнішності;
- перевірка на збіги із БД (за наявності БД користувача);
- формування результуючої видачі;
- передача результату через API модулю на пристрій.

Опираючись на розроблені сценарії користування, далі спроектовано структурну схему варіантів використання (рис. 1.6) та функціональних вимог (рис.1.7).

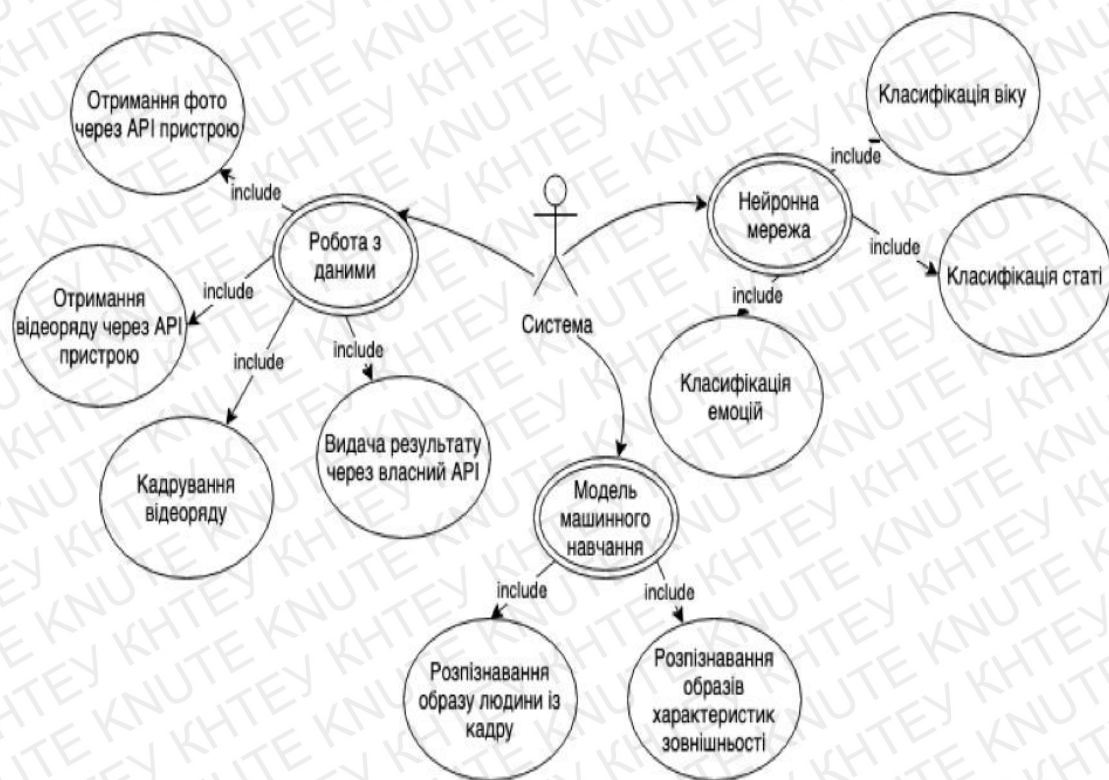


Рис. 1.6. Структурна схема варіантів використання

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

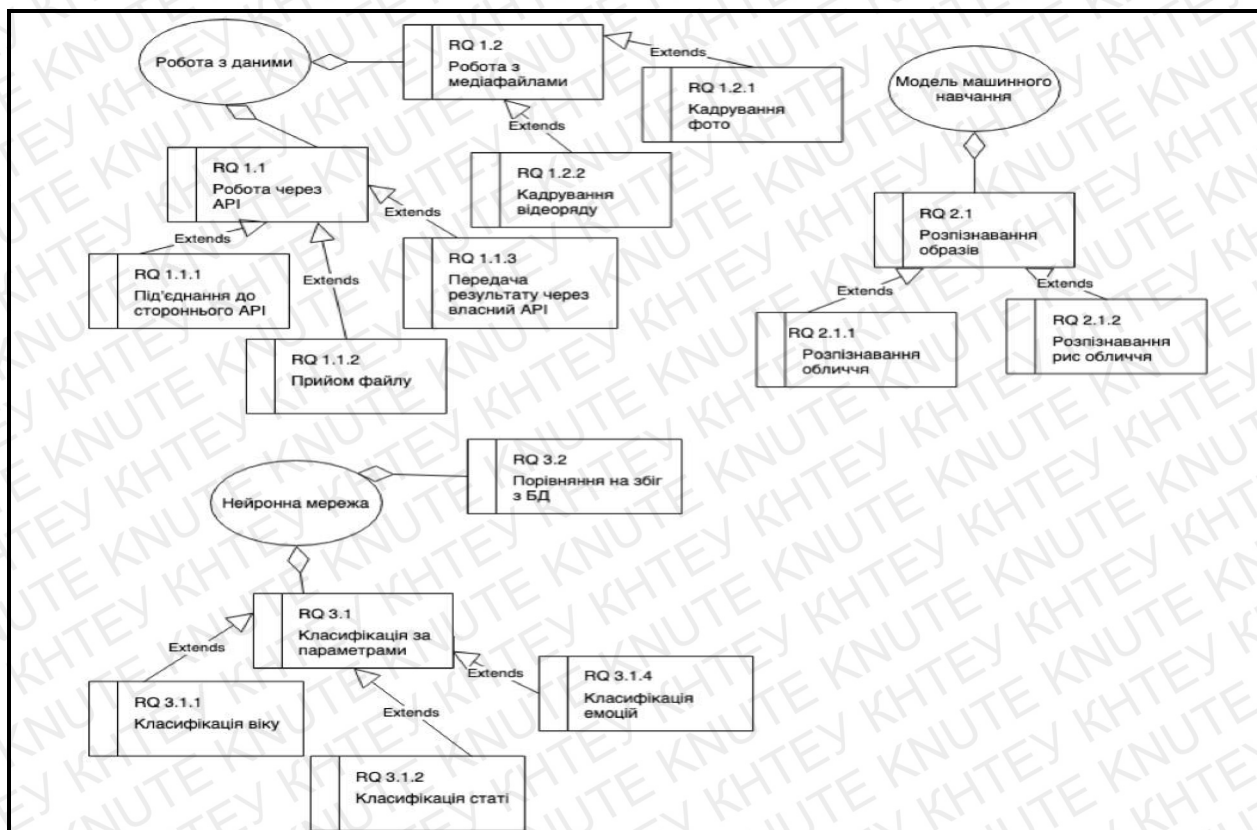


Рис. 1.7. Структурна схема моделі функціональних вимог

Постановка комплексу завдань модулю

Автоматизація розпізнавання об'єкту із наданого цифрового графічного файлу є призначенням даного модулю, та також його класифікація за характеристиками зовнішності людини. В даному випадку, це:

- стать;
- ім'я (порівняння з БД на збіг із даними про окрему особу);
- приблизний вік; (тестовий режим);
- емоційний стан (настрій); (тестовий режим)

Сам модуль може інтегруватися до операційного програмного забезпечення кампаній, барів, магазинів, тощо.

					КНТЕУ 121 063-09.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

Аби поставлена мета була досягнута, модулем повинні бути вирішені такі задачі:

- обробка цифрового файлу (фото або відео);
- кадрування;
- розмітка;
- детекція об'єкта;
- класифікація об'єкта за правилами навчального датасету;
- видача результату в уніфікованій формі.

Тож, розроблені моделі мають бути натреновані на мультикласових датасетах, для досягнення заданих умов.

Вихідні дані у кінцевому продукті мають бути однакового формату, незалежно від того, в яку систему його було інтегровано. Основною цільовою аудиторією є бізнес-структури, також, модуль повинен бути постійно доступним для всіх користувачів, хто має дозвіл на використання.

Розроблений функціонал у перспективі має бути інтуїтивно зрозумілим та простим, не переобтяженим. Через API програмного забезпечення пристрою, у яких буде інтегровано модуль, він має отримувати доступ до цифрових файлів. Ніякі особисті дані користувачів, окрім фото, не будуть збиратися.

У режимі локального використання система може функціонувати повністю автономно, тож можна опустити процес розробки серверної частини.

Тож виникає необхідність підбору найбільш ефективного алгоритму машинного навчання із розпізнавання та класифікації на даний час.

У разі ж необхідності серверного використання рекомендовано реалізувати модуль, з огляду на можливе розширення функціоналу або

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

перехіж до стану самостійної роботи, із можливістю масштабування та також слід урахувати зростаюче навантаження від запитів користувачів.

1.5. Висновки до розділу 1

Отже, у першому розділі було розглянуто принципи функціонування мережі та основна інформація щодо її створення. Було сформоване технічне завдання та вимоги до технічного устаткування та середовищ розробки, з якими буде проводитись взаємодія. Також було проведено огляд існуючих аналогів та аналіз перспектив розвитку на прикладі FindFace SDK та SearchFace.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

РОЗДІЛ 2.

МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Розробка структури та загальна концепція побудови

На основі проаналізованих функціональних фімог та також списку варіантів використання, було сформовано також структурну схему бізнес процесів, яка вказує на розшированість компонентів створюваного програмного забезпечення.

Нище наведено структурну схему бізнес процесів (рис.2.1). Тут можна чітко побачити 3 основні частини архітектури системи:

- інтерфейс для роботи з даними;
- модель для розпізнавання образів, яка інкапсулює в собі нейронну мережу для класифікації результату;
- бекенд з інтегрованою документо-орієнтованою базою даних.

Потрібно не просто вирішити задачу розпізнавання об'єкта, що є лише частиною цієї роботи, адже ціль дослідження – класифікація людини за її зовнішніми характеристиками. Однією зі складових розробки є детекція об'єкта, тобто його визначення. Аби мати змогу класифікувати людину, програмне забезпечення має визначити місцезнаходження її образу на зображенні та виокремити його в “Bounding box” (рамку) – знайти та віділити координати.

Тож, у даному випадку, системою отримано цифровий файл із графічною інформацією, у результаті його обробки має бути отримано унікальний ідентифікатор обличчя, клас та його координати на зображенні.

					<i>КНТЕУ 121 06з-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	Стадія	Аркуш	Аркушів
Зав. каф.		Криворучко О.В.		07.09.20		P2	19	49
Керівник		Криворучко О.В.		07.09.20		Факультет інформаційних технологій 2м курс, 6з група		
Гарант		Криворучко О.В.		07.09.20				
Розробив		Опенько Д.П.		07.09.20				
					<i>Моделювання та конструювання програмного забезпечення</i>			



Для цього потрібно зробити розмітку даних на фото, в датасеті знайти координати образу кожної особини, як показано нище на рис. 2.2.



Рис. 2.2. Приклад детекції об'єкта

Робити розмітку вручну – займає багато часів та уваги, тож було вирішено використати модель, натреновану на позиціонування деяких класів об'єктів на зображенні, аби скоротити цей час та автоматизувати даний процес. Далі на рис. 2.3. наведено приклад роботи цієї моделі.



Рис. 2.3. Приклад ідентифікації об'єкта

Було виконано розмітку початкової вибірки аби вона задовільняла вирішення поставлених завдань.

Вимоги до формату даних є наступними – object-class; x, y; width, height; де:

- object-class – ідентифікатор обличчя із бази класів інтерфейс для роботи з даними;
- x та y – координати центру рамки модель для розпізнавання образів, яка інкапсулює в собі нейронну мережу для класифікації результату;
- width та height – виміри ширини та висоти рамки.

Для покращення точності результатів роботи моделі та також аби зробити саму модель більш універсальною, потрібно виконати “train test split” вибірки. Тобто декомпозицію оригінальної вибірки на окремі датасети.

Тренувальна вибірка для навчання моделі має займати близько 60-80% від загальної вибірки.

Також для поліпшення характеристик узагальнення буде використана валідаційна вибірка. На ній буде калібруватись модель після завершення навчання. Дана вибірка складає близько 10-20% від загальної вибірки.

Сама ж тестова вибірка буде використана лише для перевірки моделі. Адже її точність дасть уявлення про те, наскільки коректно алгоритм навчання моделі працював та чи може модель робити передбачення на нових даних, тобто наскільки вона є узагальнена. Ця вибірка також складає приблизно 10-20% від загальної вибірки.

Можливо оптимізувати процес навчання моделі, якщо використати методи препроцесингу даних. Це може вирішити задачі комп'ютерного бачення, проте так як для задач важливі лише координати рис обличчя, колір кадру не має такого значення. Також можливо у рази зменшити кількість обчислень, якщо кадрувати кадр у чорно-білий.

					КНТЕУ 121 063-09.МР	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

Це прискорить навчання моделі, адже полегшить сходимість градієнтного спуску.

Далі, інтеграція бази даних MongoDB, - буде використано дистрибутив драйвера pymongo, для організації доступу до неї, адже основною мовою програмування серверної частини є Python. Менеджер Python – PIP, використається аби встановити та налаштувати pymongo.

У подальшій роботі з власною БД буде створено об'єкт MongoClient, який містить посилання на вірну ір-адресу, усі подальші дії будуть також вионані через цей клієнт.

Подальша робота буде виконана за допомогою стандартних команд, які використовуються у документо-орієнтованих базах даних.

Стандартне виведення відеофайлу або кадру на екран буде використано у якості графічного інтерфейсу, адже системою не надано зовнішній функціонал для взаємодії з людиною.

Сам же термін “інтерфейс” тут виступає у значенні кінцевої точки роботи з даними, що має на увазі їх приведення до зрозумілого людині виду та організація для передачі на інше ПЗ.

2.2. Конструювання програмного забезпечення

2.2.1. Методологія та підходи до розробки програмного забезпечення

Дане програмне забезпечення було розроблено відповідно до плану, що було наведено у листі завдання разом із перерахуванням сформованих вимог та термінами виконання робіт.

Правила методології ґрункої роботи над проектами, згідно з якими йшов процес роботи, забезпечували ефективну роботу над проектом та зручне

					<i>КНТЕУ 121 063-09.МР</i>	Аркуш
						23
Зм.	Аркуш	№ докум	Підпис	Дата		

планування, а також дали можливість отримувати готову для тестування частину роботи кожне декілька тижнів праці.

Аби мінімізувати код, було вирішено використовувати готові модулі повторно та здійснювати аналіз коду готових модулів.

2.2.2. Використані технології

У процесі підготовки до розробки було обрано використовані технології:

Jupyter Notebook – веб-додаток, який дозволяє опрацьовувати документи із живим кодом, рівняннями, візуалізацією та текстом розповіді.

Python – об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією.

Sublime Text – редактор файлів коду, розроблений Microsoft для Windows, Linux та macOS. Включає у себе підтримку налагодження, вбудований контроль Git та GitHub та підсвічування синтаксису.

MongoDB – це міжплатформна програмна база даних, орієнтована на документи. MongoDB, класифікований як програма баз даних NoSQL, використовує JSON-подібні документи зі схемою.

PyTorch – широкодоступна програмна бібліотека для вирішення задач машинного навчання.

OpenCV (комп'ютерне бачення з відкритим кодом) – це бібліотека функцій програмування, що головним чином спрямовані на комп'ютерний зір у режимі реального часу.

Також, для системи контролю версій та організації коду, було вирішено використати систему git, при плануванні середовища, як центральне сховище якої були використані репозиторії, безкоштовно надані сарвісом розробки ПЗ GitHub.

					КНТЕУ 121 063-09.MP	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

2.3. Висновки до розділу 2

У другому розділі була розроблена структура та загальна концепція побудови системи, були ретельно проаналізовані функціональні вимоги та перелік варіантів використання.

Було обрано середовище розробки та мову програмування, проведена розробка проекту. Роботу було поділено на три частини: серверна частина, модуль розпізнавання та інтерфейс для роботи з даними. Було інтегровано документно-орієнтовану базу даних до серверної частини, а також інкапсульовано нейронну мережу в модулі розпізнавання. Функціонал інтерфейсу буде мінімальним.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

РОЗДІЛ 3. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

3.1. Узагальнена архітектура мережі

Як вже зазначалося у попередніх розділах, робота мобільного додатка для імітації тачпада ПК побудована на взаємодії двох пристроїв, що виступають один для одного клієнтом (смартфон) та сервером (комп'ютер). Рисунок 3.1 та 3.2 схематично описують механізм роботи додатка та обґрунтовують розробку окремих модулів, що представляють собою серверну частину та клієнтську частину програмного забезпечення.



Рис.3.1. Структурна схема задач

Три підсистеми вирішують кожна своє завдання.

Головна частина, якою є модуль розпізнавання, для детекції образів облич та їх характеристик, робить у собі інкапсуляцію моделей з нейронними мережами, а також модуль підготовки датасету.

					<i>КНТЕУ 121 06з-09.МР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.	Криворучко О.В.		19.10.20	<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
Керівник	Криворучко О.В.		19.10.20		<i>РЗ</i>	<i>26</i>	<i>49</i>	
Гарант	Криворучко О.В.		19.10.20		<i>Факультет інформаційних технологій 2м курс, 6з група</i>			
Розробив	Опенько Д.П.		19.10.20	<i>Архітектура нейронних мереж</i>				

За інформаційну архітектуру, котрою оперую система, класифікацію та розмітку отриманих даних, відповідає інтерфейс для роботи з даними.

За зв'язок із іншими підсистемами у цьому модулі відповідають строго класифіковані дані, направлені до моделі для розпізнавання. Це дозволяє «навчати» модель розпізнаванню людини на фото як об'єкта, раз за разом оброблюючи цю інформацію, аби отримати можливість класифікувати обрані унікальні ознаки кожної людини для подальшої зв'язки між собою серед уже відомих класами розмічених характеристик, згадуваних раніше.

Інтерфейс для роботи з даними опрацьовує велику кількість зображень. Сама ж модель для розпізнавання працює з єдиним кадром – яким є вхідні дані, із невідомими характеристиками зовнішності, – тож метою підсистеми, і всього продукту, є класифікація ознак поданої особи.

Тож, обидві згадані вище підсистеми приховані від користувачів та несуть у собі лише обчислювальний сенс для продукту. Звичайно, прихованою є і серверна частина, яка організовує доступ до власної чи користувацької (якщо така є) бази даних.

					КНТЕУ 121 063-09.МР	Аркуш
						27
Зм.	Аркуш	№ докум	Підпис	Дата		

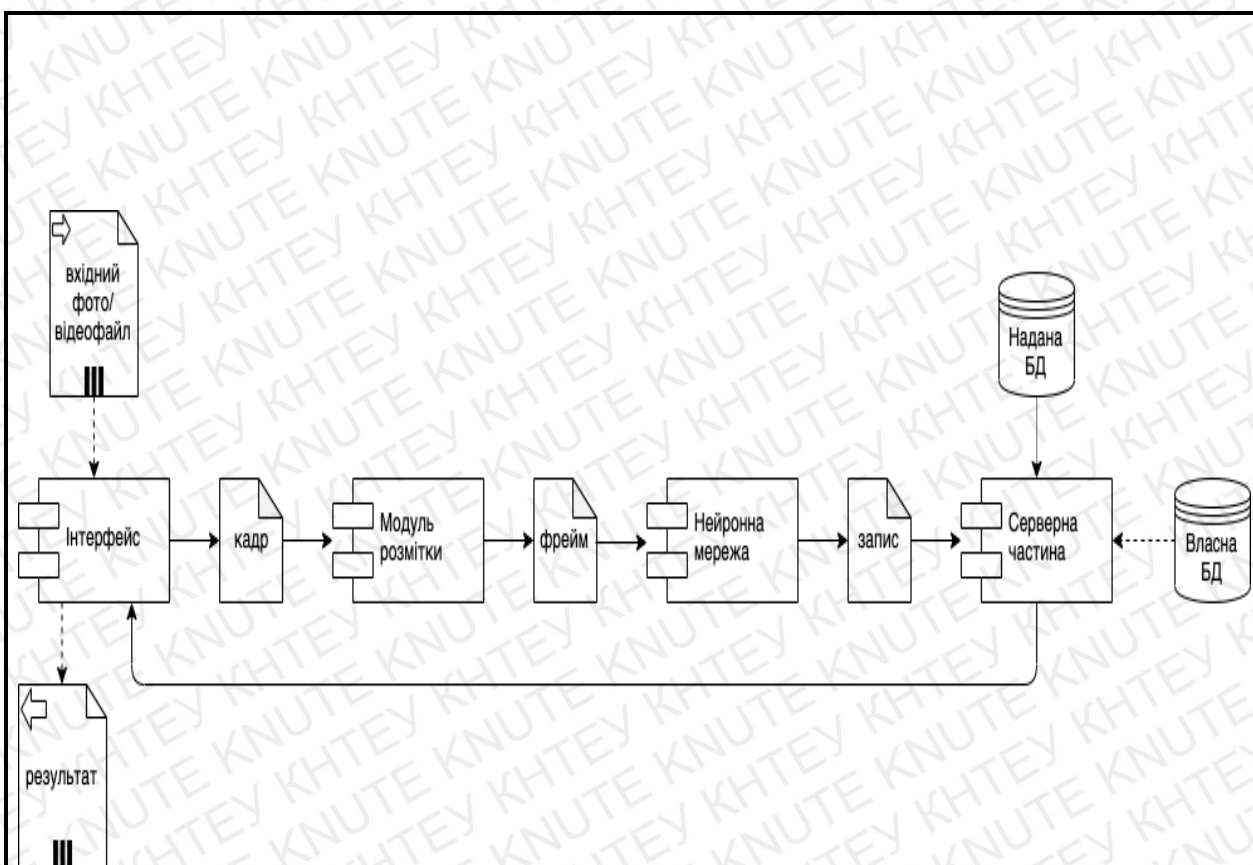


Рис. 3.2. Структурна схема компонентів системи

3.2. Модель розпізнавання та класифікації

У задачах класифікації та регресії необхідно визначити значення залежної змінної об'єкту, базуючись при цьому на значеннях інших змінних, які характеризують даний об'єкт. Формально ми можемо описати задачу класифікації та регресії таким чином. Дана множина об'єктів:

$$I = \{i_0, i_1, \dots, i_n, \dots, i_m\}, \quad (3.1)$$

де i_0 – досліджуваний об'єкт.

Для приклада таких об'єктів можемо використати таблицю залежності окремих характеристик рис обличчя людей та їх походження (таблиця 3.1).

Таблиця 3.1.

Залежність відмінних характеристик обличчя

Форма	Пропорції	Тон шкіри	Колір очей	Волосся	Походження
Округла	Мініатюрні риси, вузькі очі	Світлий	Від карих до чорних	Чорне, пряме	Азія(Японія)
Прямокутна	Гострі, важкі скули	Засмаглий	Від карих до чорних	Темне, пряме	Азія(монголи)
Округла	Пласке обличчя, вузькі очі	Світлий, жовтуватий	Від карих до чорних	Чорне, пряме	Азія(Китай, Корея)
Видовжена	Тонкі, видовжені	Блідий	Світлі – сірі та блакитні	Світле, пряме	Скандинавія
Видовжена	Виражені риси	Засмаглий	Зелені, карі, чорні	Темне, курчаве	Середземномор'я

Де кожен об'єкт охарактеризовано набором змінних:

$$I_i = \{x_s, x_+, \dots, x_-, \dots, x_-, y\}, \quad (3.2)$$

де x_+ – це незалежна змінна, значення якої відоме та на основі якої визначається значення залежної змінної y .

У наведеному варіанті форма обличчя, колір очей та тип волосся є незалежними змінними. Походження же – є залежним.

Часто набір незалежних змінних у Data Mining позначають як вектор.

$$X = \{x_s, x_+, \dots, x_-, \dots, x_-\}. \quad (3.3)$$

У результаті, кожна змінна x_i може приймати значення із деякої множини:

$$C_+ = \{c+\$, c+\&, \dots\} \quad (3.4)$$

Скінченна множина має категоріальний тип у разі, коли її елементи є значеннями змінної. Це означає, що обрана змінна “форма обличчя” буде приймати значення із множини {видовжена, округла, прямокутна, ромбовидна}.

У випадку, коли множина значень $C = \{c_\$, c_\&, \dots, c_l, \dots, c_4\}$ змінної y – скінченна, задача буде називатися задачею класифікації. Якщо ж змінна y буде приймати значення на множині дійсних чисел R , задача стане задачею регресії.

Незважаючи на описаний спосіб визначення значень залежної змінної через функцію класифікації чи регресії, зовсім не обов'язково він повинен виражатися математичною функцією.

Існує декілька основних видів подання таких способів: математичні функції, класифікаційні правила та дерева рішень.

3.2.1. Ідентифікація обличчя

Архітектура ArcFace разом із тренувальним датасетом MS1M була застосована для ідентифікації облич. ArcFace безпосередньо оптимізує запас геодезичних відстаней за рахунок точної відповідності між кутом і дугою в нормалізованій гіперсфері [де розташовані риси обличчя]. Після проведеного аналізу різних публікацій та порівняльних тестів було визначено, що ArcFace значно перевищує сучасні технології та він може бути реалізованим без значних обчислювальних витрат.

Наразі при дослідженнях є тенденція включати запас (маржу) у встановлені функції витрат задля максимально забезпеченої відокремленості облич.

					КНТЕУ 121 063-09.МР	Аркуш
						30
Зм.	Аркуш	№ докум	Підпис	Дата		

ArcFace застосовує адаптивну кутову втрату маржі аби отримувати варіативні риси розпізнавання обличчя. Ця архітектура уособлює у собі найбільш масштабну експериментальну оцінку серед сучасних методів розпізнавання обличчя, маючи більше 10 орієнтирів для розпізнавання облич, разом із новою великою базою даних зображень та відео.

Першочергово ArcFace було реалізовано через фреймворк Apache MXNet, та для реалізації цієї роботи використано бібліотеку PyTorch.

3.2.2. Розпізнавання статі та віку

Аби мати змогу визначати стать та вік обраної особи, було прийняте рішення про використання моделі архітектури MobileNet. Це дає можливість запуску мережі на найслабкіших пристроях, тому отримані результати є кращими серед невеликої кількості класів.

У нашому випадку мережа для визначення статі буде мати лише 2 класи – чоловічий та жіночий. Класифікування за віком бідбуватиметься по бінам – тобто класам, які містять 10-річні проміжки.

Система на обробку буде отримувати загалом розподілені на приблизно 10 класів файли, що дає невелике навантаження та добре підходить для роботи з MobileNet.

Далі наведена спрощена схема побудови даної мережі (рис. 3.3).

					КНТЕУ 121 063-09.МР	Аркуш
						31
Зм.	Аркуш	№ докум	Підпис	Дата		



Рис. 3.3. Спрощена схема побудови MobileNet

Згорткова частина мережі складена з одного звичайного згорткового шару 3x3 на початку та тринадцяти блоків, що зображені на рисунку, з поступово зростаючою кількістю фільтрів та просторовою розмірністю тензора, що знижується.

Два гіперпараметри архітектури MobileNet складають множник ширини та множник глибини (множник дозволу). Це дозволяють варіювати розміри самої мережі: через зменшення цих множників відбувається зниження точності розпізнавання, проте збільшення швидкості роботи і зменшення споживаної пам'яті.

3.2.3. Розпізнавання емоційного стану

Розпізнавання емоцій є одним із найбільш тяжких завдань. Аби система могла коректно проводити його, буде використана модель архітектури ResNet на 18 шарів. Враховуючи глибину цієї мережі, це дозволить проводити

класифікаю об'єктів по 1000 класам та приймати зображення для обробки починаючи від формату 224x224 на 3 (де 3 – число каналів кольору).

У результаті, така кількість шарів дозволяє отримувати більше можливостей та точних результатів, адже вона відрізняється від звичайної архітектури нейронних мереж. Нейронні мережі такого типу зуться глибокими, а область знань, що з ними пов'язана – глибоким навчанням.

Якщо узагальнити, глибока нейронна мережа є звичайною нейронною мережею, просто має додаткові шари, між якими як функції активації виступають функції ідентичності. Таких додаткових шарів може бути безліч, як і функцій, проте головне аби був збережений баланс між ресурсами, які потребує система для обчислень та точності результатів. У глибокій мережі також менше дисперсійність помилки.

Далі проілюстровано застосування функцій ідентичності (рис. 3.4.).

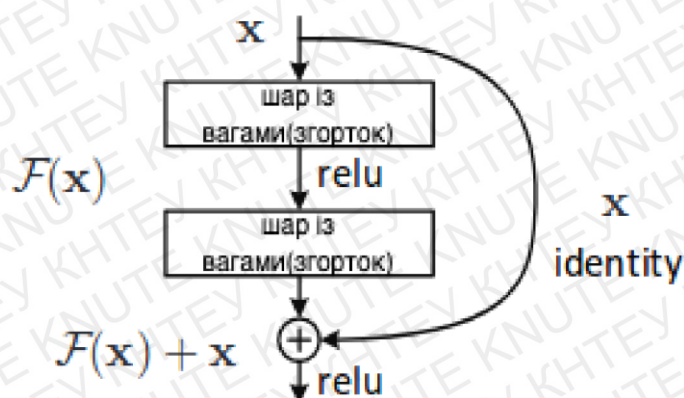


Рис. 3.4. Схема роботи глибокої залишкової нейронної мережі

Аби напряду не навчати модель використанню функції $F(x)$ для мапінгу x на y (декілька послідовних нелінійних шарів), потрібно буде визначити залишкову функцію за допомогою $F(x) = H(x) - x$, яку можна перетворити в $H(x) = F(x) + x$, де $F(x)$ та x являє собою складені нелінійні шари і функцію ідентичності (вхід = вихід).

При оптимальному маппінгу ідентичності можна легко звести залишки до нуля ($F(x) = 0$), а не підводити до $F(x) = x$.

Простіше вирішити рішення типу $F(x) = 0$, а не $F(x) = x$, через використання стеку нелінійних шарів згорткової нейронної мережі в якості функції. Тому функція $F(x)$ і є залишковою функцією.

3.3. Реалізація зберігання даних

На етапі проектування даного проекту було вирішено використати базу MongoDB для зберігання даних. Вона є документо-орієнтованою, дані зберігаються в JSON-подібних структурах. Даний спосіб прийняття та зберігання даних є більш потужним, ніж традиційна реляційна модель.

При використанні даного рішення немає потреби в описі схем таблиць та нормалізації даних, що дає можливість пришвидшити розробку та спростити зручність роботи з даними в області знань машинного навчання.

Далі наведено схему моделі бази даних (рис. 3.5.).

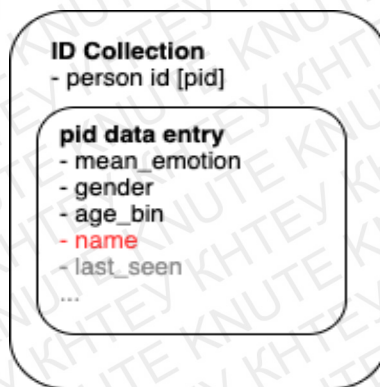


Рис. 3.5. Схема моделі бази даних

Дивлячись на схему можна переконатись, що реалізована організація зберігання інформації про розпізнаних осіб є дуже простою, адже є лише один

рівень вкладеності, що дуже спрощує та пришвидшує пошук при порівнянні з показниками решти БД, які модуть підключати зі сторони замовника.

Така структура дозволяє також горизонтальне масштабування, тобто опрацювання відеофайлів із присутністю тисяч людей.

Більш детально про поля організації даних йдеться у таблиці 3.2.

Таблиця 3.2.

Опис структури збереження даних

Ідентифікатор	Опис	Коментар
pid(person id)	Унікальний ідентифікатор одного розпізнаного за характеристиками обличчя	Обов'язковий параметр
mean_emotion	Показник усередненої емоції(від 0 до 4)	Обов'язковий параметр
gender	Бінарний показник статі, де 0 – чоловіча, 1 – жіноча.	Обов'язковий параметр
age_bin	Бін віку, тобто система розпізнає вікову категорію у діапазоні 10 років, тож 0 – 0р до 10р, 1 – 10р до 20р, і т.д	Обов'язковий параметр

Name	Запис імені особи, що була розпізнана із підключеної бази із даними про злочинців/небажаних осіб/зниклих і т.п	Обов'язковий параметр за умови, що підключена база для порівняння характеристик та розпізнана особа наявна в цій базі
other...	Перелік інших параметрів, що можуть бути підлаштовані під ту чи іншу предметну область	Не обов'язковий параметр

3.4. Висновок до розділу 3

У третьому розділі був розглянутий та проілюстрований процес проектування архітектури нейронних мереж, реалізованих у моделях розпізнавання. Для вирішення усіх питань, розроблено три моделі – кожна для своєї підзадачі. Описані особливості проектування моделі баз даних та специфікацій. Здійснена координація роботи над усіма частинами проекту.

РОЗДІЛ 4.

АНАЛІЗ ТА РОЗРОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Перевірка ефективності розробки

Показник точності розпізнавання є кращою характеристикою комп'ютерного бачення, розділу теорії машинного навчання. Задля виміру точності класифікації та розпізнавання, які були реалізовані у проекті, використовувалася метрика «*Recall*». Краще ілюструє її рисунок 4.1.

		Reference variant set	
		Positive	Negative
Variants Called by the Algorithm	Positive	True Positive (TP) Correct variant allele or position call	False Positive (FP) Incorrect variant allele or position call.
	Negative	False Negative (FN) Incorrect reference genotype or no call.	True Negative (TN) Correct reference genotype or no call.

Рис. 4.1. Графічне пояснення метрики Recall

Матриця варіантів (рис.4.1) відображає усі результати роботи моделі – [TP] правильне передбачення і правильний результат, [FP] неправильне передбачення і правильний результат, [FN] правильне передбачення і неправильний результат, [TN] неправильне передбачення і неправильний результат.

					<i>КНТЕУ 121 06з-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.	Криворучко О.В.			19.10.20	<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	Стадія	Аркуш	Аркушів
Керівник	Криворучко О.В.			19.10.20		P4	37	49
Гарант	Криворучко О.В.			19.10.20		Факультет інформаційних технологій 2м курс, бз група		
Розробив	Опенько Д.П.			19.10.20				
					<i>Аналіз та розгортання програмного забезпечення</i>			

Звідси можна легко порахувати метрику, що нас цікавить.

(4.1)

$$Precision = \frac{TP}{TP + FP}$$

Дана формула добре ілюструє значення метрики recall. Рахується так: відношення кількості вірних передбачень до суми правильних і неправильних (при вірних результатах). Чим більший результат, тим краще.

Також свою роль відіграє коефіцієнт схожості – безрозмірний показник схожості порівняних об'єктів. Його використовують аби визначити ступінь схожості серед будь-яких об'єктів, при умові, що набір характеристик дозволяє їх порівняти. Більшість коефіцієнтів є нормованими

та вони знаходяться в межах від 0 (це коли схожості немає) до 1 (повна схожість). Схожість та різність є взаємодоповненнями одне для одного.
 $sхожість = 1 - різність$

Можна взяти коефіцієнт Жаккара для порівняння:

(4.2)

$$KJ = \frac{c}{a + b - c},$$

де a – кількість характеристик одного походження, b – кількість характеристик іншого походження, c – кількість спільних характеристик.

Також, можливо відображення коефіцієнту через таблицю спряженості Сокала та Сніта.

					КНТЕУ 121 063-09.МР	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

Також, можливо відображення коефіцієнту через таблицю спряженості Сокала та Сніта. Наприклад, системою було визначено колір очей людини як коричневий, проте такому опису відповідає декілька класифікованих походжень: скандинави, слов'яни, азіати, африканці. Дані відображені у таблиці 4.1.

Таблиця 4.1.

Таблиця спряженості Сокала та Сніта

	Карі очі: шатенки	Карі очі: інші
Основний колір: коричневий	a	b
Основний колір: інший	c	d

де a , b , c , d – вже розпізнані імовірності наявності характеристики походження, яке розпізнавалося. У такому випадку, якщо параметр a матиме найбільшу імовірність, буде виданий результат, що людина на фото – азіат/азіатка.

Тож аби порівняти використані при моделюванні та конструюванні ПЗ методи та їх ефективність, були змодельовані умови аналогічно подібних архітектур для систем комп'ютерного бачення.

Порівняльне тестування відбувалось за таких умов:

- процесор Intel Core i7;
- відеокарта Nvidia GeForce 1060 6 Gb;
- RAM 16 Gb;

- SSD 512 Gb.

Було також враховано особливості обраного рішення при проведенні порівняльного тестування ідентифікації по обличчю:

- датасет MS1M;
- функція активації ReLU6;
- функція втрат Additive Angular Margin Loss;

У таблиці 4.2 відображені результати тестування рішення для ідентифікації осіб.

Таблиця 4.2.

Тестування ефективності ідентифікації

Метод	ArcFace	CosFace	SphereFace (3)	Light CNN- 29	SphereFace (1)	FaceNet
Рік	2018	2018	2017	2015	2017	2015
Точність	98.35%	82.72%	75.766%	73.75%	72.279%	70.49%

Наступні особливості були включені при тестуванні функціоналу розпізнавання віку та статі:

- finetune модель на датасеті IMDB Faces;
- функція активації ReLU6.

У таблиці 4.3 відображені результати тестування рішення для розпізнавання віку та статі.

Таблиця 4.3.

Тестування розпізнавання віку та статі

					КНТЕУ 121 063-09.МР	Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата		

Версія	Рік	Топ 1 точність	Топ 5 точність
MobileNet V1	2017	70.9%	89.9%
MobileNet V2	2018	71.8%	91.0%

При тестуванні розпізнавання емоційного стану, було враховано:

- відкритий датасет із Kaggle.com;
- функція активації ReLU.

У таблиці 4.4 відображені результати тестування рішення для розпізнавання емоційного стану.

Таблиця 4.4.

Результати тестування розпізнавання емоційного стану

Версія	Рік	Топ 1 точність	Топ 5 точність
VGG16	2014	71.3%	90.1%
InceptionV3	2015	77.9%	93.7%
Xception	2016	79.0%	94.5%
ResNets	2015	80.3%	95.3%

Звичайно, можна було використати інші версії ResNet для розпізнавання емоцій, наприклад 50, 101 шари – тоді показники точності був вищий, проте через апаратні можливості, для коректної роботи системи, потрібно було би значно більше часу на обчислення. Це робило програму не такою ефективною та не було раціональним. Тож було вирішено використовувати легшу з версій для реалізації.

4.2. РОЗГОРТАННЯ ПРОГРАМНОГО ПРОДУКТУ

При розгортанні програмного продукту потрібні такі мінімальні апаратні вимоги для коректної роботи:

- CPU/GPU: Процесор з підтримкою AVX / NVidia GPU з 6 Gb пам'яті для більш точних обчислень, 3 Gb пам'яті для більш точних обчислень.
- RAM: ~3.5Gb RAM.
- OS: ОС Windows 7, 8, 10 (64 бит), Linux Ubuntu (64 біт).

Запуск програми відбувається через командну строку із вірної директорії (рис.4.2).

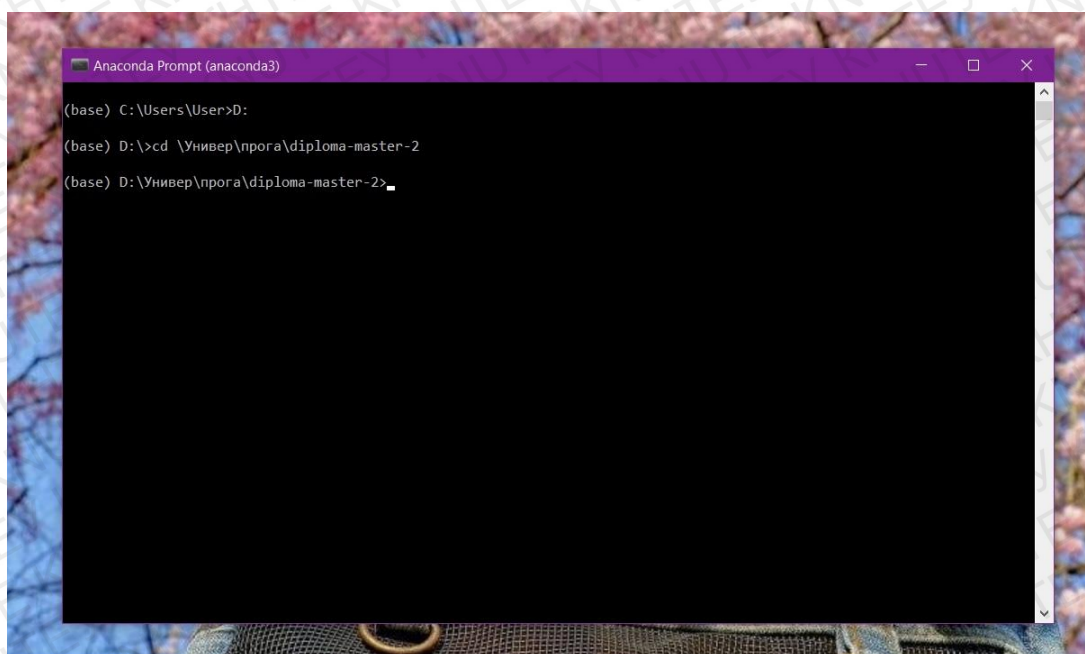


Рис. 4.2. Запуск програми

Програма буде продемонстрована на ідентифікації особи по відео, порівняння буде іти із фото особи, яке є у базі даних. Для розпізнання було обрано Джимі Фелона. Програма отримала таке відео:

					КНТЕУ 121 063-09.MP	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		



Рис. 4.3. Фрагмент відео

Це фрагмент програми, де двоє людей ведуть бесіду між собою. Вони також знаходяться боком до камери, що робить процес розпізнавання складнішим.

Для прикладу, наведено одне з фото із бази для порівняння (рис.4.4).



Рис. 4.4. Приклад фото особи

Тож, відбувається запуск програми, командою для аналізу відео:
python from_video.py.

					КНТЕУ 121 063-09.МР		Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			43

Далі можливі 2 сценарії:

1) У випадку, якщо на комп'ютері не вистачає пам'яті іде збій аналізу (рис. 4.5).

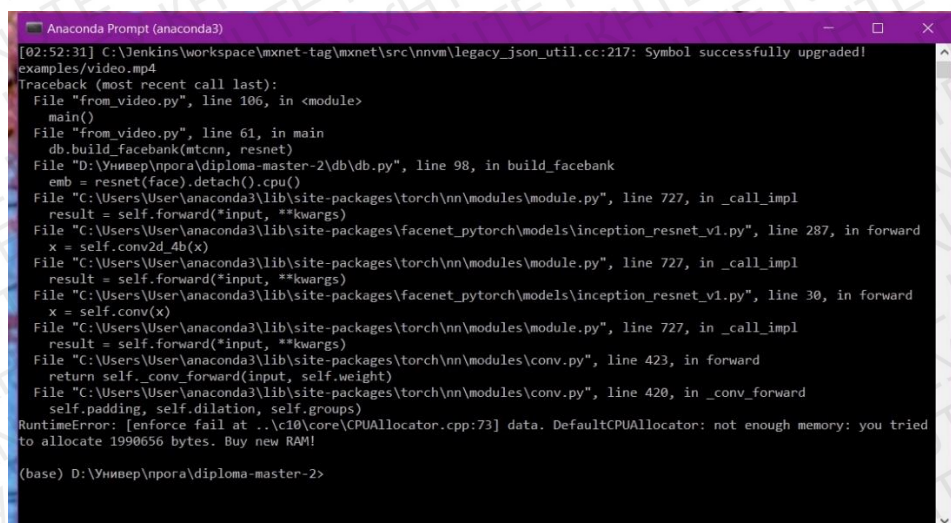


Рис. 4.5. Збій у аналізі

2) При успішному аналі відео, програма рендерить у обрану папку дані з результатами і користувач може відкрити та подивитися їх (рис. 4.6).

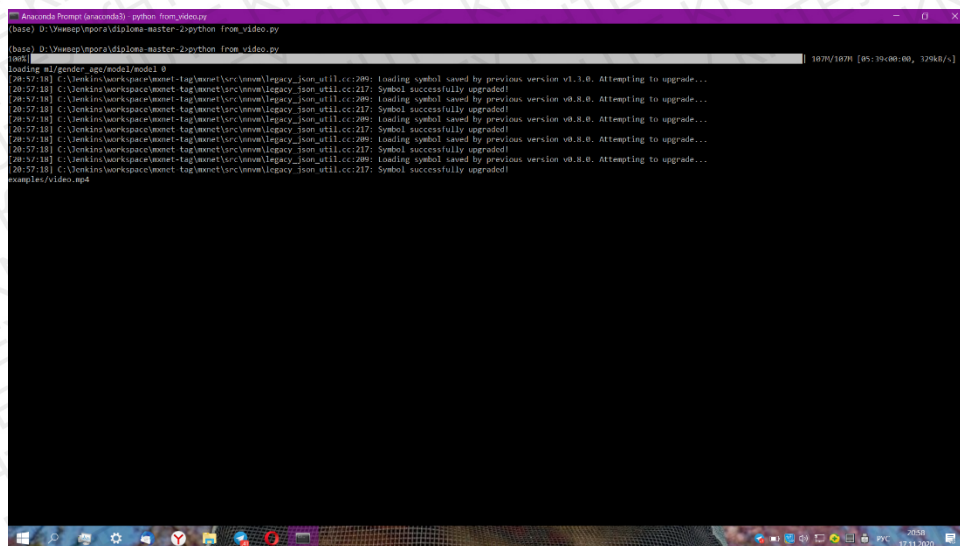


Рис. 4.6. Аналіз проведено успішно

Тож коли система сповіщає про успішну обробку поданої інформації, вона виводить відеофайл або кадр на екран. Це використовується у якості графічного інтерфейсу, адже системою не надано зовнішній функціонал для взаємодії з людиною. Сам же термін “інтерфейс” тут виступає у значенні кінцевої точки роботи з даними, що має на увазі їх приведення до зрозумілого людині виду та організація для передачі на інше ПЗ (рис.4.7).

Перевагами даної розробки є те, що більшість подібних програмних продуктів у світі створено для виключно комерційних цілей, ціни на них є дуже високими а також вони потребують постійної підтримки з боку сертифікованих спеціалістів. Також вони визначені для конкретних задач, тоді як дану розробку можна перепрофілювати на іншу – просто перевчити на тренуваннях.

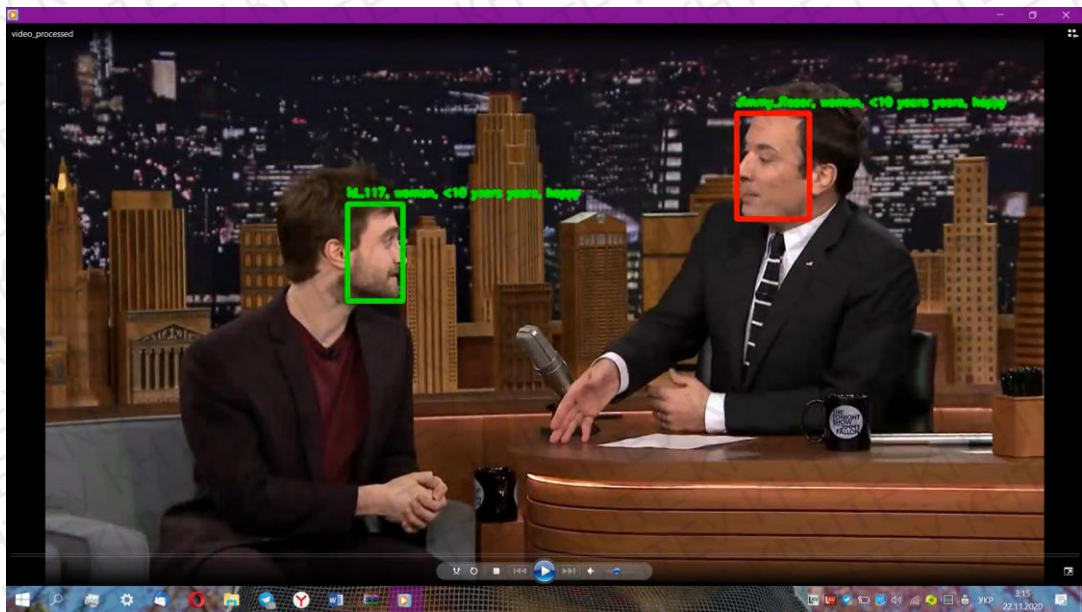


Рис. 4.7. Файл із результатом

					КНТЕУ 121 063-09.MP	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		45

4.3. Висновки до розділу 4

У четвертому розділі був виконаний аналіз метрик оцінки ефективності системи. Було порівняно ефективність роботи реалізованих моделей за однакових умов та при одному і тому ж апаратному забезпеченні. Було проілюстровано порядок розгортання програми та можливі помилки. Заданий результат був отриманий успішно.

					КНТЕУ 121 063-09.МР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання випускного кваліфікаційного проекту було розглянуте питання розробки та створення системи для організації роботи підприємства, якою виступає система ідентифікації та трекінгу особи.

Було проведено аналіз розвитку сучасних технологій та обґрунтовано зроблений вибір засобів реалізації обраних модулів. Здійснено детальний аналіз аналогічних програмних рішень, що дозволило провести підбір найбільш ефективних моделей для реалізації. Визначено актуальність питання, сучасний стан та перспективу розвитку подібних систем.

В результаті було розроблено систему відповідно до розробленого технічного завдання. Її якість була досягнута за рахунок проведеного пошуку та підбору найбільш ефективних та якісних вибірок для тренування та налаштування моделей розпізнання. Для системи було проведено тренування на тренувальних та валідаційних вибірках. Також було виконано оцінку ефективності реалізованого рішення, де продукт був порівняний з аналогічними. Система була реалізована через Jupyter Notebook та Sublime Text мовою програмування Python. Робота виконана на замовлення кампанії та наразі знаходиться на етапі бета-тесту.

Перевагами даної розробки є те, що більшість подібних програмних продуктів у світі створено для виключно комерційних цілей, ціни на них є дуже високими а також вони потребують постійної підтримки з боку сертифікованих спеціалістів. Також вони визначені для конкретних задач, тоді як дану розробку можна перепрофілювати на іншу – просто перевчити на тренуваннях. У перспективі є чимало можливостей для розвитку.

					<i>КНТЕУ 121 063-09.MP</i>			
					<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>ВП</i>	<i>47</i>	<i>49</i>
Зав. каф.	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, 63 група		
Керівник	Криворучко О.В.			30.10.20				
Гарант	Криворучко О.В.			30.10.20				
Розробив	Опенько Д.П.			30.10.20	<i>Висновки та пропозиції</i>			

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Машинне навчання [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F.
2. Основні поняття розпізнавання образів [Електронний ресурс] – Режим доступу до ресурсу: http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_01.pdf.
3. Application Programming Interface [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Application_programming_interface.
4. Інтелектуальний аналіз даних [Електронний ресурс] – Режим доступу до ресурсу: <http://evgavrilenko.ucoz.ru/index/lekciji/0-143>.
5. Коефіцієнт подібності [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B5%D1%84%D1%96%D1%86%D1%96%D1%94%D0%BD%D1%82_%D0%BF%D0%BE%D0%B4%D1%96%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%96.
6. ArcFace: Additive Angular Margin Loss for Deep Face Recognition [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1801.07698v3.pdf>.
7. (Deep) Convolutional Neural Network (DCNN/CNN) [Електронний ресурс] – Режим доступу до ресурсу:

					<i>КНТЕУ 121 063-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства</i>	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			30.10.20		СПД	48	49
Керівник	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, 6з група		
Гарант	Криворучко О.В.			30.10.20				
Розробив	Опенько Д.П.			30.10.20	<i>Список використаних джерел</i>			

[http://primo.ai/index.php?title=\(Deep\)_Convolutional_Neural_Network_\(DCNN/CNN\)](http://primo.ai/index.php?title=(Deep)_Convolutional_Neural_Network_(DCNN/CNN)).

8. Машинне навчання [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F.
9. Основні поняття розпізнавання образів [Електронний ресурс] – Режим доступу до ресурсу: http://om.univ.kiev.ua/users_upload/15/upload/file/pr_lecture_01.pdf.
10. Application Programming Interface [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Application_programming_interface.
11. Інтелектуальний аналіз даних [Електронний ресурс] – Режим доступу до ресурсу: <http://evgavrilenko.ucoz.ru/index/lekciji/0-143>.
12. Коефіцієнт подібності [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B5%D1%84%D1%96%D1%86%D1%96%D1%94%D0%BD%D1%82_%D0%BF%D0%BE%D0%B4%D1%96%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D1%96.
13. ArcFace: Additive Angular Margin Loss for Deep Face Recognition [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://arxiv.org/pdf/1801.07698v3.pdf>.
14. (Deep) Convolutional Neural Network (DCNN/CNN) [Електронний ресурс] – Режим доступу до ресурсу: [http://primo.ai/index.php?title=\(Deep\)_Convolutional_Neural_Network_\(DCNN/CNN\)](http://primo.ai/index.php?title=(Deep)_Convolutional_Neural_Network_(DCNN/CNN)).

					<i>КНТЕУ 121 063-09.МР</i>	Аркуш
						49
Зм.	Аркуш	№ докум	Підпис	Дата		

ТЕХНІЧНЕ ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ОРГАНІЗАЦІЇ РОБОТИ ПІДПРИЄМСТВА

1. Терміни початку та закінчення розробки програмного забезпечення інтелектуальної системи організації роботи підприємства.

Таблиця 1.1.

Календарний план розробки

№ пор.	Назва завдань етапів випускного кваліфікаційного проекту	Початок	Кінець
1.	Аналіз вимог та принципів функціонування біометричної мережі	29.06.2020	12.07.2019
2.	Створення технічного завдання	15.07.2020	26.09.2020
3.	Проектування інтелектуальної системи організації роботи підприємства	25.07.2020	15.08.2020
4.	Вибір мови програмування	-	-
5.	Визначення архітектури	-	-
6.	Розробка структури та загальної концепції побудови	24.08.2020	25.10.2020
7.	Архітектура нейронних мереж	-	-
8.	Конструювання програмного забезпечення	25.08.2020	25.10.2020
9.	Перевірка ефективності роботи	26.10.2020	3.11.2020
10.	Розгортання програмного продукту	25.10.2020	4.10.2020
11.	Документація	28.10.2020	2.11.2020

					<i>КНТЕУ 121 06з-09.МР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмного забезпечення інтелектуальної системи організації роботи підприємства Технічне завдання	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			30.10.20		ТЗ	50	49
Керівник	Криворучко О.В.			30.10.20		Факультет інформаційних технологій 2м курс, 6з група		
Гарант	Криворучко О.В.			30.10.20				
Розробив	Опенько Д.П.			30.10.20				

2. Метою роботи

є дослідження розвитку сучасних методів ідентифікації осіб та їх ефективного застосування у створенні власного успішного програмного рішення.

Аби мати змогу гарно організувати робочий процес на підприємстві та контролювати його, потрібно було би гарно мати зручно дані про працівників, як от: їх автоматична ідентифікація у просторі. При звірці з часом, це дозволить:

- контролювати виконання робочих завдань;
- аналізувати робочий час людей, чи працівник належно ставиться до своїх обов'язків (працює), чи займається іншими справами не у свою перерву;
- мати контроль доступу до спеціальних приміщень аби не було змоги потрапити, наприклад, до складу чи кабінету начальника непоміченим та можна було встановити особу.

Це дозволить вчасно прийняти міри у випадку необхідності, навчить бути більш обізнаним у своїх працівниках та їх звичках, та краще контролювати робочий процес.

3. Аналіз вимог до програмного забезпечення

Завдяки аналізу вимог та характеристик аналогічних програмних розробок були визначені основні варіанти використання даного програмного забезпечення, що відображають функціонал розроблюваної системи. Серед акторів можна виділити тільки систему, оскільки розробка не потребує користувацьких або адміністративних функцій.

Основний функціонал модулю:

- отримання медіафайлу через API пристрою, до якого підключений;

					КНТЕУ 121 063-09.МР	Аркуш
						51
Зм.	Аркуш	№ докум	Підпис	Дата		

- розпізнавання образу людини із медіафайлу;
- розпізнавання характеристик зовнішності особи з образу;
- класифікація характеристик зовнішності;
- перевірка на збіги із БД(за наявності БД користувача);
- формування результуючої видач
- передача результату через API модулю на пристрій.

Опис варіантів використання наведений у таблиці переліку варіантів використання.

Таблиця 1.2.

Основні варіанти використання

ID	Актор	Назва/Розділ	Передумови	Опис
UC01	Система	Отримання фото через API пристрою/Робота з даними	Ендпойнт з API пристрою прописаний при підключенні модуля. Пристрій може надати фото.	Модуль отримує цифровий файл – фото від пристрою, до якого інтегрований.
UC02	Система	Отримання відеоряду через API пристрою/Робота з даними	Ендпойнт з API пристрою прописаний при підключенні модуля. Пристрій може надати відеоряд.	Модуль приймає стрімінг файлу – відео від пристрою, до якого інтегрований.
UC03	Система	Кадрування відеоряду/Робота з даними	Модуль отримав відеоряд із пристрою.	Модуль ділить відеоряд на кадри – самостійні цифрові файли.
UC04	Система	Розпізнавання образу людини із кадру/Машинне навчання	Модуль має єдиний кадр для подальшої обробки.	Модель МН модуля розпізнає образ людини на кадри.

Продовження таблиці 1.2.

UC05	Система	Розпізнавання образів характеристик зовнішності/ Машинне навчання	Образ обличчя людини успішно розпізнано.	Модель МН модуля розпізнає образи рис обличчя людини на кадрі.
------	---------	---	--	--

UC06	Система	Класифікація віку/Нейронна мережа	Образи рис обличчя людини успішно розпізнано.	НМ модуля класифікує вік за рисами обличчя.
------	---------	-----------------------------------	---	---

UC07	Система	Класифікація статі/Нейронна мережа	Образи рис обличчя людини успішно розпізнано.	НМ модуля класифікує стать за рисами обличчя.
------	---------	------------------------------------	---	---

UC08	Система	Класифікація емоцій/Нейронна мережа	Образи рис обличчя людини успішно розпізнано.	НМ модуля класифікує емоції за рисами обличчя.
------	---------	-------------------------------------	---	--

UC09	Система	Перевірка на збіги з БД користувача/ Нейронна мережа	Користувачка БД доступна та підключена через відповідний ендпоінт.	НМ модуля порівнює класифікований образ особи із датасетом із БД.
------	---------	--	--	---

UC10	Система	Видача результату/ Робота з даними	Принаймні одна перевірка на розпізнавання пройдена.	Модуль формалізує результати і видає через власний API до пристрою.
------	---------	------------------------------------	---	---

Опираючись на дані з наведеної таблиці варіантів використання, стало можливо представити зв'язки між ними на структурній схемі варіантів використання (рисунок 1.1).

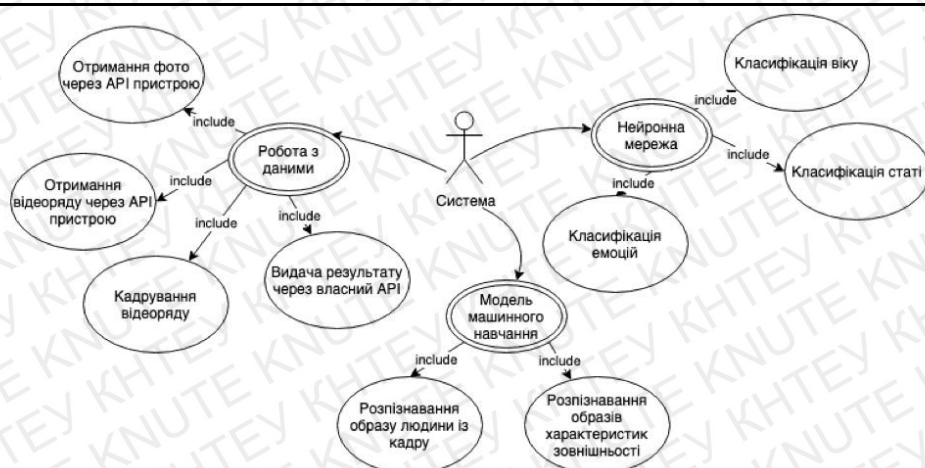


Рис. 1.1. Схема структурна варіантів використання

4. Розробка функціональних вимог

Маючи дані з таблиці варіантів використання, наступні функціональні вимоги можуть бути сформульовані (наведені у таблиці 1.3).

Таблиця 1.3.

Функціональні вимоги

Розділ	Функціональна вимога	Пріоритет
Робота з даними	Модуль дозволяє під'єднатися до стороннього API для отримання даних.	Високий
	Модуль дозволяє використовувати дані з користувацької БД(за наявності).	Середній
	Модуль має приймати дані у форматі фото.	Високий
	Модуль має приймати дані у форматі потокового відео.	Високий
	Модуль має зберігати потокове відео в буфері для подальшого кадрування.	Високий
	Модуль має кадрувати відеоряд для отримання окремого кадру(зображення).	Високий
	Модуль має збирати результати роботи нейронної мережі для формалізованої видачі.	Високий

Продовження таблиці 1.3.

	Модуль має власний API.	Середній
Машинне навчання	Модель МН модулю навчається розпізнавати образи обличчя людей.	Високий
	Модель МН модулю навчається розпізнавати образи окремих рис обличчя людей.	Високий
Нейронна мережа	НМ модуля класифікує вік за рисами розпізнаного обличчя.	Високий
	НМ модуля класифікує стать за рисами розпізнаного обличчя.	Високий
	НМ модуля класифікує емоції за рисами розпізнаного обличчя.	Високий
	НМ модуля використовує датасет користувацької БД для порівняння образу обличчя.	Середній

Базуючись на інформації, наведеній у таблиці 1.3, була сформована модель функціональних вимог, що наведена на рисунку 1.2.

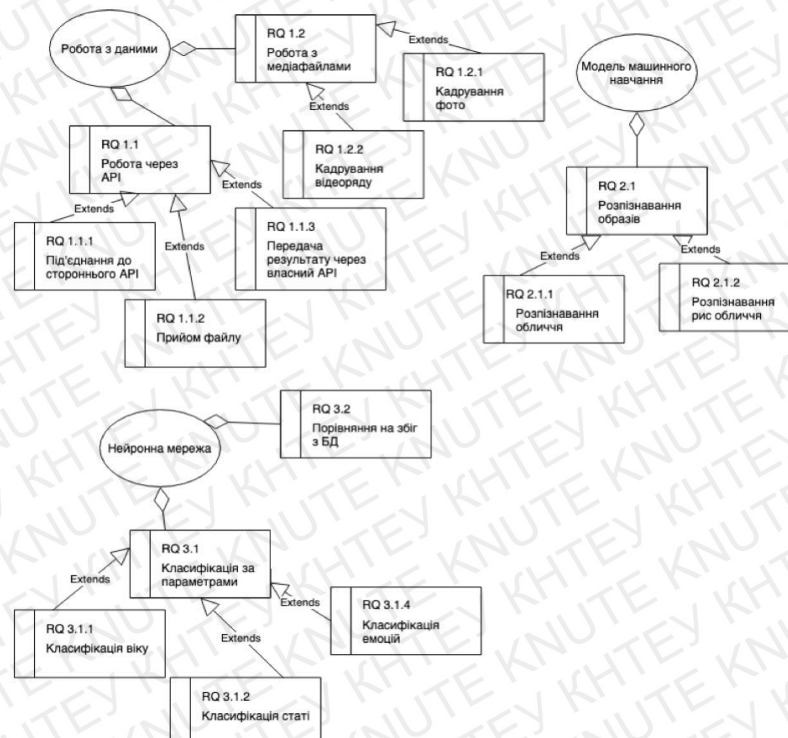


Рис. 1.2. Схема структурна моделі функціональних вимог

Отже, керуючись наявними варіантами використання (табл. 1.2) та моделлю функціональних вимог (рис. 1.2), можлива генерація матриці трасування вимог (рис. 1.3), що представляє зв'язки між варіантами використання системи для користувача та вимоги до функціоналу системи, виділених за допомогою користувацьких вимог.

	RQ 1.1	RQ 1.1.1	RQ 1.1.2	RQ 1.1.3	RQ 1.2	RQ 1.2.1	RQ 1.2.2	RQ 2.1	RQ 2.1.1	RQ 2.1.2	RQ 3.1	RQ 3.1.1	RQ 3.1.2	RQ 3.1.3	RQ 3.2
UC01	X	X	X												
UC02	X	X	X												
UC03					X	X	X								
UC04								X	X						
UC05								X		X					
UC06											X	X			
UC07											X		X		
UC08											X			X	
UC09															X
UC10				X											

Рис. 1.3. Матриця трасування вимог

5. Розробка нефункціональних вимог

Кінцевий продукт має видавати вихідні дані однакового формату незалежно від системи, в яку інтегрується. Модуль цілодобово має бути доступним для всіх користувачів, проте основна цільова аудиторія являє собою бізнес-структури з постійним потоком клієнтів.

Функціонал має бути простим та інтуїтивно зрозумілим.

Модуль має отримати доступ до цифрових файлів через API програмного забезпечення пристрою, до якого інтегрується.

Модуль не має збирати особисту інформацію користувачів, окрім фото.

Модуль у режимі локального використання має можливість функціонувати повністю автономно, що дозволяє опустити процес розробки серверної частини. Через це постає вимога підбору найефективнішого алгоритму машинного навчання для класифікації і розпізнавання на сьогоднішній день.

Модуль у режимі серверного використання рекомендовано реалізувати із урахуванням можливого навантаження від запитів користувачів та із можливістю масштабування для випадків розширення функціоналу або перехід до стану самостійної системи.

6. Постановка комплексу завдань модулю

Призначенням даного модулю є автоматизація розпізнавання об'єкту із наданого цифрового графічного файлу та його класифікація на предмет характеристик зовнішності людини. Характеристики зовнішності, що розглядаються в даному випадку:

- стать;
- приблизний вік;
- емоційний стан(настрій);
- порівняння з БД на збіг із даними про окрему особу.

Даний модуль має інтегруватися до операційного програмного забезпечення банкоматів, оплатних терміналів, вендінг-автоматів, систем безпеки місць масового скупчення людей, тощо.

Для досягнення поставленої мети, модуль має вирішувати наступні задачі:

- обробка цифрового файлу (фото чи відео);
- кадрування;
- розмітка;
- детекція об'єкта;
- класифікація об'єкта згідно правил із навчального датасету;
- видача результату в уніфікованій формі.

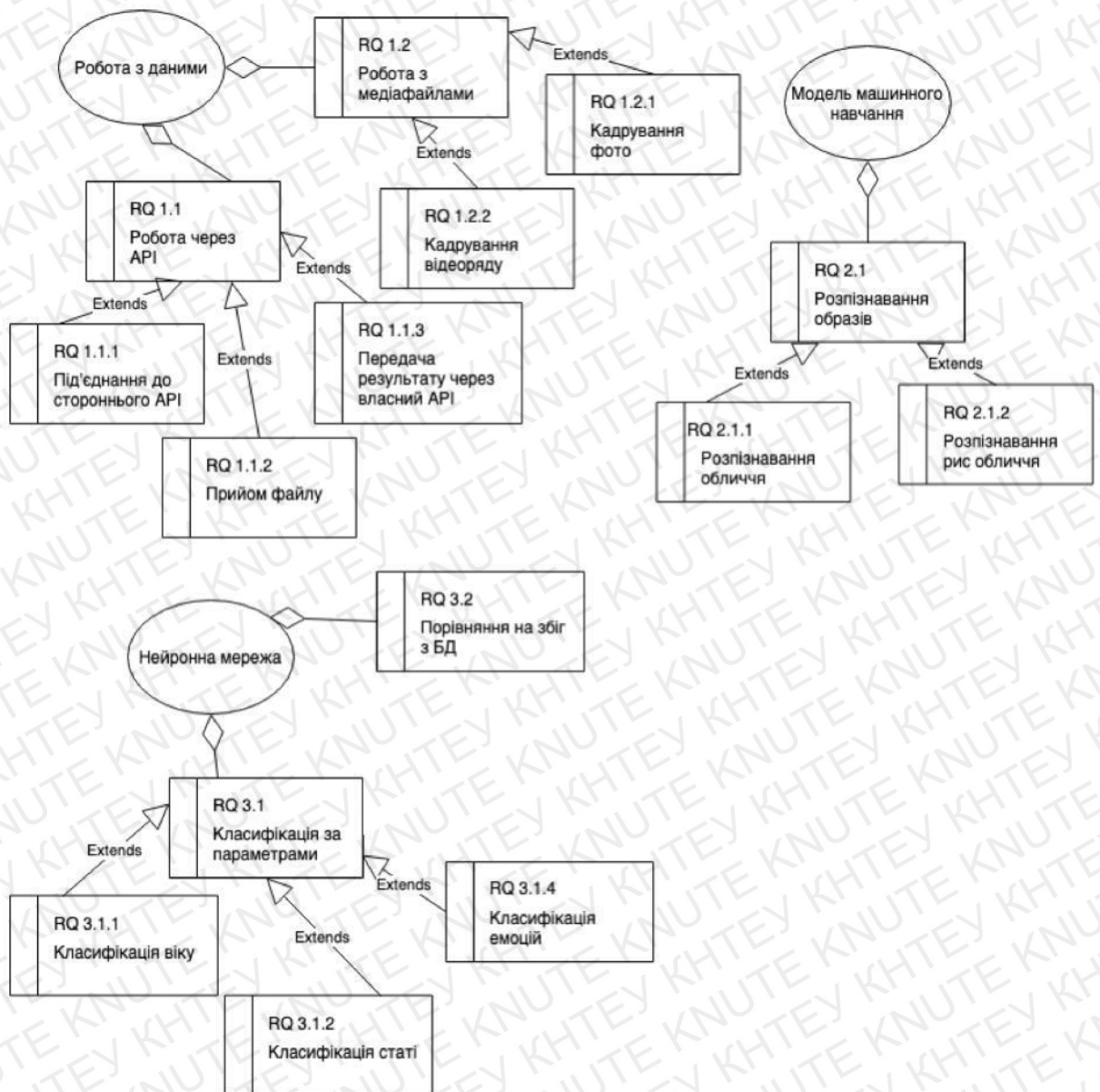
Відповідно, для досягнення цього, розроблені моделі мають бути натреновані на мультикласових датасетах.

					КНТЕУ 121 063-09.МР	Аркуш
						57
Зм.	Аркуш	№ докум	Підпис	Дата		

ДОДАТКИ

Додаток А

Структурна схема моделі функціональних вимог



Опис програми

```

from torch.nn
import Linear,
Conv2d,
BatchNorm1d,
BatchNorm2d,
PReLU,
ReLU,
Sigmoid,
Dropout, \
MaxPool2d, AdaptiveAvgPool2d, Sequential, Module, Parameter

import torch
from collections import namedtuple
import math

##### Original Arcface Model
#####

class Flatten(Module):
    def forward(self, input):
        return input.view(input.size(0), -1)

def l2_norm(input, axis=1):
    norm = torch.norm(input, 2, axis, True)
    output = torch.div(input, norm)
    return output

class SEModule(Module):
    def __init__(self, channels, reduction):
        super(SEModule, self).__init__()
        self.avg_pool = AdaptiveAvgPool2d(1)
        self.fc1 = Conv2d(
            channels, channels // reduction, kernel_size=1, padding=0, bias=False)

```


Продовження дод. Б

```
self.relu = ReLU(inplace=True)
self.fc2 = Conv2d(
    channels // reduction, channels, kernel_size=1, padding=0, bias=False)
self.sigmoid = Sigmoid()

def forward(self, x):
    module_input = x
    x = self.avg_pool(x)
    x = self.fc1(x)
    x = self.relu(x)
    x = self.fc2(x)
    x = self.sigmoid(x)
    return module_input * x

class bottleneck_IR(Module):
    def __init__(self, in_channel, depth, stride):
        super(bottleneck_IR, self).__init__()
        if in_channel == depth:
            self.shortcut_layer = MaxPool2d(1, stride)
        else:
            self.shortcut_layer = Sequential(
                Conv2d(in_channel, depth, (1, 1), stride, bias=False), BatchNorm2d(depth))
        self.res_layer = Sequential(
            BatchNorm2d(in_channel),
            Conv2d(in_channel, depth, (3, 3), (1, 1), 1, bias=False), PReLU(depth),
            Conv2d(depth, depth, (3, 3), stride, 1, bias=False), BatchNorm2d(depth))

    def forward(self, x):
        shortcut = self.shortcut_layer(x)
        res = self.res_layer(x)
        return res + shortcut

class bottleneck_IR_SE(Module):
    def __init__(self, in_channel, depth, stride):
        super(bottleneck_IR_SE, self).__init__()
        if in_channel == depth:
```


Продовження дод. Б

```
self.shortcut_layer = MaxPool2d(1, stride)
else:
    self.shortcut_layer = Sequential(
        Conv2d(in_channel, depth, (1, 1), stride, bias=False),
        BatchNorm2d(depth))
self.res_layer = Sequential(
    BatchNorm2d(in_channel),
    Conv2d(in_channel, depth, (3, 3), (1, 1), 1, bias=False),
    PReLU(depth),
    Conv2d(depth, depth, (3, 3), stride, 1, bias=False),
    BatchNorm2d(depth),
    SEModule(depth, 16)
)

def forward(self, x):
    shortcut = self.shortcut_layer(x)
    res = self.res_layer(x)
    return res + shortcut

class Bottleneck(namedtuple('Block', ['in_channel', 'depth', 'stride'])):
    """A named tuple describing a ResNet block."""

def get_block(in_channel, depth, num_units, stride=2):
    return [Bottleneck(in_channel, depth, stride)] + [Bottleneck(depth, depth, 1) for i in range(num_units - 1)]

def get_blocks(num_layers):
    if num_layers == 50:
        blocks = [
            get_block(in_channel=64, depth=64, num_units=3),
            get_block(in_channel=64, depth=128, num_units=4),
            get_block(in_channel=128, depth=256, num_units=14),
            get_block(in_channel=256, depth=512, num_units=3)
        ]
    elif num_layers == 100:
        blocks = [
            get_block(in_channel=64, depth=64, num_units=3),
            get_block(in_channel=64, depth=128, num_units=13),
```

Продовження дод. Б

```

        get_block(in_channel=128, depth=256, num_units=30),
        get_block(in_channel=256, depth=512, num_units=3)
    ]
    elif num_layers == 152:
        blocks = [
            get_block(in_channel=64, depth=64, num_units=3),
            get_block(in_channel=64, depth=128, num_units=8),
            get_block(in_channel=128, depth=256, num_units=36),
            get_block(in_channel=256, depth=512, num_units=3)
        ]
    return blocks

class Backbone(Module):
    def __init__(self, num_layers, drop_ratio, mode='ir'):
        super(Backbone, self).__init__()
        assert num_layers in [50, 100, 152], 'num_layers should be 50,100, or 152'
        assert mode in ['ir', 'ir_se'], 'mode should be ir or ir_se'
        blocks = get_blocks(num_layers)
        if mode == 'ir':
            unit_module = bottleneck_IR
        elif mode == 'ir_se':
            unit_module = bottleneck_IR_SE
        self.input_layer = Sequential(Conv2d(3, 64, (3, 3), 1, 1, bias=False),
                                      BatchNorm2d(64),
                                      PReLU(64))
        self.output_layer = Sequential(BatchNorm2d(512),
                                       Dropout(drop_ratio),
                                       Flatten(),
                                       Linear(512 * 7 * 7, 512),
                                       BatchNorm1d(512))

        modules = []
        for block in blocks:
            for bottleneck in block:
                modules.append(
                    unit_module(bottleneck.in_channel,
                                bottleneck.depth,
                                bottleneck.stride))
        self.body = Sequential(*modules)

    def forward(self, x):
        x = self.input_layer(x)
        x = self.body(x)

```

Продовження дод. Б


```

x = self.output_layer(x)
return l2_norm(x)

##### MobileFaceNet
#####

class Conv_block(Module):
    def __init__(self, in_c, out_c, kernel=(1, 1), stride=(1, 1), padding=(0, 0), groups=1):
        super(Conv_block, self).__init__()
        self.conv = Conv2d(in_c, out_channels=out_c, kernel_size=kernel, groups=groups, stride=stride, padding=padding,
                             bias=False)
        self.bn = BatchNorm2d(out_c)
        self.prelu = PReLU(out_c)

    def forward(self, x):
        x = self.conv(x)
        x = self.bn(x)
        x = self.prelu(x)
        return x

class Linear_block(Module):
    def __init__(self, in_c, out_c, kernel=(1, 1), stride=(1, 1), padding=(0, 0), groups=1):
        super(Linear_block, self).__init__()
        self.conv = Conv2d(in_c, out_channels=out_c, kernel_size=kernel, groups=groups, stride=stride, padding=padding,
                             bias=False)
        self.bn = BatchNorm2d(out_c)

    def forward(self, x):
        x = self.conv(x)
        x = self.bn(x)
        return x

```

Продовження дод. Б


```

class Depth_Wise(Module):
    def __init__(self, in_c, out_c, residual=False, kernel=(3, 3), stride=(2, 2), padding=(1, 1), groups=1):
        super(Depth_Wise, self).__init__()
        self.conv = Conv_block(in_c, out_c=groups, kernel=(1, 1), padding=(0, 0), stride=(1, 1))
        self.conv_dw = Conv_block(groups, groups, groups=groups, kernel=kernel, padding=padding, stride=stride)
        self.project = Linear_block(groups, out_c, kernel=(1, 1), padding=(0, 0), stride=(1, 1))
        self.residual = residual

    def forward(self, x):
        if self.residual:
            short_cut = x
            x = self.conv(x)
            x = self.conv_dw(x)
            x = self.project(x)
            if self.residual:
                output = short_cut + x
            else:
                output = x
            return output
        else:
            return x

class Residual(Module):
    def __init__(self, c, num_block, groups, kernel=(3, 3), stride=(1, 1), padding=(1, 1)):
        super(Residual, self).__init__()
        modules = []
        for _ in range(num_block):
            modules.append(
                Depth_Wise(c, c, residual=True, kernel=kernel, padding=padding, stride=stride, groups=groups))
        self.model = Sequential(*modules)

    def forward(self, x):
        return self.model(x)

class MobileFaceNet(Module):
    def __init__(self, embedding_size):

```

Продовження дод. Б

```

super(MobileFaceNet, self).__init__()
self.conv1 = Conv_block(3, 64, kernel=(3, 3), stride=(2, 2), padding=(1, 1))
self.conv2_dw = Conv_block(64, 64, kernel=(3, 3), stride=(1, 1), padding=(1, 1),
groups=64)
self.conv_23 = Depth_Wise(64, 64, kernel=(3, 3), stride=(2, 2), padding=(1, 1),
groups=128)
self.conv_3 = Residual(64, num_block=4, groups=128, kernel=(3, 3), stride=(1, 1),
padding=(1, 1))
self.conv_34 = Depth_Wise(64, 128, kernel=(3, 3), stride=(2, 2), padding=(1, 1),
groups=256)
self.conv_4 = Residual(128, num_block=6, groups=256, kernel=(3, 3), stride=(1, 1),
padding=(1, 1))
self.conv_45 = Depth_Wise(128, 128, kernel=(3, 3), stride=(2, 2), padding=(1, 1),
groups=512)
self.conv_5 = Residual(128, num_block=2, groups=256, kernel=(3, 3), stride=(1, 1),
padding=(1, 1))
self.conv_6_sep = Conv_block(128, 512, kernel=(1, 1), stride=(1, 1), padding=(0,
0))
self.conv_6_dw = Linear_block(512, 512, groups=512, kernel=(7, 7), stride=(1, 1),
padding=(0, 0))
self.conv_6_flatten = Flatten()
self.linear = Linear(512, embedding_size, bias=False)
self.bn = BatchNorm1d(embedding_size)

def forward(self, x):
    out = self.conv1(x)

    out = self.conv2_dw(out)

    out = self.conv_23(out)

    out = self.conv_3(out)

    out = self.conv_34(out)

    out = self.conv_4(out)

    out = self.conv_45(out)

```

Продовження дод. Б


```

        out = self.conv_5(out)

        out = self.conv_6_sep(out)

        out = self.conv_6_dw(out)

        out = self.conv_6_flatten(out)

        out = self.linear(out)

        out = self.bn(out)
        return l2_norm(out)

##### Arcface head
#####

class Arcface(Module):
    # implementation of additive margin softmax loss in https://arxiv.org/abs/1801.05599
    def __init__(self, embedding_size=512, classnum=51332, s=64., m=0.5):
        super(Arcface, self).__init__()
        self.classnum = classnum
        self.kernel = Parameter(torch.Tensor(embedding_size, classnum))
        # initial kernel
        self.kernel.data.uniform_(-1, 1).renorm_(2, 1, 1e-5).mul_(1e5)
        self.m = m # the margin value, default is 0.5
        self.s = s # scalar value default is 64, see normface
        https://arxiv.org/abs/1704.06369
        self.cos_m = math.cos(m)
        self.sin_m = math.sin(m)
        self.mm = self.sin_m * m # issue 1
        self.threshold = math.cos(math.pi - m)

    def forward(self, embeddings, label):
        # weights norm
        nB = len(embeddings)
        kernel_norm = l2_norm(self.kernel, axis=0)
        # cos(theta+m)
        cos_theta = torch.mm(embeddings, kernel_norm)
        # output = torch.mm(embeddings, kernel_norm)

```

Продовження дод. Б


```

cos_theta = cos_theta.clamp(-1, 1) # for numerical stability
cos_theta_2 = torch.pow(cos_theta, 2)
sin_theta_2 = 1 - cos_theta_2
sin_theta = torch.sqrt(sin_theta_2)
cos_theta_m = (cos_theta * self.cos_m - sin_theta * self.sin_m)
# this condition controls the theta+m should in range [0, pi]
# 0<=theta+m<=pi
# -m<=theta<=pi-m
cond_v = cos_theta - self.threshold
cond_mask = cond_v <= 0
keep_val = (cos_theta - self.mm) # when theta not in [0,pi], use cosface instead
cos_theta_m[cond_mask] = keep_val[cond_mask]
output = cos_theta * 1.0 # a little bit hacky way to prevent in_place operation on
cos_theta
idx_ = torch.arange(0, nB, dtype=torch.long)
output[idx_, label] = cos_theta_m[idx_, label]
output *= self.s # scale up in order to make softmax work, first introduced in
normface
return output

```

```

##### Cosface head
#####

```

```

class Am_softmax(Module):
    # implementation of additive margin softmax loss in https://arxiv.org/abs/1801.05599
    def __init__(self, embedding_size=512, classnum=51332):
        super(Am_softmax, self).__init__()
        self.classnum = classnum
        self.kernel = Parameter(torch.Tensor(embedding_size, classnum))
        # initial kernel
        self.kernel.data.uniform_(-1, 1).renorm_(2, 1, 1e-5).mul_(1e5)
        self.m = 0.35 # additive margin recommended by the paper
        self.s = 30. # see normface https://arxiv.org/abs/1704.06369

    def forward(self, embeddings, label):
        kernel_norm = l2_norm(self.kernel, axis=0)
        cos_theta = torch.mm(embeddings, kernel_norm)
        cos_theta = cos_theta.clamp(-1, 1) # for numerical stability
        phi = cos_theta - self.m

```

Продовження дод. Б

```

label = label.view(-1, 1) # size=(B,1)
index = cos_theta.data * 0.0 # size=(B,Classnum)

```

```
index.scatter_(1, label.data.view(-1, 1), 1)
index = index.byte()
output = cos_theta * 1.0
output[index] = phi[index] # only change the correct predicted output
output *= self.s # scale up in order to make softmax work, first introduced in
normface
return output
```