Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ РОЗПІЗНАВАННЯ КУЛІНАРНИХ СТРАВ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗДОРОВ'Я
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-20-2
Подшивалова О.Є.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Творошенко І.С.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Подшиваловій Ользі Євгеніївні
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 03 червня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека машинного навчання з відкритим кодом PyTorch, відкритий набір даних з фото кулінарних страв iFood-2019, бібліотека кросплатформних API з відкритим кодом Kivy.

4. Перелік питань, що потрібно опрацювати в роботі

1. Аналіз існуючих застосунків для розпізнавання кулінарних страв в Україні та за кордоном.

2. Особливості розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я.

3. Розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми, постановка задачі, етапи виконання роботи, отриманні результати, перспективи для подальшої роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-17.04.24	
3	Аналіз літератури з досліджуваної проблеми	18.04.24-20.04.24	
4	Аналіз існуючих методів розпізнавання та класифікації кулінарних страв	21.04.24-25.04.24	
5	Розробка методу розпізнавання кулінарних страв на зображенні	26.04.24-14.05.24	
6	Програмна реалізація	15.05.24-24.05.24	
7	Оформлення пояснювальної записки	25.05.24-03.06.24	
8	Перевірка на плагіат	10.06.24	
9	Рецензування	10.06.24	
10	Підготовка презентації та доповіді	05.06.24-11.06.24	
11	Занесення роботи в електронний архів	10.06.24	
12	Попередній захист кваліфікаційної роботи	12.06.24	

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____ _____ доц. Творошенко І.С.____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 59 с., 24 рис., 32 джерела.

ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, ГЛИБОКЕ НАВЧАННЯ, РОЗПІЗНАВАННЯ ЇЖІ, МАШИННЕ НАВЧАННЯ, КЛАСИФІКАЦІЯ ЗОБРАЖЕНЬ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ.

Об’єктом роботи є процес розроблення мобільного застосунку розпізнавання кулінарних страв.

Метою роботи є розроблення за допомогою згорткових нейронних мереж мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров’я.

Проаналізовано різні методи розпізнавання зображень у мобільних застосунках, що є наразі найбільш використовуваними. Обрано підхід до виконання поставленої задачі, досліджено предметні області, за допомогою інструментів бібліотеки глибокого навчання PyTorch побудовано R-CNN, а також проаналізовано результати її тренування.

У результаті виконання роботи розроблено мобільний застосунок для розпізнавання кулінарних страв.

CONVOLUTIONAL NEURAL NETWORKS, DEEP LEARNING, FOOD RECOGNITION, MACHINE LEARNING, IMAGE CLASSIFICATION, IMAGE RECOGNITION.

The object of the work is the process of developing an application for recognizing images of culinary dishes.

The purpose of the work is to develop a software application using convolutional neural networks for the recognition of culinary dishes.

Different methods of image recognition in mobile applications, which are currently the most used, are analyzed. Based on this, an approach to the task was chosen, the subject areas were investigated, and R-CNN was built using the tools of the PyTorch deep learning library, and the results of its training were analyzed.

As a result of the work, a mobile application for recognition of culinary dishes was developed.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 6
Вступ 7
1 Аналіз існуючих застосунків для розпізнавання кулінарних страв в Україні та за кордоном 9
1.1 Сучасний стан розвитку застосунків для розпізнавання кулінарних страв в Україні та за кордоном 9
1.2 Аналіз літературних джерел щодо існуючих підходів розпізнавання кулінарних страв 18
1.3 Постановка задачі 21
2 Особливості розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я 23
2.1 Механізми оброблення зображень в мобільних застосунках 23
2.2 Принципи розпізнавання кулінарних страв у мобільних застосунках 29
2.3 Застосування методів системного аналізу для виявлення особливостей предметної області 33
2.4 Методика розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я 35
3 Розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я 37
3.1 Вибір інструментальних засобів для створення застосунку 37
3.1.1 Основні інструменти для створення застосунку 37
3.1.2 Використання PyTorch для побудови моделі нейронної мережі 38
3.1.3 Вибір інструментів для клієнтської частини застосунку 40
3.2 Етапи розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров'я 40
3.3 Тестування розробленого мобільного застосунку та аналіз результатів 51
3.4 Перспективи подальшої роботи 54
Висновки 55
Перелік джерел посилання 56

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN – Convolutional Neural Network (згорткова нейронна мережа)

R-CNN – Region-based Convolutional Neural Network (згорткова нейронна мережа на основі регіонів)

ResNet – Residual Network (резидуальна мережа)

AIM – Automated Machine Learning Interpretation Methods (автоматизовані методи інтерпретації машинного навчання)

VGG – Visual Geometry Group (візуальна геометрична група)

GIST – Global Image Structural Textural (глобальна структурна текстура зображення)

HoG – Histogram of Oriented Gradients (гістограма орієнтованих градієнтів)

LBP – Local Binary Patterns (локальні бінарні патерни)

SVM – Support Vector Machine (метод опорних векторів)

ViT – Vision Transformers (трансформери зору)

YOLO – You Only Look Once («Ви дивитесь тільки раз»)

ReLU – Rectified Linear Unit (випрямляючий лінійний блок)

RPN – Region Proposal Network (мережа пропозицій регіонів)

RoI – Region of Interest (регіон інтересу)

API – Application Programming Interface (інтерфейс прикладного програмування)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

ВСТУП

У сучасному світі інформаційні технології є невід'ємною частиною людського життя. Вони слугують незамінним помічником, який допомагає вирішувати ті чи інші задачі. Їх роботу можна побачити як у побуті, так і у специфічних сферах людської діяльності, наприклад, методи комп'ютерного обчислення широко використовуються у досить новій галузі як біоінформатика, яка спеціалізується на розшифруванні людського геному.

Серед тих технологій, які використовуються людиною у повсякденному житті, наразі стають усе більш популярними мобільні застосунки. Суттєвою їх перевагою є можливість носити деякий функціонал у кишені, оскільки він буде міститися у телефоні.

Існує велика кількість мобільних програмних платформ, які, так чи інакше, полегшують слідкування за здоров'ям та харчуванням людини: слідкування за калоріями, фітнес-трекери, слідкування за випитою водою та інше.

Розпізнавання образів зараз є перспективною галуззю для розроблення. Можливість навчити комп'ютер розуміти те, що є на поданому йому зображенні, відкриває перед розробниками високий потенціал використання такого методу у різних галузях людської діяльності. Такий підхід вже використовується для розпізнавання обличь в охоронних системах, в контроль за сільськогосподарськими рослинами (виявлення захворювань, а також дослідження ступені зрошення культур), у виявленні захворювань та у багатьох інших нішах.

Портативність мобільних застосунків та інноваційність методу розпізнавання образів не могли не об'єднатися, тож останнім часом програмних платформ, що містять у собі таке поєднання, з'являється на ринку інформаційних технологій все більше.

І хоча найбільш популярним напрямом у комп'ютерному зорі вважається розпізнавання обличь чи тварин, тема класифікації кулінарних

страв на зображенні є також досить цікавою для дослідження та розроблення різноманітних програмних платформ. Існує багато робіт, що так чи інакше торкаються цієї теми.

Розпізнавання їжі на фотографії може бути корисним насамперед у галузі здоров’я та краси. За допомогою такого підходу можна здійснювати контроль харчування користувача, а також розрахунок його раціону без зайвих складнощів та втрат часу, оскільки розрахунок вручну може потребувати великої кількості часу та знань.

Класифікація кулінарних страв вже використовується у мобільних застосунках. У більшості випадків така функція призначена для розрахунку калорійності раціону користувача.

Технічно такий підхід можна використовувати не тільки як інструмент для спрощення зниження ваги, але і для застереження користувачів про інші проблеми життєдіяльності, наприклад, про алергії чи якісь небезпечні захворюваннях.

1 АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ ДЛЯ РОЗПІЗНАВАННЯ КУЛІНАРНИХ СТРАВ В УКРАЇНІ ТА ЗА КОРДОНОМ

1.1 Сучасний стан розвитку застосунків для розпізнавання кулінарних страв в Україні та за кордоном

У сучасному світі все більше людей набувають проблеми зі здоров’ям внаслідок погіршення екологічного стану навколишнього середовища у світі, а також пришвидшення темпу життя, через що збільшується кількість факторів виникнення стресу та знижується якість прийомів їжі внаслідок зменшення на це часу. На базі цього з’являються нові випадки алергічних реакцій, спричинених різними подразниками, а також захворювання внутрішніх органів, лікування яких передбачає дотримання дієтичного харчування. За даними ВОЗ України наразі у світі на харчову алергію страждають 1-2% дорослих, а також 4-6% дітей [1].

Багато сучасних IT-компаній, наприклад, таких як Microsoft [2] та Google, виділяють частину ресурсів на дослідження та розробку систем, що відносяться до догляду за здоров’ям. Причому поле розробки таких компаній включає в себе не тільки інструменти для забезпечення персоналу в закладах сфери охорони здоров’я, але й задля індивідуальних потреб конкретної людини також.

Для полегшення дослідження стану організму та полегшення дотримання обмежень у харчуванні з кожним днем розробляється все більше різних застосунків, пристроїв та утиліт. Вони покликані для автоматичного зчитування різних показників організму людини на кшталт рівня цукру в крові, визначення оптимального часу прийомів їжі та виявлення вмісту потенційних алергенів у продуктах. Також слід виділити застосунки із добіркою низки інтерактивних рецептів, які підбираються персонально з урахуванням зазначених особливостей людини.

Найбільш популярним інструментом для догляду за собою та своїм здоров’ям є мобільний застосунок. Перевагами такого підходу є мобільність та зручність використання. У сучасному світі не можна й уявити життя без мобільного телефону, тож такий інструмент буде у людини завжди під рукою.

Наразі існує багато зручних застосунків для знаходження потенційного вмісту алергенів, які допомагають поліпшити життя людей, що мають алергію чи змушені дотримуватися певної дієти. Вони мають зручний дизайн та в них просто розібратися. Хоча повних аналогів, які могли б показати потенційний вміст алергенів при зчитуванні напряму зображення страви знайдено не було, слід зазначити, що існують застосунки, які частково виконують аналогічні функції та мають схожу мету.

Найбільш поширеним видом застосунків, подібних до розроблюваного, є застосунок для сканування штрих-коду або зчитуванні складу продукції напряму. Серед них найпопулярнішим інструментом є Food scanner (рис. 1.1).

Рисунок 1.1 – Інтерфейс застосунку Food scanner

Окрім досить зручного дизайну (рис. 1.1), він має низку функціональних переваг, таких як:

– сканування в реальному часі (застосунок виводить результат сканування одразу після потрапляння до поля зору камери штрих-коду продукту (рис. 1.2));

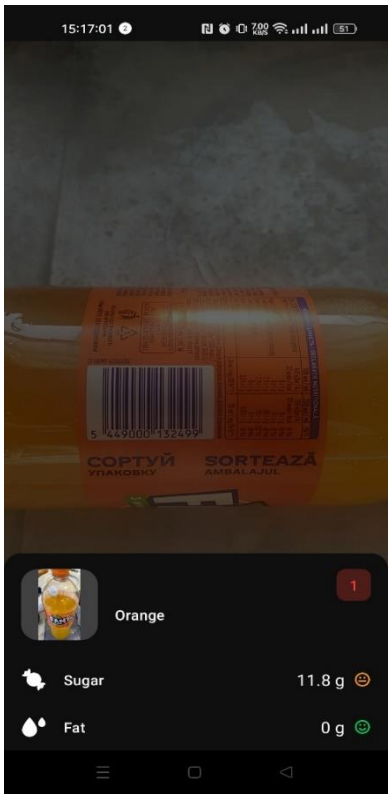


Рисунок 1.2 – Результат сканування продукту у застосунку Food scanner

– збір продуктів-алергенів у групи;

– додатковий вміст рецептів деяких страв, що не мають у вмісті алергенів (рис. 1.3).

Слід додатково приділити увагу другому пункту, оскільки такий підхід є досить зручним, оскільки користувачеві з алергією чи дієтою, яка передбачає вилучення певної категорії не доведеться шукати кожен продукт у великому списку окремо. Загалом, групування можна здійснювати за такими ознаками:

– за напрямом дієти;

– за домінуючим вмістом білків, жирів та вуглеводів;

– за належністю до категорії продуктів.

Також можна індивідуально формувати групу та заносити туди продукти власноруч.

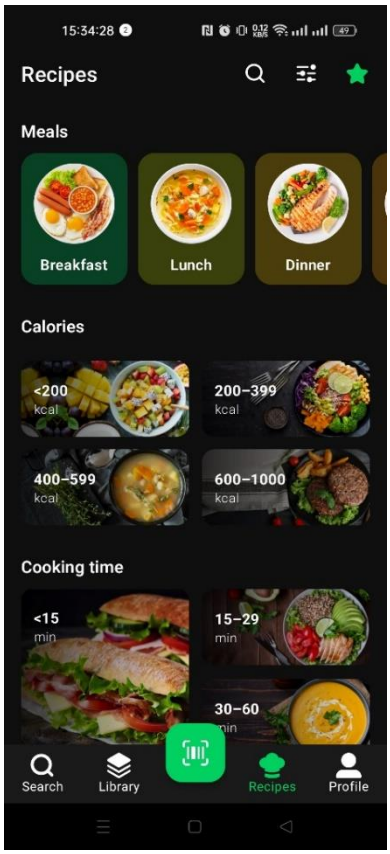


Рисунок 1.3 – Вкладення з рецептами у застосунку Food scanner

Однак, даний застосунок має декілька суттєвих проблем, серед яких є відсутність української мови інтерфейсу та обмежена база даних, яка не містить дані щодо маркування українських продуктів. Внаслідок вищезазначеного використовувати цей застосунок на території немає можливості.

Іншою низкою схожих за функціоналом є застосунки, що безпосередньо сканують склад товару з пакування. Серед таких найбільш досконалим є Avoid Food Additives. Він сканує склад продукту на пакуванні та віднаходить харчові добавки, які там присутні.

При реєстрації в програмній платформі користувач може обрати харчові добавки, яких треба уникати, і в подальшому скануванні вона складатиме рекомендацію щодо доцільності вживання такого продукту (рис. 1.4).

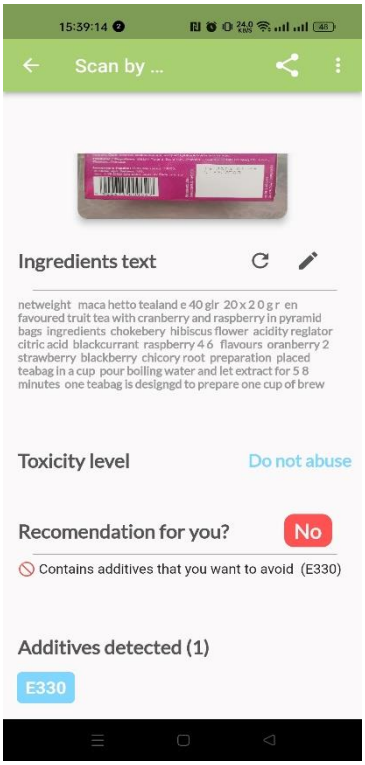


Рисунок 1.4 – Результат сканування складу продукту застосунком Avoid Food Additives

Серед переваг Avoid Food Additives є гнучкий вибір алергенів за захворюваннями та власноручне додавання харчових добавок у список уникнення (рис. 1.5, 1.6). Також суттєвою перевагою є знаходження перефразованих харчових добавок на упаковці (наприклад, коли Е330 позначається на упаковці як лимонна кислота).

Недоліком такого застосунку є розпізнавання складу продукту виключно англійською мовою, що ускладнює його використання в неангломовних країнах, оскільки маркування продуктів, призначених для внутрішнього ринку, в більшості випадків відбувається виключно державною мовою країни-виробника.

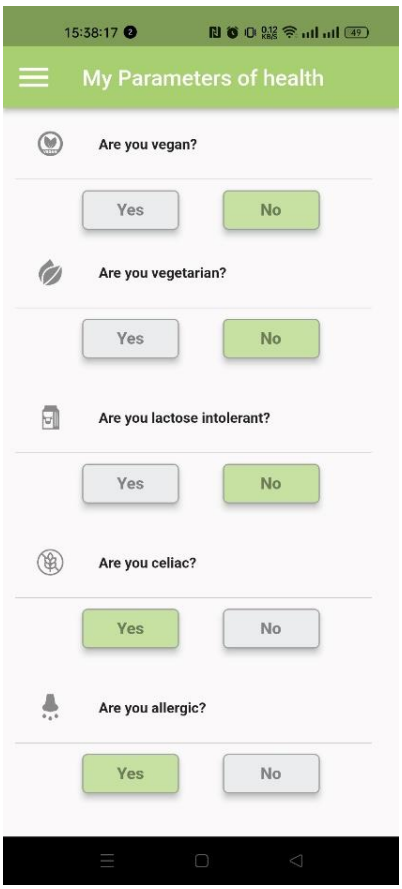


Рисунок 1.5 – Вибір уникання за групами вподобань чи захворювань у застосунку Avoid Food Additives

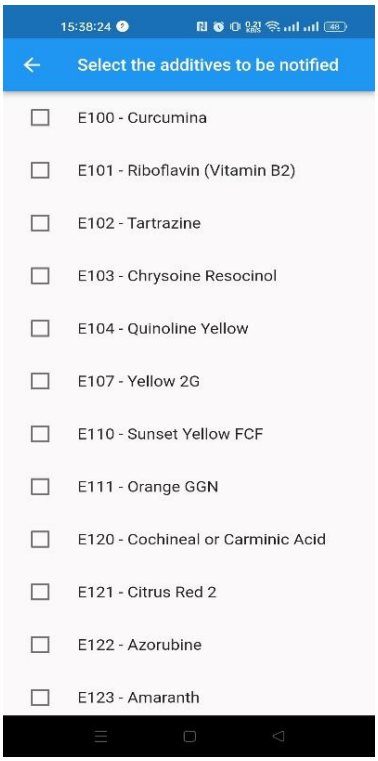


Рисунок 1.6 – Вибір уникання власноруч у застосунку Avoid Food Additives

Серед великої кількості застосунків, що ґрунтуються на підрахуванні калорій у стравах, дуже багато таких, що підпадають під розглянені вище за функціоналом типи. На їх фоні виділяється застосунок MyFitnessPal.

Інноваційним підходом цього застосунку є безпосереднє сканування продукту: для цього треба лише піднести продукт до камери і застосунок покаже список продуктів, які мають класифікаційні ознаки відсканованого (рис. 1.7).

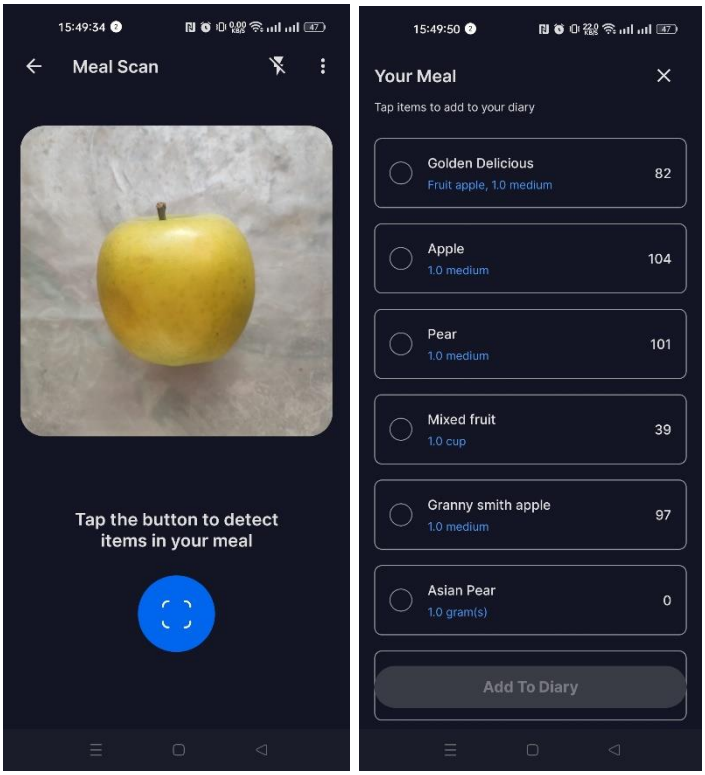
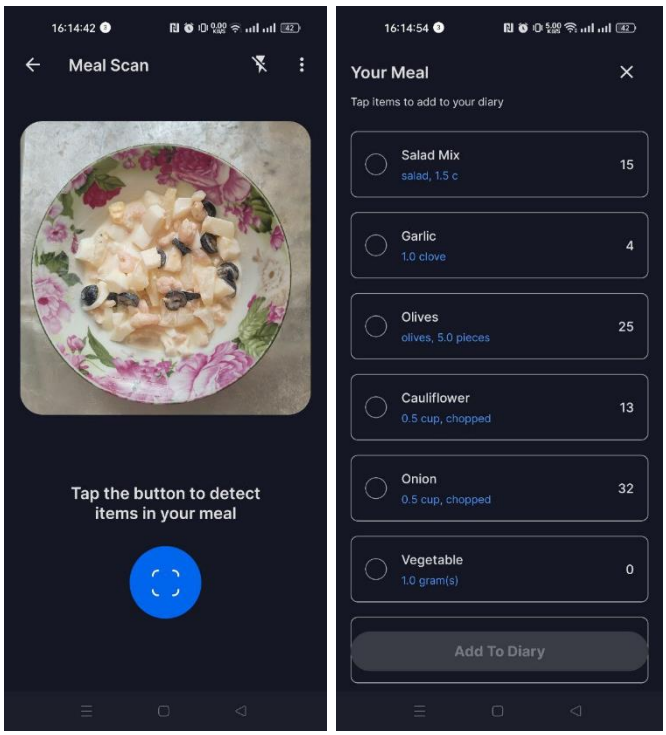


Рисунок 1.7 – Класифікація продукту застосунком MyFitnessPal

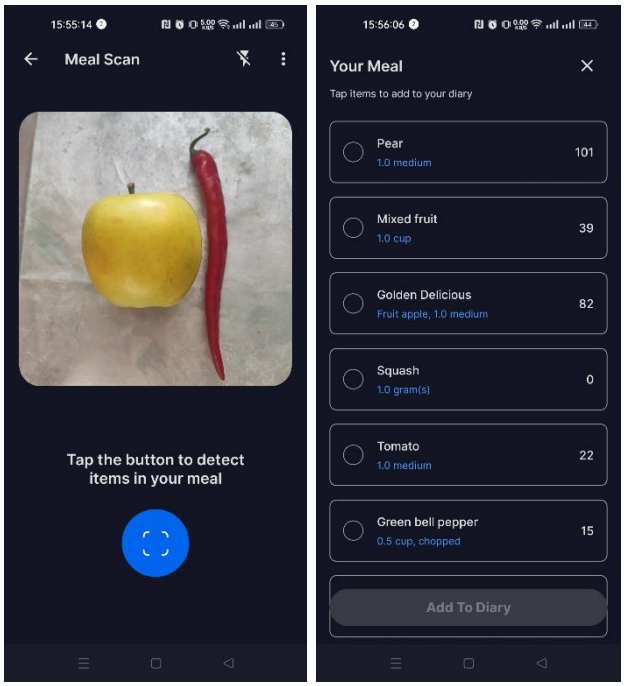
Також за допомогою даного застосунку можна класифікувати комплексні страви (рис. 1.8).

Суттєвою перевагою такого розпізнавання страв є подоланням мовного бар’єру та доланням проблем з малою базою даних, що зберігає штрих-коди, оскільки основні продукти виглядають однаково у більшості країн.

Недоробкою такого підходу є класифікація лише одного продукту у кадрі. При знаходженні декількох продуктів у кадрі застосунок має хибну класифікацію (рис. 1.9).

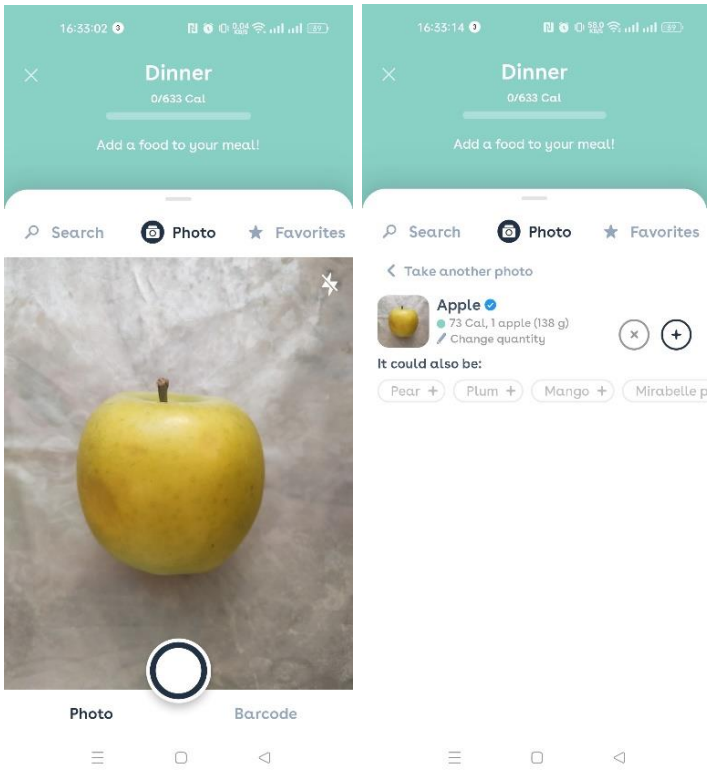


Рисунки 1.8 – Класифікація страви застосунком MyFitnessPal



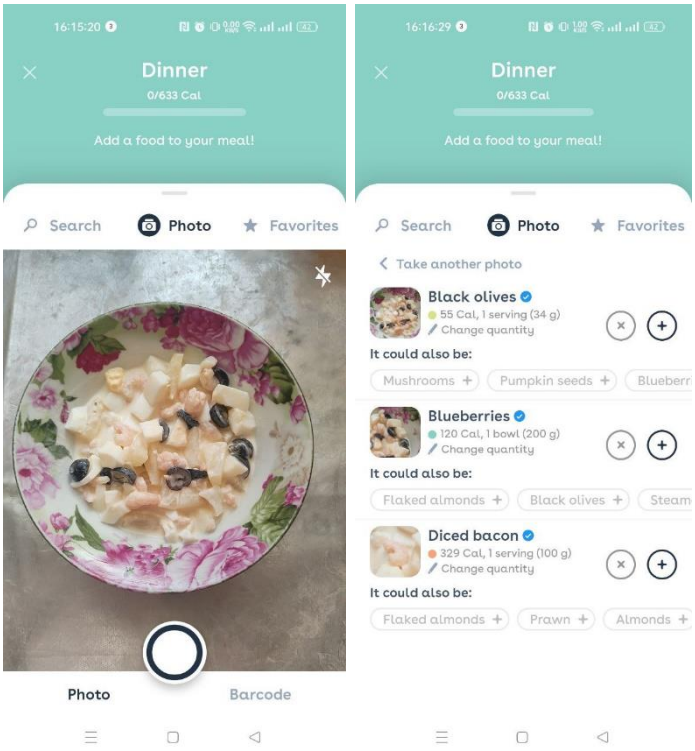
Рисунки 1.9 – Хибна класифікація декількох продуктів застосунком MyFitnessPal

Подібним до розглянутого вище є застосунок під назвою Foodvisor (рис. 1.10). Він також має функцію класифікації продуктів за фотографією, його суттєвою перевагою є розпізнавання декількох видів їжі в одному кадрі.



Рисунки 1.10 – Класифікація продукту застосунком Foodvisor

Слід зазначити також, що Foodvisor при класифікації видає не список продуктів, а той що найбільш схожий на відсканований (рис. 1.11).



Рисунки 1.11 – Класифікація страви застосунком Foodvisor

Отже, було розглянуто та проаналізовано функціонал застосунків, що стосуються розпізнавання продуктів та кулінарних страв, а також класифікацію їх за належністю до груп певних продуктів, що можуть бути обмежені чи вилучені з раціону при дотриманні дієтотерапії.

Можна побачити, що зараз на ринку застосунків, які призначені для контролю та дослідження раціону, існують платформи, покликані на підрахування збалансованості та калорійності продуктів. Слід зазначити, що більшість з них працюють за принципом сканування штрих-коду чи зчитування складу.

Щодо українських аналогів класифікації кулінарних страв, наразі не було знайдено застосунків з подібними функціями чи інформації щодо їх розроблення. Треба додати, що більшість з розглянутих вище застосунків не мають можливості працювати на території України через брак інформації щодо українських продуктів, а також через відсутність розпізнавання складу, записаного українською мовою. Враховуючи це, можна стверджувати, що розвиток цього напрямку є досить перспективним.

1.2 Аналіз літературних джерел щодо існуючих підходів розпізнавання кулінарних страв

При розпізнаванні зображень використовуються декілька суттєвих підходів, за допомогою яких відбувається класифікація на ту чи іншу категорію. Найбільш розповсюдженим та передовим методом, який використовується більшістю розробників та дослідників, є згорткові нейронні мережі CNN. Група дослідників з інституту Алабами використали Faster R-CNN для виявлення об’єктів їжі та напоїв на зображеннях, зроблених AIM. Faster R-CNN – це двоетапна виявляюча структура, що використовується для одночасного створення обмежувальних рамок і міток

класів. Налаштовано Faster R-CNN з використанням попередньо підготовлених ваг ResNet-50 [3].

Студенти з інституту комп'ютерних наук застосували для класифікації фруктів VGG-19 [4]. VGG-19 є відомою мережею своєю глибокою архітектурою і має 19 вагових шарів, у тому числі згорткові та повнозв'язані шари. Підхід надає досить непогані результати класифікації. У своїй роботі дослідники зробили порівняльний аналіз мереж VGG-19 та VGG-16, більш точною мережею виявилась VGG-19.

Окрім згорткових нейронних мереж, методом класифікації, який використовується в обраній предметній області, є глибока нейронна мережа під назвою Vision Transformer. Вона вважається інноваційним інструментом для класифікації. Такий підхід був застосований науковцями з Індії. При вирішенні проблеми розпізнавання різних кухонь світу застосовано Vision Transformer [5]. Задіяно ручні функції: GIST, HoG (гістограма орієнтованих градієнтів) і LBP (локальний бінарний шаблон). Підхід досяг точності 94,63%.

Одним з широко використовуваних класичних методів класифікації є метод опорних векторів SVM. Однак, такий метод наразі використовують як додатковий рівень класифікації [6, 7] до класичних CNN.

Окрім класичних моделей CNN та ViT, є алгоритм «You Look Only Once» (YOLO). Принципом роботи мережі є розподілення зображення на сегменти певного розміру та поетапна їх обробка. Даний метод застосовано корейськими науковцями-розробниками для розпізнавання продуктів [8]. Розмір сітки, розділеної на зображення, становив 13×13, а сама мережа мала всього 3 шари. Слід зазначити, що класифікація продуктів відбувається не тільки графічно. Корейські дослідники при розробці системи захвату та класифікатору зображень рибних котлет зробили порівняльний аналіз різних версій алгоритму YOLO [9]. У результаті досліджень, YOLOv5 виявився у 6,25 разів швидшим за швидкість обробки YOLOv7, через що в роботі було

використано саме YOLOv5. Однак, у документації до YOLOv7 зазначається, що ця версія у 1,5 рази швидша за попередню.

У деяких роботах показано поєднання архітектур кількох більш примітивних нейронних мереж в одну комплексну. Деякі з дослідників [10–12] вважають, що такі гібридні моделі є більш ефективними за використання однієї конкретної.

Студенти з Китаю [10] для розроблення комплексної системи ідентифікації ступеня придатності їжі використали п’ять класичних моделей CNN (Densenet-121, ResNet-50, Inception-V3, VGG-16 і VGG-19) для ідентифікації дефектної моркви та виявили, що моделі CNN можуть бути ефективними для класифікації такого роду. Зрештою, ними було застосовано вдосконалену мережеву модель щільної капсули, що поєднує декілька запропонованих моделей.

Науковці з Канади пропонують для класифікації овочів використовувати дворівневу класифікацію, що також містить у собі підхід YOLO [4]. Класифікатором першого рівня в їхній роботі є Support Vector Machine (SVM), а класифікатором другого рівня – YOLOv3. Вектор ознак, що містить витягнуту інформацію про колір і текстуру, є входом для класифікатора першого рівня, а вихід класифікатора першого рівня з’єднується з класифікатором другого рівня. Загальна точність такого підходу становить 96,4%.

Аналіз літературних джерел щодо існуючих підходів розпізнавання кулінарних страв показав, що найбільш поширеним зараз є використання згорткових нейронних мереж, однак існують й інші підходи для успішної класифікації.

Також можна побачити, що ефективніше використовувати поєднання декількох нейронних мереж, ніж якусь конкретну. Такі роботи показують ліпші результати.

Якщо розглядати більш конкретно, зазвичай в роботах за основу беруть такі моделі, як YOLO та Faster R-CNN, тож доцільніше використовувати в даній роботі саме одну з них.

1.3 Постановка задачі

Актуальність даної роботи полягає у тому, що у світі досить велика кількість людей мають харчову алергію на ту чи іншу категорію продуктів, а на фоні незбалансованих прийомів їжі все більше людей набувають захворювання, які пов’язані із дотриманням дієти.

Для того, щоб полегшити життя для таких людей, а також допомогти їм у розпізнаванні тих страв, які можуть бути потенційно небезпечними для них, необхідно розробити такий набір інструментів, що міг би задовольнити їх потреби.

Об’єктом роботи є процес розроблення мобільного застосунку розпізнавання кулінарних страв.

Метою роботи є розроблення за допомогою згорткових нейронних мереж мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров’я.

Спираючись на описане вище пропонується такі задачі для вирішення:

– дослідження аналогів застосунків що спеціалізуються на класифікації страв;

– розглядання літературні джерела з метою виявлення сучасних підходів до вирішення даної проблеми;

– виявлення механізмів оброблення зображень в мобільних застосунках;

– визначення принципів розпізнавання кулінарних страв у мобільних застосунках;

– виявлення особливості предметних областей даної роботи;

– розроблення методики будування мобільного застосунку для розпізнавання кулінарних страв для людей з вадами здоров’я;

– вибір інструментальних засобів для створення застосунку;

– детальне висвітлення етапів розроблення мобільного застосунку;

– тестування розробленого застосунку та аналіз результатів;

– виявлення перспектив для подальшої роботи;

– сформування висновків щодо виконаної роботи.

2 ОСОБЛИВОСТІ РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ РОЗПІЗНАВАННЯ КУЛІНАРНИХ СТРАВ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗДОРОВ'Я

2.1 Механізми оброблення зображень в мобільних застосунках

Важливу роль при класифікації зображень відіграє їх попереднє оброблення. Накладання фільтрів та масок, зміщення, зміна розміру, регресія – усе це є суттєвим кроком у реалізації. Механізми оброблення зображень в мобільних застосунках несуттєво залежать від платформи, однак для кожного з методів класифікації існує свій оптимальний шлях оброблення та його сегментацію [11].

Серед комплексних згорткових нейронних мереж, що мають широке використання в розробленні мобільних застосунків, слід звернути увагу на принцип роботи R-CNN та похідних від неї, а саме було взято Faster R-CNN, в якості прикладу для дослідження обробки зображень та подальшої їх класифікації.

Перш за все слід визначити основні складові архітектури мережі R-CNN, щоб мати представлення її принципу роботи:

– backbone (основа) – попередньо натренована побічна згорткова нейронна мережа, яка використовується для витягнення ознак із вхідного зображення;

– функція вибору регіональних пропозицій – складова, що використовується для генерації можливих регіонів на зображенні, які можуть містити об'єкти;

– класифікація ознак – ознаки, витягнуті з кожної регіональної пропозиції, подаються на вхід класифікатору (наприклад, SVM), який визначає, до якого класу належить об'єкт у даній пропозиції;

– регресія Bounding Boxes – паралельно з класифікацією, для кожної пропозиції виконується регресія, щоб уточнити координати bounding box, де знаходиться об’єкт.

Відмінності ж двох версій будуть висвітлені нижче, оскільки їх порівняльний аналіз є досить цікавим та корисним для розуміння.

У класичній версії роботи мережі R-CNN в якості функції вибору регіональних пропозицій для попереднього оброблення вхідного зображення виступає селективний пошук selective search. Він вважається першим шаром методу, в результаті роботи якого вхідне зображення поділяється на декілька сегментів за різними ознаками та передає вектор з ними до другого шару, в якому знаходиться класифікатор.

Селективний пошук має досить простий та зрозумілий алгоритм роботи: вхідне зображення розбивається на велику кількість областей (приблизно 2000) за подібністю. З кожною ітерацією деякі з областей зливатимуться до тих пір, поки області не утворять області для подальшої класифікації.

Даний метод передбачає розрахунок вектора для двох потенційних для злиття ділянок. Основні чотири ознаки, за якими ведеться розрахунок, є подібність кольорів, текстури, розміру та перетин [12].

Першим з розрахованих параметрів для порівняння ділянок є подібність кольорів. Спеціально для кожної області генерується гістограма кожного каналу кольорів, присутніх на зображенні. У гістограмі кожен з колірних каналів може містити від 0 до 25 бінів (змінних, що набувають значень 0 чи 1). Загальною кількістю набраних бінів є всього 75 бінів (по 25 для кожного R, G і B), і всі канали об’єднуються у вектор (n = 75) для кожної області. Потім знаходиться подібність областей за кольором за формулою, приведеною нижче

$$S_{color}(r_i; r_j) = \sum_{k=1}^{n} min(c_i^k; c_j^k), \quad (2.1)$$

де r_i, r_j – області, що розглядаються;

c_i^k, c_j^k – біни різних областей, що розглядаються.

Подібність текстури обчислюється за допомогою згенерованих 8 похідних Гауса від зображення та використовує гістограму з 10 відрізками для кожного колірного каналу. В результаті отримується 240-вимірний вектор (тобто, n = 240) для кожної області. Формула для обчислення приведена нижче

$$S_{texture}(r_i; r_j) = \sum_{k=1}^{n} min(t_i^k; t_j^k), \quad (2.2)$$

де c_i^k, c_j^k – значення векторів, що розглядаються.

Подібність розміру: основна ідея подібності розміру полягає в тому, щоб менші області легко об'єднувалися. Якщо цю подібність не взяти до уваги, більший регіон продовжуватиме зливатися з більшим регіоном, і пропозиції щодо регіонів у різних масштабах будуть створюватися лише в цьому місці.

$$S_{size}(r_i; r_j) = 1 - \frac{\left(size(r_i)+size(r_j)\right)}{size(img)}, \quad (2.3)$$

де $size(r_i)$, $size(r_j)$ – розміри областей;

$size(img)$ – розмір повного зображення.

Ще одною зі змінних, що розглядається в областей, що розглядаються, є перевірка на перетин. Обчислення даного параметру вимірює, наскільки дві області підходять одна одній. Якщо дві області добре вписуються одна в одну (наприклад, одна область присутня в іншій або мають спільну сторону), їх слід об'єднати, якщо дві області навіть не торкаються одна одної, їх не слід об'єднувати.

$$S_{fill}(r_i; r_j) = 1 - \frac{\left(size(BB_{ij})-size(r_i)-size(r_j)\right)}{size(img)}, \quad (2.4)$$

де $size(BB_{ij})$ – розмір обмежувальної рамки (bounding box) навколо областей i та j.

За допомогою обчислених вище величин розраховується загальна подібність, визначена формулою нижче

$$S(r_i, r_j) = a_1 * S_{color}(r_i, r_j) + a_2 * S_{texture}(r_i, r_j) + a_3 * S_{size}(r_i, r_j) + \\ + a_4 * S_{fill}(r_i, r_j), \quad (2.5)$$

де a_k – ваги функції, які можуть приймати значення між 0 та 1.

Отримані в результаті обробки зображення ділянки передаються у виді мапи до другого шару мережі R-CNN.

У Faster R-CNN замість селективного пошуку застосовується Region Proposal Network (RPN). RPN – це легка нейронна мережа, яка приймає на вхід карти ознак із функції backbone і генерує регіональні пропозиції bounding boxes. Вона є більш швидкою порівняно з її аналогом у класичній R-CNN.

Вона має 3 шари: спільний згортковий, класифікаційний та регресійний. Перший перетворює вхідні дані в їхню більш стиснену форму задля ефективності оброблення іншими шарами. Другий прогнозує ймовірність того, що кожна з ділянок зображення містить об’єкт. Останній шар прогнозує координати bounding box для кожної з ділянок. Він генерує чотири значення для кожної з них, які використовуються для коригування позиції і розміру.

Також слід приділити увагу архітектурі шару, що мітить у собі backbone, тобто основу. Її зазвичай представляють такі побічні згорткові нейронні мережі, як ResNet або VGG. Далі буде розглянуто в якості цієї функції модель VGG-16.

VGG-16 є досить цікавою згортковою нейронною мережею для обробки зображень, оскільки основний метод її роботи був запозичений з механізму зору ссавців, а більше людського зору. Алгоритм у своїй основі

представляє поетапне зменшення зображення та здійснення за допомогою згорток дослідження ознак, що залишилися після.

Вхідне зображення повинно мати розмір 224×224 пікселі, причому якщо вхідні дані є більшими за стандарт, відбувається трансформування з приведенням до потрібного формату чи відсічення країв, що є зайвими.

З кожним етапом об'єкт сегментування буде зменшувати свій розмір вдвічі аж до 1×1. Між зменшеннями вхідні дані два рази піддаються згортці 3×3 пікселі.

Під час згортки відбувається фільтрація: певна кількість фільтрів, упорядкована етапом зменшення вхідної фотографії, проходять по всьому зображенню в пошуках специфічних ознак, таких як границі, текстура та колір об'єкта. На першому етапі їх величина становить 64, з кожним перетворенням вона зростає вдвічі. При знаходженні хоча б однієї з видів ознак застосовується функція активації ReLU.

ReLU (Rectified Linear Unit) – це широко використовувана функція активації у глибокому навчанні, яка додає нелінійності до результатів нейронної мережі. Принципом її роботи є перетворення вхідних значень на невід'ємні. Рівнянням це можна описати так

$$ReLU(x) = \max(0, x), \tag{2.6}$$

де x – це вхідне значення.

Така функція має низку переваг, серед яких є суттєва швидкість обчислення та легкість реалізації в порівнянні з іншими класичними функціями активації, такими як, наприклад, сигмоїда, проте має деякі несуттєві недоліки.

Після фінального етапу згорток до отриманих результатів застосовується функція SoftMax. За допомогою експоненціальної функції отриманий раніше вектор результатів нормалізується, а значення переводить у числа в діапазоні від 0 до 1.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=0}^{n} e^{z_j}}, \quad (2.7)$$

де i – номер елемента z, що перетворюється;

z_i – значення елемента під номером i;

n – загальна кількість елементів у векторі z.

Слід також зазначити, що після перетворення кожного з елементів їх загальна сума буде дорівнювати 1.

Архітектура та принцип роботи класифікатору для двох розглянутих згорткових нейронних мереж є подібним: після попереднього оброблення зображення утворена мапа ознак потрапляє до цього шару. Тут відбувається класифікація зображення.

Faster R-CNN також має у тому шаром, де знаходиться класифікаторна функція, додатково також наявний RoI (Region of Interest), який використовується для обробки виокремлених регіонів інтересу. Ці регіони виникають під час процесу обробки зображення попередніми шарами алгоритму. Кожен такий регіон може містити об’єкти різних розмірів та форм.

У цьому шарі кожен регіон інтересу піддається ресайзу або регресії для стандартизації їх розмірів перед подальшою обробкою мережею. Це важливо, оскільки нейронна мережа має фіксований розмір вхідних даних. Таким чином, кожен регіон інтересу, незалежно від його вихідного розміру, приводиться до однакового стандартного розміру перед тим, як буде подано на вхід класифікатору.

Після цього стандартизовані регіони інтересу проходять через класифікатор, де вони аналізуються для визначення наявності об’єктів та їх класифікації. Класифікатор може використовувати різні методи, включаючи згорткові нейронні мережі або інші моделі машинного навчання.

Для кожного з сегментів окремо розраховується функція втрат.

Отже, можна зробити висновки, що розглянуті моделі є досить ефективними та простими у розумінні.

Класична R-CNN показує досить непогані результати при класифікації з декількома об’єктами на зображенні, однак її швидкість внаслідок окремої класифікації кожного з сегментів, яких може бути досить велика кількість, може бути низькою. Вона не може бути використана в задачах розпізнавання в реальному часі.

І хоча Faster R-CNN є суттєво швидшою за рахунок іншого методу сегментації зображення, її швидкості для цього бракує. Тож доцільно використовувати їх методом надсилання фіксованого зображення до модуля з нейронною мережею.

2.2 Принципи розпізнавання кулінарних страв у мобільних застосунках

Виходячи з попереднього підрозділу, процес розпізнавання образів, а в даному випадку кулінарних страв, на зображенні є досить складним процесом, тож було розроблено основні принципи для будови модуля розпізнавання кулінарних страв у мобільний застосунок:

- збір даних та їх підготування;
- вибір моделі для реалізації;
- налаштування моделі;
- тренування моделі;
- тестування моделі;
- інтегрування до застосунку.

Подані етапи у розробленні програмної платформи з означеними вище функціями є досить значними та об’ємними, тож далі кожен з них буде розглядатися більш детально.

Збір даних, які слугуватимуть тренувальними прикладами, а також тестовою вибіркою для класифікування є досить фундаментальним етапом при використанні нейронних мереж. Важливо підібрати такі набори даних для навчання, які були б досить різноманітними: зображення

класифікаційних об’єктів повинні бути зроблені з різних кутів зору, при різних освітленнях, а також в різних контекстах та з різними фонами. Таким чином, можна позбутися перенавчання вибірки, а також набуття мережею хибних ознак.

Ще однією важливою ознакою наборів даних повинна бути влучність його тематики в контексті області опрацювання [13–15]. Оскільки в поданій роботі розглядається розпізнавання кулінарних страв, насамперед цікавими залишаються тільки ті пакети з даними, які мають у собі зображення їжі.

Одним з найбільш популярних наборів даних, що спеціалізуються на зображеннях кулінарних страв та їжі загалом є Food-101 (рис. 2.1). Він є широко використовуваним в області розпізнавання їжі при розробленні різноманітних застосунків, які того вимагають. Ця вибірка містить у собі 101 різний вид кулінарних страв, кожний з яких налічує по 1000 зображень. Саме звідти і походить назва цього набору.

Рисунок 2.1 – Різноманіття страв у вибірці Food-101

Поданий набір включає в собі різноманітні страви з різноманітних категорій їжі. У ньому наявні як фаст-фуд чи легка їжа типу піци, суші, бургерів тощо, так і повноцінні страви різних кухонь світу. У ньому можна знайти досить поширені комплексні страви на кшталт різноманітних салатів типу цезарю чи олів'є, а також м'ясних страв. Це означає, що цей набір є прийнятним для застосування у класифікаторі, націленому на роботу в українському просторі.

Набір даних iFood-2019 є ще одним значним ресурсом для досліджень у галузі комп'ютерного зору та машинного навчання, зосереджених на розпізнаванні кулінарних страв на зображеннях.

Він є більш новим за розглянений вище набір даних і містить у собі більшу кількість страв з різних куточків світу. Основною його перевагою є те, що у своєму складі iFood-2019 має усього 251 видів кулінарних страв, натомість Food-101 має лише 101.

Також слід зазначити що страви, які наявні в цьому наборі також мають розбиття за категорією їжі (десерти, салати, бургери, суші, основні страви тощо) та за регіональною складовою, що полегшує процес роботи з цим пакетом даних.

Вибір моделі також є важливою частиною при розробленні застосунку. При цьому слід звернути увагу на особливості, які притаманні проєкту: платформу, на якій подана програма працюватиме, вид процесу класифікації (у реальному часі чи за фотографією), а також на системні можливості пристрою.

Найбільш поширеними стосовно задачі класифікації зображень у мобільному застосунку є інструменти, які базуються на згорткових нейронних мережах (CNN). Вище вже були представлені види, що мають в основі роботу цього методу, оскільки він є одним з популярніших та одним з ефективніших на сьогоднішній день.

Налаштування моделі дозволяє підвищити ефективність основного тіла обраного метода для локальної проблеми, здійснити це можна різними методами:

– визначення ваг функцій (це допоможе визначити, які з параметрів є важливими у поставленій задачі, а які є несуттєвими);

– заміна чи вдосконалення функції втрат (більшість з них є ефективними тільки у певних ситуаціях);

– заміна чи прибирання деяких з класичних складових основного алгоритму (деякі складові можуть бути неефективними при вирішенні окремих проблем);

– добудова додаткових функцій (впровадження нових модулів може покращити класифікацію в конкретній задачі).

Деякі з поданих вище методів калібрування алгоритму розпізнавання образів не є обов’язковими, але можуть бути присутніми при розробленні специфічного проєкту.

При тренуванні моделі важливо зберігати баланс при використанні тренувальної вибірки. Вона повинна бути досить різноманітною, а також добре репрезентувати ті категорії, за якими йде навчання. Кожен з класів повинен мати приблизно однакову кількість представлених екземплярів.

Для підвищення різноманітності та узагальненості набору даних можна використовувати методи аугментації даних. Це включає в себе зміну зображень шляхом обертання, зміну масштабу, зсуву, яскравості [16–21]. Таким чином, можна швидко отримати велику кількість зображень, коли є нестача зразків.

Також слід обирати оптимальну кількість епох при навчанні, при цьому з інтервалом у деяку кількість епох слід проводити контрольну класифікацію з метою виявлення проблеми перенавчання.

Після тренування навчальної моделі необхідно перевірити її ефективність на незалежному тестовому наборі даних. На цьому етапі проводиться аналіз моделі, оцінюється результати класифікації методом

обчислення точності. За отриманими результатами приймається рішення: чи треба повторне або додаткове тренування мережі, чи ні.

У разі успішного навчання навчену модель треба інтегрувати в застосунок. Зробити це можна, наприклад, за допомогою архітектури API. API – це набір програмних інструкцій і інструментів, які забезпечують взаємодію між різними програмними компонентами.

Отже, було розглянуто основні принципи становлення моделі розпізнавання кулінарних страв. Можна побачити, що більша частина з них так чи інакше відносяться до побудови нейронної мережі, тож це можна вважати найважливішою частиною. На неї і треба робити акцент при розробленні. Також можна виділити відбір набору даних, як один із найважливіших принципів навчання моделі. Оскільки розглянутий вище набір даних iFood-2019 є більш новим за Food-101, а також він має більшу кількість видів кулінарних страв, доцільніше використовувати його при навчанні мережі.

2.3 Застосування методів системного аналізу для виявлення особливостей предметної області

Оскільки тематика даної роботи є досить об’ємною, важливо провести детальний її аналіз та окреслити обмежувальні границі задля можливості подальшого планування роботи над проєктом.

Перш за все, слід визначити ту націленість користувача, яка буде впливати на результат. Оскільки розроблювана програмна платформа є націленою на український ринок, слід враховувати особливості обраного простору щодо окреслення предметних областей. Таким чином, розроблювана модель буде працездатною для основної цільової аудиторії при певних обмеженнях.

Слід розглянути ту предметну область, яка є основною у даному застосунку. Такою є простір страв у світі. Їх налічується величезна кількість, а кухонь, на які вони розбиті, є сотні. Тож в даному випадку слід окреслити ту область, яка в подальшому буде опрацьовуватися в даній роботі.

Оскільки вище було визначено, що обмеження предметної області буде здійснюватися в основному за регіональною ознакою, є доцільним обрати суто такі страви, які є притаманними українській кухні. Таким чином, чітко окреслені рамки.

Продовжуючи тему кулінарних страв, важливо означити, що страви є комплексною структурою, тож при розробленні слід вважати класифіковану страву за перелік її інгредієнтів. Таким чином, складне буде розібрано на прості частини. Це буде в подальшому корисно для надання рекомендації користувачеві.

Також слід звернути увагу на предметну область, яка охоплює вади здоров’я, при яких треба дотримуватися дієтичного харчування, наприклад, харчова алергія.

Найбільш розповсюдженими продуктами-алергенами є такі:

- продукти, що містять глютен;
- ракоподібні;
- яйця;
- риба;
- горіхи;
- молочні продукти (лактоза);
- соя.

Оскільки в даній роботі розглядається лише алергія і не за окремими продуктами, а за категоріями, деякі з перелічених у рецепті інгредієнтів повинні також мати свою визначену категорію.

Також до вад здоров’я, які передбачають дієтотерапію чи утримання від вживання певної категорії їжі, є інші захворювання, наприклад, такі, що пов’язані з шлунковим трактом, а також із гормональним фоном, однак, ці

категорії не є єдиними, що передбачають спеціальний раціон. Нижче виявлено ті, що будуть надалі розглядатися при розробленні:

– шлункова виразка, (заборонені так продукти, як алкогольні напої, спеції, молочні продукти, солодощі);

– цукровий діабет (другого типу), (заборонені так продукти, як свинина, алкогольні напої, спеції, рис, солодкі фрукти);

– гастрит, (заборонені так продукти, як соя, спеції, алкогольні напої, глютенові продукти).

Слід зазначити, що дієтотерапія в такому випадку є більш складною, оскільки передбачає собою виключення не одного продукту чи категорії, а цілого переліку. Зазначене вище є ускладнювальним фактором, тож його треба також враховувати при плануванні розроблення.

Враховуючи зазначене вище, кожне захворювання із визначених повинне мати перелік категорій, що є забороненими (не рекомендованими) до вживання. Таким чином, можна буде легко класифікувати страву.

У результаті застосування методів системного аналізу, розглянуто та досліджено дві предметні області, що є ключовими для розроблюваного застосунку: кулінарні страви та вади здоров’я. У результаті виявлено певні особливості та окреслено для кожної з них кордони для подальшої реалізації.

2.4 Методика розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров’я

Спираючись на висвітлені вище результати, можна сформулювати певні етапи, які будуть основними в даній роботі.

Щодо розглянутих у підрозділі 2.1 методів обробки та подальшої класифікації вхідного зображення вирішено взяти гурткову нейронну мережу Faster R-CNN, методом сегментації якої буде підмережа VGG-16. Вона є

досить простою у реалізації та досить точною при використанні, тому доцільно використовувати саме її.

Щодо наборів даних вирішено взяти iFood-2019 із-за великої кількості налічуваних екземплярів, а також власноруч зроблені зображення [22–26].

У результаті цього отримана більша вибірка, яка буде ефективніше задовольняти потреби роботи. Також до утвореного набору даних будуть враховані регіональні обмеження даної предметної області.

Пропонованими кроками щодо розроблення мобільного застосунку для розпізнавання кулінарних страв для людей з вадами здоров’я є такими:

Крок 1. Підготовка набору даних, додавання своїх екземплярів, розбиття вибірки на тренувальну та тестову.

Крок 2. Побудова моделі VGG-16.

Крок 3. Побудова моделі Faster R-CNN, її калібрування.

Крок 4. Навчання моделі.

Крок 5. Розроблення інтерфейсу.

Крок 6. Імплементування модуля з нейронною мережею.

3 РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ РОЗПІЗНАВАННЯ КУЛІНАРНИХ СТРАВ ДЛЯ ЛЮДЕЙ З ВАДАМИ ЗДОРОВ'Я

3.1 Вибір інструментальних засобів для створення застосунку

3.1.1 Основні інструменти для створення застосунку

Задля ефективного вирішення поставлених задач, перш за все, слід обрати ті інструменти, які найбільш підходять до їх специфікації. Вибір правильних інструментів є критичним для досягнення оптимальних результатів, мінімізації витрат часу і ресурсів, а також забезпечення високої якості кінцевого продукту [27-29].

Найбільш поширеною мовою програмування для розроблення проєктів, що мають в собі елементи машинного навчання та розпізнавання образів вважають Python. Вона має велику кількість переваг, що роблять її такою популярною. Детальніше кожну з них висвітлено нижче.

Насамперед, подана мова програмування має високу швидкодію, що дозволяє здійснювати велику кількість обчислень без витрати великої кількості часу. Ця перевага є найбільш помітною в ситуації, коли є великий обсяг даних для оброблення.

Синтаксис є досить простим та інтуїтивно зрозумілим для тих, хто тільки починає здобувати навички програмування, тож вибір її в якості початкової мови спрощує процес навчання. Код таких застосунків не міститиме багато складних конструкцій, як це буває в об'єктно-орієнтованих мовах, таких як C# та Java.

Слід виокремити, що Python має велику кількість фреймворків для опрацювання у галузі комп'ютерного зору. Найбільш популярними з них є Tensorflow та PyTorch [30], вони мають широкий асортимент інструментів задля побудови та тренування моделі, її тестування, а також надають можливість зберегти натреновану модель для подальшого використання. Існує бібліотека Keras, що є добіркою найбільш популярних наборів даних.

Ще одною безсумнівною перевагою цієї мови програмування є кросплатформність. Python має інтерпретатор для більшості з популярних операційних систем, що дозволяє виконувати один і той же код на різних системах без необхідності вносити зміни в сам код. Це значно спрощує процес розробки та розгортання програмного забезпечення, знижуючи витрати на адаптацію програм до різних середовищ.

Для розроблення кросплатформних застосунків існує такий фреймворк, як Kivy [31]. Він дозволяє розробляти віконні форми, які можуть виконуватися на більшості операційних систем. Kivy є потужним інструментом для створення користувацьких інтерфейсів, забезпечуючи можливість створення застосунків для Windows, macOS, Linux, iOS та Android.

Враховуючи універсальність розглянутої вище мови програмування та різноманіття бібліотек, що вона містить, вирішено взяти Python у якості основної мови для розроблення.

3.1.2 Використання PyTorch для побудови моделі нейронної мережі

У якості інструментарію для побудови, тренування та тестування нейронної мережі обрано фреймворк мови програмування Python під назвою PyTorch. Він є потужним інструментом для розробки і дослідження нейронних мереж, який здатен задовольнити потреби як початківців, так і досвідчених дослідників в галузі машинного навчання та розпізнавання образів.

Бібліотека містить у собі великий асортимент методів для роботи з класичними моделями, а також надає можливість спроєктувати власну нейронну мережу. Завдяки цьому можна ефективно корегувати основу шаблону задля його вдосконалення та пристосування до специфікації поставленої задачі.

Лістинг 3.1 Конструювання моделі Faster R-CNN з декількох модулів:

```
model = FasterRCNN(
   backbone,
   num_classes=18,
   rpn_anchor_generator=rpn_anchor_generator,
   box_roi_pool=roi_pooler
)
```

У бібліотеці PyTorch також містяться досить зручні інструменти для тестування нейронних мереж, що дозволяють розрахувати втрати Loss для моделі та виявити відсоток правильності класифікатора на обробленій вибірці даних.

Слід зазначити, фреймворк, що розглядається, також має засоби для завантаження власного набору даних, його налаштування (перемішування, трансформування даних тощо), створення підвибірок. Застосування таких функцій допомагає оптимізувати тренування, а також дозволяє зробити набір даних більш різноманітним, зменшуючи ймовірність перенавчання моделі.

Лістинг 3.2 Приклад опрацювання:

```
train_csv_file = ‘ifood-2019/train_labels.csv’
train_image_folder = ‘ifood-2019/train_set/’
train_dataset = CustomDataset(csv_file=train_csv_file,
root_dir=train_image_folder, transform=data_transform)
train_dataloader = DataLoader(train_dataset, batch_size=2, shuffle=True)
```

Також PyTorch співпрацює з бібліотекою Keras, яка містить у собі найбільш поширені з наборів даних, що застосовуються для навчання нейронних мереж.

3.1.3 Вибір інструментів для клієнтської частини застосунку

Мова програмування Python є універсальною, оскільки має велику кількість фреймворків, спрямованих на їх застосування у конкретній області розроблення: машинне навчання, вебзастосунки, графічне програмування, розроблення ігор тощо.

Для розроблення кросплатформних застосунків існує потужний набір інструментів під назвою Kivy. За допомогою нього можна швидко зробити простий інтерфейс для застосунку, що буде працювати на більшості класичних операційних систем.

Структура написання коду під подану бібліотеку є подібною до CSS формату верстання. Головний простір можна розбити на контейнери, які далі наповнюються об’єктами, що відображатимуться на сторінці. Синтаксис є інтуїтивно зрозумілим, що допомагає швидко розібратися з побудовою форми.

Інструментарій надає можливість додавати такі інтерактивні елементи, як кнопка, поле для вводу, список тощо, а також змінювати їхні властивості в залежності від поставленої задачі.

Також цей фреймворк має декілька додаткових розширень, що допомагають конвертувати проєкт у єдиний файл. Таким чином, можна досить легко отримати готовий застосунок, що відкриватиметься без середовища розроблення.

3.2 Етапи розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров’я

Першим етапом при розробленні мобільного застосунку є завантаження та подальше оброблення набору даних. На основі проведеного вище огляду існуючих вибірок, що містять зображення кулінарних страв,

обрано iFood-2019, як найбільш оптимальну для подальшого застосування. Оскільки поданий набір даних вдосконалюється, вирішено взяти поточну його версію, яка налічує 250 класів, а тестовий набір сягає 120000 зображень.

Оскільки прийнято рішення щодо обмеження вибірки за регіональним принципом, із основної вибірки виділено 17 класів (рис. 3.1).

Рисунок 3.1 – Розподілення вибірки на класи

Також до деяких класів додатково додано декілька екземплярів задля врівноваження вибірки, а саме 32 до 15-го класу (омлет), 24 до 6-го класу (яблучний пиріг) та 42 до 11-го класу (вирізка зі свинини). Після оброблення кількість екземплярів сягає 9000 зображень. Також введено підвибірку обсягом 1000 зображень задля проведення прискореного тренування.

Набір даних представлено файлом, що має розширення .csv, та каталогом з зображеннями. Задля можливості користуватися ним, слід розробити похідний від основного класу Dataset, що є в бібліотеці PyTorch за замовчуванням. Він матиме шляхи до файлу та до каталогу з екземплярами та трансформуватиме їх для подальшої роботи з інструментами даної бібліотеки.

Лістинг 3.3 Ініціювання класу для завантаження вибірки:

```
class CustomDataset(Dataset):
    def __init__(self, csv_file, root_dir, transform=None):
        self.data_frame = pd.read_csv(csv_file)
        self.root_dir = root_dir
        self.transform = transform

    def __len__(self):
        return len(self.data_frame)

    def __getitem__(self, idx):
        img_name = os.path.join(self.root_dir, self.data_frame.iloc[idx, 0])
        image = Image.open(img_name)
        label = self.data_frame.iloc[idx, 1]
        if self.transform:
            image = self.transform(image)
        return image, label
```

Другим кроком для створення застосунку є побудова та тренування нейронної мережі. Раніше обрано Faster R-CNN з VGG-16 в якості функції backbone.

Оскільки заздалегідь натреновану на зображеннях кулінарних страв модель VGG-16 знайти не вдалося, прийнято рішення провести власне тренування. За допомогою засобів фреймворку PyTorch завантажено дану нейронну мережу.

Лістинг 3.4 Ініціалізація мережі та її класифікатора:

```
vgg16 = torchvision.models.vgg16()
vgg16.classifier = nn.Sequential(
```

```
        nn.Linear(25088, 4096),
        nn.ReLU(True),
        nn.Linear(4096, 4096),
        nn.ReLU(True),
        nn.Linear(4096, 18),
    )
    criterion = nn.CrossEntropyLoss()
    optimizer = optim.AdamW(vgg16.parameters(), lr=0.0001, weight_decay=0.01)
```

На рисунку 3.2 можна побачити повної структуру. Саме шар MaxPool2d відповідає за згортання екземпляру.

Оскільки значення kernel_size дорівнює 2, вхідне зображення згортатиметься вдвічі.

Усього модель містить в собі 13 повнозв’язаних шарів. Слід звернути увагу на те, що кількість вхідних та вихідних нейронів збільшується після згортки зображення вдвічі – до 512. Після кожного шару виконується функція активація ReLU.

Після конструювання нейронної мережі починається її тренування. Усього було виставлено 20 епох, але також додатково було додано систему раннього зупинення. Вона працює таким чином: результати нової епохи порівнюються з попередніми, якщо вони є вищими, то модель зберігається станом на дану епоху. Якщо ж ні: спеціальний лічильник рахує кількість результатів, що є нижчими за максимальний, якщо їх вже 3, тоді тренування завершується.

При досягненні ним фіксованої кількості результатів, що є нижчими за максимальний, функція зупиняє тренування. Таким чином, отримується найбільш успішний результат, запобігається проблема перенавчання та зменшуються витрати часу.

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
  (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (22): ReLU(inplace=True)
  (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (25): ReLU(inplace=True)
  (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (27): ReLU(inplace=True)
  (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (29): ReLU(inplace=True)
  (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)
```

Рисунок 3.2 – Архітектура моделі VGG-16

Лістинг 3.5 Система раннього зупинення:

```
if val_acc > best_val_acc:
    best_val_acc = val_acc
    early_stop_count = 0
    torch.save(vgg16.state_dict(), 'vgg16_ifood2019.pth')
else:
    early_stop_count += 1
```

Як можна побачити на рисунку 3.3, усього нейронна мережа пройшла вісім епох, однак оскільки останні три епохи показала нижчий результат класифікації на валідаційній вибірці за п’яту епоху, то спрацювала функція раннього зупинення.

Процес тренування на розширеній вибірці зайняв всього дві з половиною години на графічному процесорі. Попередні запуски на центральному процесорі спрацьовували значно пізніше.

За час тренування модель змогла досягнути 72% ефективності на валідаційній вибірці на п’ятій епосі. Такий результат є прийнятним для поставленої задачі, оскільки очікуваний діапазон для поставленої задачі сягав від 70% до 90%.

```
------------------- 1 --------------------
Train Loss: 1.9791, Train Accuracy: 0.3808
Val Loss: 1.4893, Val Accuracy: 0.5483
------------------- 2 --------------------
Train Loss: 1.4096, Train Accuracy: 0.5669
Val Loss: 1.1289, Val Accuracy: 0.6596
------------------- 3 --------------------
Train Loss: 1.0961, Train Accuracy: 0.6699
Val Loss: 1.1473, Val Accuracy: 0.6530
------------------- 4 --------------------
Train Loss: 0.9330, Train Accuracy: 0.7122
Val Loss: 1.0817, Val Accuracy: 0.6825
------------------- 5 --------------------
Train Loss: 0.7632, Train Accuracy: 0.7627
Val Loss: 0.9828, Val Accuracy: 0.7234
------------------- 6 --------------------
Train Loss: 0.6397, Train Accuracy: 0.7959
Val Loss: 0.9827, Val Accuracy: 0.7070
------------------- 7 --------------------
Train Loss: 0.5439, Train Accuracy: 0.8322
Val Loss: 0.9781, Val Accuracy: 0.7185
------------------- 8 --------------------
Train Loss: 0.4310, Train Accuracy: 0.8682
Val Loss: 1.1008, Val Accuracy: 0.6989
Early stopping triggered
```

Рисунок 3.3 – Результати навчання моделі VGG-16

Наступним етапом розроблення є побудова нейронної мережі Faster R-CNN (рис. 3.4). Вона складається з трьох частин, за ними можна поділити конструювання на підетапи. Першим з них є ініціалізація backbone функції, яка в даному випадку буде представлена підготовленою вище моделлю VGG-16, другим – побудова підмережі RPN та третім – функція RoI.

```
  (rpn): RegionProposalNetwork(
    (anchor_generator): AnchorGenerator()
    (head): RPNHead(
      (conv): Sequential(
        (0): Conv2dNormActivation(
          (0): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
          (1): ReLU(inplace=True)
        )
      )
      (cls_logits): Conv2d(512, 15, kernel_size=(1, 1), stride=(1, 1))
      (bbox_pred): Conv2d(512, 60, kernel_size=(1, 1), stride=(1, 1))
    )
  )
  (roi_heads): RoIHeads(
    (box_roi_pool): MultiScaleRoIAlign(featmap_names=['0'], output_size=(7, 7), sampling_ratio=2)
    (box_head): TwoMLPHead(
      (fc6): Linear(in_features=25088, out_features=1024, bias=True)
      (fc7): Linear(in_features=1024, out_features=1024, bias=True)
    )
    (box_predictor): FastRCNNPredictor(
      (cls_score): Linear(in_features=1024, out_features=18, bias=True)
      (bbox_pred): Linear(in_features=1024, out_features=72, bias=True)
    )
  )
)
```

Рисунок 3.4 – Архітектура 2-го та 3-го шарів Faster R-CNN

Лістинг 3.6 Завантаження моделі Faster R-CNN та її складових:

vgg16_model = vgg16(pretrained=False)

pretrained_dict = torch.load('vgg16_ifood2019.pth', map_location=device)

model_dict = vgg16_model.state_dict()

pretrained_dict = {k: v for k, v in pretrained_dict.items() if k in model_dict and v.size() == model_dict[k].size()}

model_dict.update(pretrained_dict)

vgg16_model.load_state_dict(model_dict).

```
rpn_anchor_generator = AnchorGenerator(
    sizes=((32, 64, 128, 256),),
    aspect_ratios=((0.5, 1.0, 2.0),) * len((32, 64, 128, 256))
)

roi_pooler = torchvision.ops.MultiScaleRoIAlign(
    featmap_names=['0'],
    output_size=7,
    sampling_ratio=2
)

model = FasterRCNN(
    backbone,
    num_classes=18,
    rpn_anchor_generator=rpn_anchor_generator,
    box_roi_pool=roi_pooler
)
```

Оскільки для тренування дана мережа потребує вибірку, в якій чітко розділені регіони, де містяться об'єкти, а сформована вибірка не задовольняє цій умові, то прийнято рішення взяти центральну частину зображення, як єдину ділянку для кожного із зображень.

Як можна побачити, натренована нейронна мережа змогла дійти лише до результату 30% (рис. 3.5). Такий результат є неефективним, оскільки за порогове значення було взято 70%.

Причинами такої ситуації можуть бути недосконала форма регіонів, а також брак можливості збільшення batchiv та додавання додаткових функцій для розбавлення вибірки.

```
---------------- 1 ----------------
Train Loss: 2.8425 Acc: 0.0914
Val Loss: 2.5948, Val Accuracy: 0.0703
---------------- 2 ----------------
Train Loss: 2.3987 Acc: 0.1765
Val Loss: 2.4248, Val Accuracy: 0.1457
---------------- 3 ----------------
Train Loss: 2.1163 Acc: 0.2479
Val Loss: 2.3794, Val Accuracy: 0.2185
---------------- 4 ----------------
Train Loss: 1.9251 Acc: 0.2838
Val Loss: 2.3502, Val Accuracy: 0.2264
---------------- 5 ----------------
Train Loss: 1.8187 Acc: 0.2924
Val Loss: 2.3145, Val Accuracy: 0.2145
---------------- 6 ----------------
Train Loss: 1.7115 Acc: 0.3267
Val Loss: 2.2498, Val Accuracy: 0.2461
---------------- 7 ----------------
Train Loss: 1.6841 Acc: 0.3318
Val Loss: 2.0057, Val Accuracy: 0.2839
---------------- 8 ----------------
Train Loss: 1.6586 Acc: 0.3597
Val Loss: 1.9456, Val Accuracy: 0.3145
---------------- 9 ----------------
Train Loss: 1.6587 Acc: 0.3623
Val Loss: 1.9696, Val Accuracy: 0.2926
---------------- 10 ----------------
Train Loss: 1.6031 Acc: 0.3878
Val Loss: 1.9462, Val Accuracy: 0.3105
---------------- 11 ----------------
Train Loss: 1.5831 Acc: 0.4147
Val Loss: 1.9262, Val Accuracy: 0.3167
---------------- 12 ----------------
Train Loss: 1.5594 Acc: 0.4501
Val Loss: 1.9194, Val Accuracy: 0.3079
---------------- 13 ----------------
Train Loss: 1.5494 Acc: 0.4723
Val Loss: 1.9202, Val Accuracy: 0.2965
---------------- 14 ----------------
Train Loss: 1.5794 Acc: 0.4967
Val Loss: 1.9172, Val Accuracy: 0.3011
---------------- 15 ----------------
Train Loss: 1.5794 Acc: 0.4967
Val Loss: 1.9224, Val Accuracy: 0.2998
Early stopping triggered
```

Рисунок 3.5 – Результат навчання моделі Faster R-CNN

На основі цього пропонується взяти її підмережу в якості тимчасового варіанту класифікатора, при цьому модель Faster R-CNN залишається як перспектива для подальшої роботи та вдосконалення класифікатора застосунку.

Лістинг 3.7 Конструювання головного екрану за допомогою Kivy:

```
MDScreen:
    name: "main_screen"

    MDBoxLayout:
        orientation: ‘vertical’
        spacing: dp(18)
        padding: dp(0)
```

MDTopAppBar:

title: "AllergyDetected"

#elevation: 10

left_action_items: [[‘arrow-left’, lambda x: app.go_back()]]

right_action_items: []

md_bg_color: app.theme_cls.primary_color

pos_hint: {"top": 30}

size_hint_x: 1

Таким чином, реалізовано дві екранні форми (рис. 3.6): перша є основним екраном для завантаження та оброблення зображення, а друга слугує для визначення того, які продукти повинен уникати користувач в раціоні.

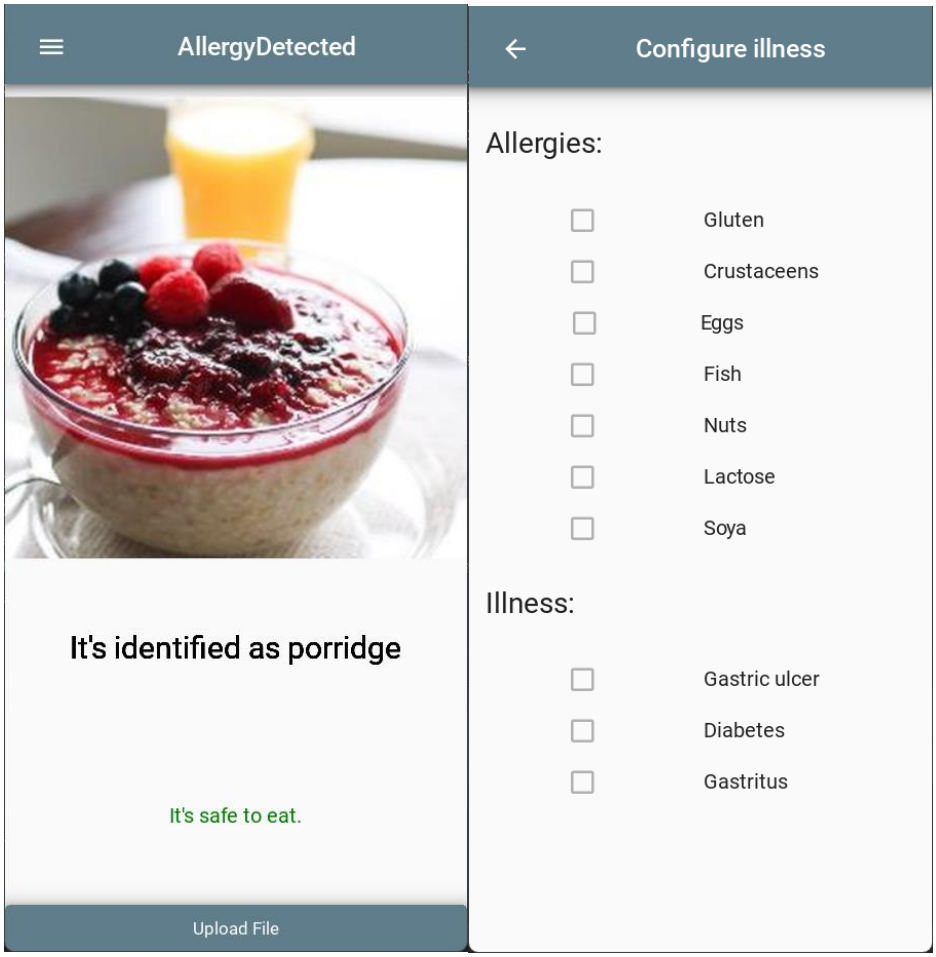


Рисунок 3.6 – Екранні форми застосунку

Кожен із класів зберігає свій стан у файлі classes.csv (рис. 3.7), у якому прописано, які з категорій він може потенційно в собі містити. Для кожної із страв позначають відсутність чи вміст одного із алергенів. Таким чином, наприклад, у класі, що позначає варене яйце, буде стояти одиниця у тому полі, де відбувається індикація наявності алергенів.

```
1,dumpling,1,0,1,0,0,0,0,1,1,1
2,knish,1,0,1,0,0,0,0,0,0,1,1,1
3,porridge,1,0,0,0,0,0,0,1,0,1
4,stuffed cabbage,0,0,0,0,0,0,0,0,1,1
5,chicken kiev,0,0,0,0,0,0,0,0,1,1
6,apple pie,1,0,1,0,0,0,0,0,1,1
7,sauerkraut,0,0,0,0,0,0,0,0,0,0
8,beet salad,0,0,0,0,0,0,0,0,0,0
9,pancake,1,0,1,0,0,0,0,1,1,1
10,compote,0,0,0,0,0,0,0,1,1,0
11,pork chop,0,0,0,0,0,0,0,1,1,1
12,boiled egg,0,0,1,0,0,0,0,0,0,0
13,brisket,0,0,0,0,0,0,0,0,0,0
14,fried egg,0,0,1,0,0,0,0,1,0,1
15,omelette,0,0,1,0,0,1,0,1,0,0
16,pirogi,1,0,1,0,0,0,0,1,1,1
17,rissole,0,0,1,0,0,0,0,1,0,1
```

Рисунок 3.7 – Зберігання метаданих класів у classes.csv

Тоді, при класифікації, якщо до цього було обрано конкретні алергії чи захворювання, під зображенням з’являться повідомлення «It’s might have: *перелік інгредієнтів-алергенів*» та «It’s not allowed (перелік захворювань, що забороняють вживання цієї страви)» (рис. 3.8), що попередять користувача щодо вмісту страви та доцільності її вживання.

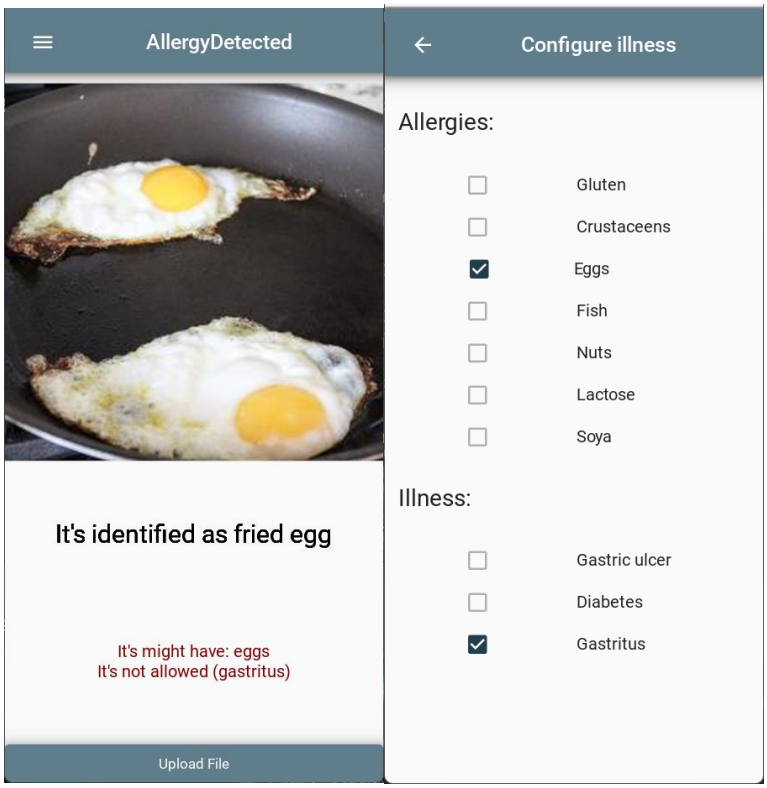


Рисунок 3.8 – Повідомлення-застереження
щодо вживання класифікованої страви

Отже, пройдено кожен етап, що зазначено вище, і застосунок можна вважати готовим.

3.3 Тестування розробленого мобільного застосунку та аналіз результатів

Перш за все, слід звернути увагу на те, що оскільки в застосунок імплементовано мережу VGG-16, то далі буде проводитися аналіз саме її результатів.

На рисунках 3.9 та 3.10 можна побачити графіки тенденції зниження втрат та збільшення відсотку точності класифікації для тренувальної та тестової вибірок. На них чітко видно рівномірність спадання втрат та підвищення точності на тренувальній вибірці, однак, валідаційна має високий підйом на початку та зниження росту/спадання у подальших епохах.

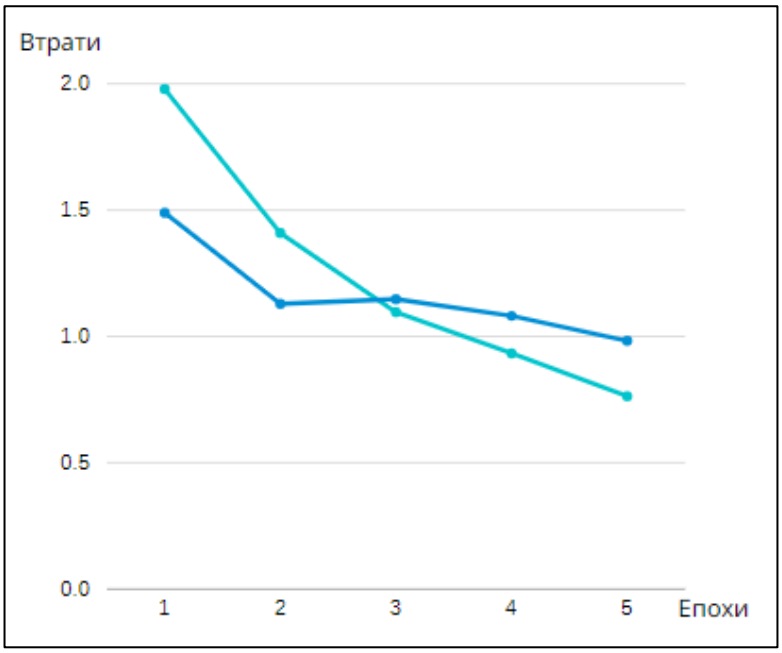


Рисунок 3.9 – Графік зниження втрат на тренувальній (зелена) та валідаційній (синя) вибірках

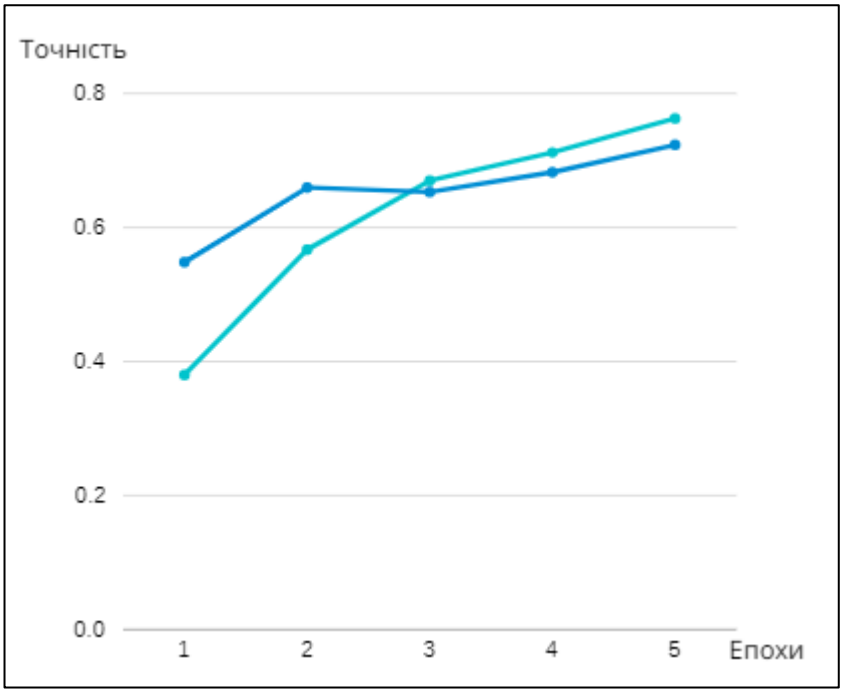


Рисунок 3.10 – Графік втрат на тренувальній (зелена) та валідаційній (синя) вибірках

Не зважаючи на те, що зображення у більшості випадків класифікуються правильно (рис. 3.11), точність мережі складає приблизно 72%, а деякі з класифікацій бувають помилковими (рис. 3.12).

Рисунок 3.11 – Приклад правильної класифікації страви

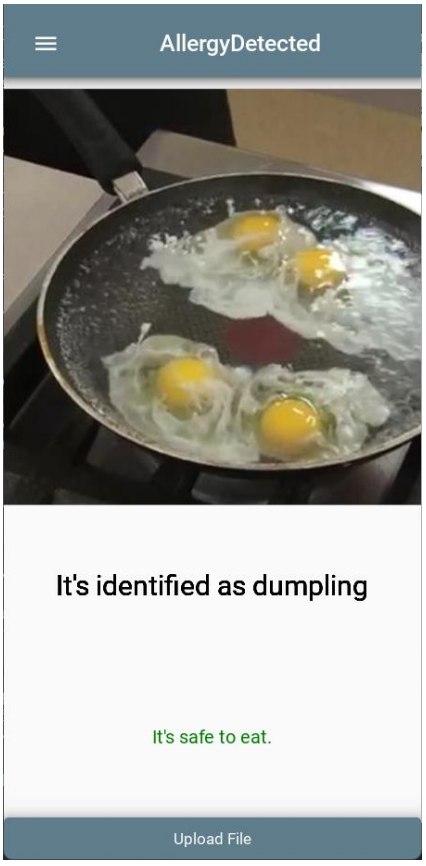


Рисунок 3.12 – Приклад неправильної класифікації страви

3.4 Перспективи подальшої роботи

Не зважаючи на те, що певна робота вже була проведена, існують пропозиції щодо перспектив до подальшого вдосконалення:

– збільшення класів у тестовий вибірці;

– доопрацювання моделі Faster R-CNN та подальша її імплементація в застосунок;

– впровадження можливості класифікації декількох страв на одному зображенні;

– збільшення кількості вибору алергій, та подальше їх розгалуження на підгрупи;

– збільшення кількості вибору захворювань, що передбачають дієтотерапію;

– вдосконалення інтерфейсу застосунку.

ВИСНОВКИ

У результаті виконання даної кваліфікаційної роботи розроблено мобільний застосунок для розпізнавання кулінарних страв.

У ході роботи було вирішено такі поставлені задачі:

– досліджено аналоги застосунків, що спеціалізуються на класифікації страв;

– розглянуто літературні джерела з метою виявлення сучасних підходів до вирішення даної проблеми;

– виявлено механізми оброблення зображень в мобільних застосунках;

– визначено принципи розпізнавання кулінарних страв у мобільних застосунках;

– виявлено особливості предметних областей даної роботи;

– визначено методику розроблення мобільного застосунку розпізнавання кулінарних страв для людей з вадами здоров’я;

– обрано інструментальні засоби для створення застосунку;

– детально висвітлено етапи розроблення мобільного застосунку;

– протестовано розроблений застосунок та проаналізовано отримані результати;

– виявлено низку перспектив для подальшої роботи;

– зроблено висновки щодо виконаної роботи.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [32].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. На харчову алергію страждає 1–3% дорослих та 4–6% дітей у світі | Центр громадського здоров'я. URL: https://phc.org.ua/news/na-kharchovu-alergiyu-strazhdae-1-3-doroslikh-ta-4-6-ditey-u-sviti (дата звернення 27.04.2024).

2. Microsoft Cloud for Healthcare: Reshaping the future of healthcare - Microsoft Industry Blogs. URL: https://www.microsoft.com/en-us/industry/blog/healthcare/2022/03/15/microsoft-cloud-for-healthcare-reshaping-the-future-of-healthcare/ (дата звернення 27.04.2024).

3. Ghosh, T., Han, Y., Raju, V., Hossain, D., McCrory, M. A., Higgins, J., ... & Sazonov, E. (2024). Integrated image and sensor-based food intake detection in free-living. Scientific Reports, 14(1), 1665.

4. SAMUNDI, R., GUNANIDHI, G., & KARTHIKEYAN, R. (2024). Automated Food Prediction Using Deep Learning with Calorie Estimation Algorithm.

5. Nijhawan, R., Sinha, G., Batra, A., Kumar, M., & Sharma, H. (2024). VTnet+ Handcrafted based approach for food cuisines classification. Multimedia Tools and Applications, 83(4), 10695-10715.

6. Zhu, L., & Spachos, P. (2021). Support vector machine and YOLO for a mobile food grading system. Internet of Things, 13, 100359.

7. Yasar, A. (2024). Analysis of selected deep features with CNN-SVM-based for bread wheat seed classification. European Food Research and Technology, 1-11.

8. Joshua, S. R., Shin, S., Lee, J. H., & Kim, S. K. (2023). Health to eat: a smart plate with food recognition, classification, and weight measurement for type-2 diabetic mellitus patients' nutrition control. Sensors, 23(3), 1656.

9. Kim, S., Baek, J., Jeong, M., Suh, J., & Lee, J. (2024). Development of Fishcake Gripping and Classification Automation Process Based on Suction Shape Transformation Gripper. Inventions, 9(1), 17.

10. Ghosh, T., Han, Y., Raju, V., Hossain, D., McCrory, M. A., Higgins, J., ... & Sazonov, E. (2024). Integrated image and sensor-based food intake detection in free-living. Scientific Reports, 14(1), 1665.

11. Xiao, Z., Wang, J., Han, L., Guo, S., & Cui, Q. (2022). Application of machine vision system in food detection. Frontiers in Nutrition, 9, 888245.

12. Beltran, Joshua & Ibarlin, Dominee Kyle & Mapa, Mark & Arboleda, Edwin. (2024). Exploring computer vision, machine learning, and robotics applications in banana grading: A review. International Journal of Science and Research Archive. 11. 1159-1166. 10.30574/ijsra.2024.11.1.0180.

13. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

14. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

15. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

16. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

17. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ*.

18. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

19. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

20. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

21. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

22. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

23. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

24. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.

25. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.

26. Gorokhovatskyi, V.O., Tvoroshenko, I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

27. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

28. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., and Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.

29. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.

30. PyTorch documentation – PyTorch 2.3 documentation. URL: https://pytorch.org/docs/stable/index.html (дата звернення 27.05.2024).

31. Welcome to Kivy – Kivy 2.3.0 documentation. URL: https://kivy.org/doc/stable/ (дата звернення 27.05.2024).

32. Подшивалова О.Є. (2024) Аналіз існуючих застосунків для розпізнавання кулінарних страв. Радіоелектроніка та молодь у XXI столітті: тези доповідей 28-го Міжнародного молодіжного форуму (Харків, 16–18 квітня 2024 р.). Харків: ХНУРЕ. Т. 7. С. 104-106.