Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Комп’ютерних наук\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва)

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Програмної інженерії\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва)

**АТЕСТАЦІЙНА РОБОТА**

**Пояснювальна записка**

\_\_\_\_\_\_\_\_\_\_\_\_\_другий (магістерський)\_\_\_\_\_\_\_\_\_\_\_\_\_

(рівень вищої освіти)

Дослідження підходів та можливостей використання технологій машинного зору \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_у дієтології \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(тема)

Виконав: студент 2 курсу, групи ПЗСм-16-1

спеціальності 121 - Інженерія програмного забезпечення\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код і повна назва спеціальності)

спеціалізації Програмне забезпечення систем \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва спеціалізації)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мезенцев П.А. \_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ініціали)

Керівник \_\_\_\_\_\_к.т.н. проф. Білоус Н.В.\_\_\_\_\_\_\_

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_Дудар З.В.\_\_\_\_\_\_\_

(підпис) (прізвище, ініціали)

2018 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Комп’ютерних наук\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Програмної інженерії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рівень вищої освіти\_\_\_\_\_\_\_\_\_\_\_другий (магістерський) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність \_\_\_\_\_\_121 – Інженерія програмного забезпечення\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код і повна назва)

Спеціалізація \_\_\_\_\_\_\_\_\_\_\_\_\_ Програмне забезпечення систем \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_ р.

**ЗАВДАННЯ**

НА АТЕСТАЦІЙНУ РОБОТУ

студентові\_\_\_\_\_\_\_\_\_\_\_\_\_*Мезенцеву Павлу Андрійовичу*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім’я, по батькові)

1. Тема роботи (проекту*)\_ Дослідження підходів та можливостей використання технологій машинного зору у дієтології*

затверджена наказом по університету від \_16\_ \_квітня\_ 2018 р. № 450 Ст

2. Термін подання студентом роботи до екзаменаційної комісії \_12\_ \_червня\_ 2018 р.

3. Вихідні дані до роботи *Дослідити методи та можливості технологій машинного зору, які вже використовуються в системах, пов’язаних з дієтологією. Визначити основні недоліки методів та особливостей їх використання. Знайти методи покращення існуючих систем, пов’язаних з дієтологією. Використовувати будь-які методи та засоби програмної інженерії.*

4. 3міст пояснювальної записки (перелік питань, що потрібно розробити) *мета роботи, аналіз користувальницьких і розробка функціональних вимог до програмного продукту,* *опис прийнятих проектних рішень, методи та алгоритми, що використовувались, структура бази даних, опис роботи застосування, тестування ПЗ та аналіз дослідної експлуатації. Додатки: а) наукова публікація, б) довідка про наявність публікації, в) приклади програмних кодів, г) слайди презентації, д) електронні матеріали до проекту на CD.*

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень) *зображення конкуруючих програмних продуктів, схема складників CRM систем, діаграма прецедентів, діаграма розгортання, схема архітектури додатку, фізична схема бази даних проекту, приклади програмного коду, інтерфейс веб-системи.*

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 |
| Снецчастина | проф. Білоус Н.В. |  |  |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва eтапів роботи | Терміни  виконання етапів роботи | Примітка |
| 1 | Об'єктний аналіз поставленої задачі | 25-02-2018 | виконано |
| 2 | Аналіз предметної галузі | 12-03-2018 | виконано |
| 3 | Аналіз існуючих рішень | 15-04-2018 | виконано |
| 4 | Дослідження існуючих методів | 28-05-2018 | виконано |
| 5 | Пошук та вирішення недоліків алгоритмів та систем | 15-05-2018 | виконано |
| 6 | Підготовка пояснювальної записки. | 20-06-2018 | виконано |
| 7 | Підготовка презентації та доповіді | 01-06-2018 | виконано |
| 8 | Попередній захист | 04-06-2018 | виконано |
| 9 | Нормоконтроль, рецензування | 07-06-2018 | виконано |
| 10 | Занесення диплома в електронний архів | 07-06-2018 | виконано |
| 11 | Допуск до захисту у зав. кафедри | 08-06-2018 | виконано |

Дата видачі завдання «22» січня 2018 р.

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис)

Kepiвник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_проф. Білоус Н.В.\_\_

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: x с., y рис., z табл., w додатки, q джерел.

МАШИННИЙ ЗІР, ХАРЧУВАННЯ, НАВЧАННЯ, АВТОМАТИЗАЦІЯ, OPENCV, C++, QT, АНАЛІЗ ЗОБРАЖЕНЬ, БАЗА ДАНИХ.

Об’єктом дослідження є методи і технології машинного зору у застосуванні до дієтології, а саме такі, що можна використати у програмних системах оцінки харчової цінності їжі за зображенням у реальному часі.

Метою роботи є аналіз існуючих програмних систем та методів машинного зору, що вони застосовують; знаходження способів покращення цих систем; розробка моделі програмної системи з використанням пропонованих методів машинного зору.

Пропоновані методи розробки базуються на технологіях C++, Qt, OpenCV, базі даних SQLite.

У результаті роботи було проведено аналіз можливостей існуючих систем, підходів та методів машинного зору що вони використовують; запропоновано спосіб покращення існуючих систем; порівняні та обрані технології розробки програмного забезпечення для оцінки харчової цінності їжі за зображенням у реальному часі.

COMPUTER VISION, NUTRITION, STUDYING, AUTOMATION, OPENCV, C++, QT, IMAGE PROCESSING, DATABASE.

The object of research is methods and technologies of computer vision with application to needs of dietetics, namely those that can be used in a system to estimate nutritional value of food by image in real time.

The aim of this work is to analyze existing systems and methods of computer vision that they use; finding and suggesting a way to improve them; developing a model of a system using proposed solution.

The development methods are based on C++, Qt, OpenCV, SQLite database.

As a result of this work existing systems and their functionality were analyzed and compared; a solution was proposed to improve them; the software development technologies were compared and selected to accomplish real time food calorie estimation based on an image.

ЗМІСТ

|  |  |
| --- | --- |
| Перелік скорочень | 5 |
| Вступ | 6 |
| 1 Аналіз предметної області та постановка задачі | 8 |
| 1.1 Аналіз предметної області | 8 |
| 1.2 Аналіз можливостей існуючих систем | 12 |
| 1.3 Постановка задачі | 15 |
| 2 Методи машинного зору | 18 |
| 2.1 Фільтрація | 18 |
| 2.2 Сегментація | 20 |
| 2.3 Виявлення ознак | 24 |
| 2.4 Модель Bag-of-words | 28 |
| 2.5 Машинне навчання | 31 |
| 3 Експериментальне дослідження | 34 |
| 3.1 Визначення напряму дослідження | 34 |
| 3.2 Алгоритм класифікації | 36 |
| 3.3 Практична реалізація | 37 |
| 4 UML моделювання та вибір технологій розробки | 41 |
| 4.1 UML – моделювання програмної системи | 41 |
| 4.2 Аналіз та вибір технологій розробки | 43 |
| Висновки |  |
| Перелік посилань | 47 |
| Додаток А СТАТЬЯ | 49 |
| Додаток Б Лист від оргкомітету XIV Міжнародної конференції «Стратегія якості в промисловості та освіті» | 0 |
| Додаток В Приклади програмного коду | 0 |
| Додаток Г Слайди презентації | 0 |
| Додаток Д Електронні матеріали атестаційної роботи (CD диск) | 0 |
|  |  |
|  |  |

ВСТУП

Комп'ютерне бачення - це міждисциплінарна галузь, яка займається питаннями створення технологій для отримання високого рівня розуміння цифрових зображень або відеозаписів [1]. З точки зору техніки, вона прагне автоматизувати завдання, які може зробити людська зорова система.

Завдання комп'ютерного зору включають методи отримання, обробки, аналізу та розуміння цифрових зображень та вилучення великогабаритних даних з реального світу з метою отримання числової чи символічної інформації, наприклад, у формах прийняття рішень. Розуміння в цьому контексті означає перетворення візуальних зображень (введення сітківки) в описи світу, які можуть взаємодіяти з іншими процесами мислення та виявляти відповідні дії. Це розуміння зображення можна розглядати як збір символічної інформації з даних зображення за допомогою моделей, побудованих за допомогою геометрії, фізики, статистики та теорії навчання.

Як наукова дисципліна, комп'ютерне бачення стосується теорії за штучними системами, які витягують інформацію з зображень. Дані зображення можуть мати різні форми, такі як відеопослідовності, перегляди з кількох камер або багатовимірні дані з медичного сканера. Як технологічна дисципліна, комп'ютерне бачення прагне застосовувати свої теорії та моделі для побудови систем комп'ютерного зору [2].

Піддомени комп'ютерного бачення включають реконструкцію сцен, виявлення подій, відстеження відео, розпізнавання об'єктів, оцінку 3D-об’єктів, навчання, індексацію, оцінку руху та відновлення зображення.

Завдяки швидкому прогресу смартфонів та планшетів, які на сьогоднішній день мають достатню обчислювальну потужність для розпізнавання зображень у реальному часі, технології машинного зіру набули великий практичний потенціал та використовуються для вирішення різноманітних повсякденних задач. Чотирьохядерні процесори є стандартом для смартфонів, що робить їх обчислювальні можливості більшими за персональний комп’ютер декілька років тому. Раніше системи з розпізнавання зображень мали проводити обчислення на сервері, що вимагає багато додаткових витрат на зв’язок, передачу даних та інше. Тепер це стало можливо на мобільних пристроях, що є набагато більш перспективним напрямом розвитку.

Комп'ютерне бачення охоплює базову технологію автоматизованого аналізу зображень, яка використовується у багатьох областях. Машинним зором, як правило, називається процес об'єднання автоматизованого аналізу зображень з іншими методами та технологіями для забезпечення автоматизації якогось процесу.

Метою цієї роботи є дослідження методів машинного зору для застосування у дієтології, а саме для допомоги притримування здорового образу харчування через автоматизоване оцінювання харчової цінності їжі у реальному часі за допомогою розпізнавання зображень. Система, що буде використовувати ці методи, допоможе користувачам слідкувати за кількістю споживаних калорій, притримуватись дієт та здорово харчуватись.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Основну частину інформації про зовнішній світ людина отримує за допомогою зору і далі обробляє отриману інформацію за допомогою апарату аналізу та інтерпретації візуальної інформації. Тому ще в минулому столітті постало питання про можливість машинної реалізації даного процесу. Машинний зір - необхідний компонент сучасних технологій. Цей елемент є одним з найбільш перспективних методів автоматизації дій із застосуванням комп'ютерних технологій і робототехніки. Системи машинного зору мають на увазі перетворення даних, що надходять з пристроїв захоплення зображення, з виконанням подальших операцій на основі цих даних.

Машинний зір зосереджується на застосуванні, в основному промисловому, наприклад, автономні роботи і системи візуальної перевірки та вимірювань. Це означає, що технології датчиків зображення і теорії управління пов'язані з обробкою відеоданих для управління роботом і обробка отриманих даних в реальному часі здійснюється програмно або апаратно.

Обробка зображень і аналіз зображень в основному зосереджені на роботі з 2D зображеннями, тобто як перетворити одне зображення в інше. Наприклад, попіксельні операції збільшення контрастності, операції по виділенню країв, усунення шумів або геометричні перетворення, такі як обертання зображення. Дані операції припускають, що обробка та аналіз зображення діють незалежно від змісту самих зображень.

Комп'ютерний зір зосереджується на обробці тривимірних сцен, спроектованих на одне або кілька зображень. Наприклад, відновленням структури або іншої інформації про 3D сцену по одному або декільком зображень. Комп'ютерне зір часто залежить від більш-менш складних припущень щодо того, що представлено на зображеннях. Також існує область названа візуалізація, яка спочатку була пов'язана з процесом створення зображень, але іноді мала справу з обробкою і аналізом. Наприклад, рентгенографія працює з аналізом відеоданих медичного застосування.

Розпізнавання образів є областю, яка використовує різні методи для отримання інформації з відеоданих, в основному, засновані на статистичному підході. Значна частина цієї області присвячена практичному застосуванню цих методів.

Таким чином, можна зробити висновок, що поняття «машинний зір» на сьогоднішній день включає в себе: комп'ютерний зір, розпізнавання зорових образів, аналіз і обробку зображень та багато іншого.

Серед основних задач машинного зору можна виділити:

* розпізнавання - класична задача в комп'ютерному зорі, обробці зображень і машинному зорі це визначення містять віддання деякий характерний об'єкт, особливість чи активність (див. рис. 1.1). Це завдання може бути достовірно і легко вирішена людиною, але до сих пір не вирішена задовільно в комп'ютерному зорі в загальному випадку: випадкові об'єкти в випадкових ситуаціях. Один або кілька попередньо заданих або вивчених об'єктів або класів об'єктів можуть бути розпізнані (зазвичай разом з їх двомірним становищем на зображенні або тривимірним становищем в сцені);
* ідентифікація - розпізнавання індивідуального екземпляру об'єкта, що належить до певного класу;
* виявлення – перевірка зазвичай відеоданих на наявність якихось об'єктів або на виконання якихось умов;
* розпізнавання тексту;
* відновлення 3D форми по 2D зображенню - стереореконструкція карти глибини, реконструкція поля нормалей і визначення форми;
* оцінка руху;
* відновлення зображень – видалення шуму, розмитості рухомих об'єктів за допомогою різноманітних фільтрів;
* виділення на зображеннях структур певного виду, сегментація зображень;



Рисунок 1.1 – Результат роботи алгоритму розпізнавання

В даний час машинний зір знаходить найбільше затребування в медицині, робототехніці, автомобільної промисловості, військової галузі та біотехнологіях. Це пов'язано з тим, що в цих галузях вже є чітко сформульовані завдання для комп'ютерного зору, вирішенням яких займаються провідні дослідницькі центри і компанії. При цьому можливе поле застосування машинного зору, по-справжньому, величезне - ці технології можуть бути використані практично у всіх сферах життєдіяльності. За рахунок зростання складності розв'язуваних науково-технічних завдань, автоматична обробка і аналіз візуальної інформації стають все більш актуальними питаннями.

Разом з зростанням кількості та складності потенційних місць застосування технологій машинного зору також зростає загальний рівень технологій у світі, які намагаються вирішити якнайбільше з цих задач. Переламний момент, що викликав появу величезної кількості нових задач для автоматизації за допомогою технологій машинного зору, наступив при появі смартфонів з камерами та потужними процесорами, здатними проводити необхідні обчислення.

Велика кількість цих задач були вирішені и на сьогоднішній день існують додатки для зчитування та перекладу тексту за допомогою камеру смартфону у реальному часі; провідні виробники техніки використовують техніки розпізнавання обличчя для забезпечення безпечної авторизації (див. рис. 1.2); існують додатки які дозволяють примірити на себе одяг та косметику не виходячи з дому; автоматизовані няні, що повідомляють батьків про дії дитини та багато іншого.



Рисунок 1.2 – Розпізнавання обличчя у Apple Face ID

Не виключенням стала і харчова індустрія, де технології машинного зору давно використовуються для забезпечення автоматичного контролю якості продукту та упаковки. Більш актуальною та не повністю вирішеною проблемою стало застосування подібних технологій у дієтології, де з недавніх часів з’явилася потреба в системах, які б допомагали притримуватися здорового образу харчування.

1.2 Аналіз можливостей існуючих систем

На даний момент на ринку представлена велика кількість рішень, що не використовують технології машинного зору або використовують в мінімальному об’ємі, і потребують вводу даних в ручному режимі.

Гарним прикладом таких додатків є MyFitnessPal – додаток для смартфонів та планшетів, що допомагає слідкувати за споживанням калорій та дієтою. Даний додаток має мінімальну автоматизацію за рахунок можливості вводу продукту через сканування баркоду з етикетки на упаковці, у випадках коли це неможливо потрібно вводити дані в ручному режимі (див. рис. 1.3).

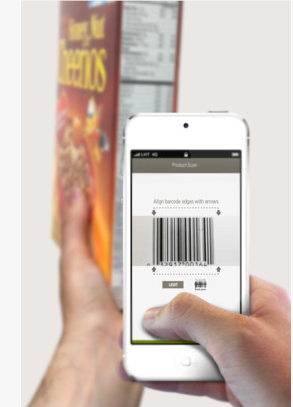


Рисунок 1.3 – Додаток для сканування баркоду

Серед подібних додатків також можна виділити більш комплексні системи, де підтримка образу харчування є лише однією з функцій, наприклад Lose It! – додаток, що допомагає скинути зайву вагу. Серед численних функцій цієї системи: сканер баркодів, можливість підключення фітнес трекерів, завдання і змагання, планування меню та багато іншого.

Гарні позиції мають і більш мінімалістичні додатки типу Simple Calorie Tracker, що складаються з поля для ручного вводу даних і графіку, складеному з історії за останній тиждень.

Також на ринку представлені декілька більш системних рішень, що передбачають застосування спеціалізованого пристрою, наприклад SmartPlate – так звана харчувальна платформа, що складається с спеціальної тарілки та мобільного додатку (див. рис. 1.4).



Рисунок 1.4 – Додаток та пристрій SmartPlate

В даній системі тарілка використовується для вимірювання маси вмісту та у якості еталону розміру. Система передбачає зваження та фотографування тарілки перед їжею, після цього мобільний додаток проводить аналіз зображення і на підставі результатів вираховує харчову цінність продуктів. SmartPlate і подібні рішення мають декілька недоліків, що погано впливають на практичність їх застосування: по-перше, вони працюють лише з однією спеціальною тарілкою, по-друге, не завжди зручно або навіть можливо використовувати свою тарілку.

Крім вищезазначених видів залишається ніша, яка на даний момент майже не представлена на ринку – системи, які б використовували методи машинного зору для автоматичного вводу через обробку фотографій, та не використовували б додаткових пристроїв крім смартфона. У цьому напрямку проводиться робота у останній час і є декілька прототипів та бета-версій рішень [3-4].

Серед методів оцінки харчових параметрів точність результату оцінки визначається двома основними чинниками: алгоритмом класифікації та методом обчислення об'єму. У аспекті розпізнавання класифікаційні алгоритми використовуються для розпізнавання типу продуктів у загальних умовах. У аспекті об'ємної оцінки калібрування їжі та обчислення обсягу є двома ключовими питаннями.

Наприклад, при використанні тарілки як об'єкта калібрування, вона знаходиться за допомогою виявлення еліпсу і обсяг їжі оцінюється за допомогою відповідної об’ємної моделі. Іншим прикладом є використання великого пальця людини як об'єкта калібрування, великий палець виявляється шляхом перетворення колірного простору, а обсяг оцінюється просто, розглядаючи їжу як стовпець. Проте пальці не є стабільним і не гарантовано, що у кожної особи можна виявити великий палець на зображенні. Залучення допомоги людей може підвищити точність оцінки, але споживає більше часу, що робить систему в цілому менш зручною. Після отримання об'єму їжі, калорії їжі розраховується шляхом пошуку її щільності в таблиці щільності харчових продуктів та енергії в таблиці харчування.

Описані вище методи ще потрібно покращити у наступних двох аспектах: використання більш зручних об'єктів калібрування та більш ефективних алгоритмів виявлення об'єктів.

1.3 Постановка задачі

Метою роботи є дослідити методи машинного зору для розпізнавання, класифікації харчових продуктів та оцінки їх об’єму; розробити систему оцінки харчової цінності їжі за її зображенням.

Після проведення відповідних досліджень, створення програмної системи складається з реалізації наступних задач [5-6]:

– інтеграція камери мобільного пристрою для імпортування у додаток та наступної обробки. При цьому треба визначити обмеження для початкових зображень, такі як:

1. роздільна здатність – зображення мають бути достатнього рівня деталізації для забезпечення ефективної роботи методів обробки зображень;
2. навколишнє освітлення – елементи зображення потрібні чітко виділятися один від одного для застосування сегментації;
3. кут, під яким робиться фото – в залежності від обраного алгоритму оцінки об’єму;
4. на зображенні має бути присутній об’єкт калібрації, який буде використовуватись як еталон розміру.

– розпізнавання і класифікація їжі: дослідження та вибір алгоритмів для сегментації елементів зображення та наступної класифікації і розпізнавання вмісту цих елементів;

– калібрація та оцінка об’єму: для визначення об’єму потрібно дослідити алгоритми побудови об’ємних моделей та їх обмеження і на основі цього вибрати зручний еталонний об’єкт та алгоритм калібрації;

– знайти та перетворити в зручний для роботи формат дані про щільність та харчову цінність продуктів;

– ведення історії споживання користувачем калорій: відображення поточного значення за день, відображення графіків споживання за останні тижні/місяці;

– розрахунок та налаштування щоденної норми споживання калорій.

Для реалізації системи було обрано кросплатформенний фреймворк для користувацького інтерфейсу Qt та найбільш широко використовувану бібліотеку з машинного зору OpenCV. За допомогою Qt забезпечена робота додатку у будь якій системі – iOS, Android, Windows та інших. В якості сховища даних була обрана СУБД SQLite, інтегровану у Qt.

Qt включає в себе модулі, які можуть знадобитися при розробці прикладного програмного забезпечення, починаючи від елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних і апаратними функціями пристрою.

OpenCV (Open Source Computer Vision Library) [7] випущена під ліцензією BSD і, отже, безкоштовна як для академічного, так і для комерційного використання. Має інтерфейси C ++, Python і Java і підтримує Windows, Linux, Mac OS, iOS та Android. OpenCV була розроблена з зосередженням на обчислювальної ефективності та сильної спрямованості на додатки в режимі реального часу. Написана в оптимізованому C / C ++, бібліотека може скористатися багатоядерною обробкою. За допомогою OpenCL, вона може користуватися апаратним прискоренням базової неоднорідної обчислювальної платформи.

Прийнята по всьому світу, у OpenCV налічується понад 47 тисяч користувачів. Приблизна кількість завантажень понад 14 мільйонів. Використання від інтерактивного мистецтва до огляду мін, шиття карти в Інтернеті та розширена робототехніка.

У ході виконання роботи необхідно зробити такі завдання:

– провести аналіз, концептуальне та UML-моделювання предметної галузі;

– провести дослідження існуючих підходів та можливостей систем, що використовують методи машинного зору для допомоги притримування здорового харчування;

– розробити алгоритм класифікації зображень продуктів.

2 МЕТОДИ МАШИННОГО ЗОРУ

2.1 Фільтрація

У реальному світі сигнали не існують без шуму, який виникає під час придбання (оцифрування) та / або передачі зображення (див. рис. 2.1). Коли є зображення придбані за допомогою камери, рівні освітленості та температури датчика є основними факторами, що впливають на кількість шуму. Під час передачі зображення пошкоджені головним чином через перешкоди в каналі, що використовується для передачі. Видалення шуму зображень - це важлива проблема при обробці зображень [8].

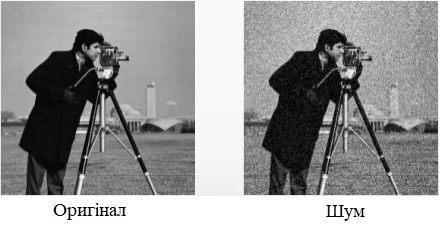


Рисунок 2.1 – Приклад зображення з шумом

Шум - це небажана інформація, яка руйнує якість зображення. У процесі видалення шуму, інформація про тип шуму, присутній у вихідному зображенні відіграє значну роль. Типові зображення пошкоджені шумом, який моделюється як з гауссовським, рівномірним, так і з розповсюдженням солі та перцю.

Гаусовий шум, який має функцію щільності імовірності нормальної розподілу, також відомий як гауссовський розподіл. Інакше кажучи, значення, які може спричинити шум, є гауссово розподіленими. Гауссовський шум належним чином визначається як шум з розподілом амплітуди Гауса.

Шум солі та перцю обумовлений різкими, раптовими порушеннями сигналу зображення, а його поява випадковим чином розподіляється білим або чорним (або обома) пікселями над зображенням. Зображення, що містить шум солі та перцю, матиме темні пікселі у яскравих областях та яскраві пікселі у темних областях (див рис. 2.2). Цей тип шуму може бути викликаний мертвими пікселями, помилками аналогових та цифрових перетворювачів та бітових помилок у передачі



Рисунок 2.2 – Приклад зображення з шумом солі та перцю

Mean фільтрація - це простий, інтуїтивно зрозумілий та простий у застосуванні спосіб зменшення шуму на зображеннях, зменшуючи різницю інтенсивності інтенсивності між одним пікселем і наступним. Він часто використовується для згладжування. Ідея mean фільтрації полягає в тому, щоб просто замінити кожне значення пікселя у образі середнім або середнім значенням сусідів, включаючи саме його. Це призводить до знищення значень пікселів, які не відображають їх оточення. Середнє значення фільтрації зазвичай розглядається як фільтр згортки. Як і інші згортки, воно ґрунтується навколо ядра, яке відображає форму та розмір сусіднього регіону для відбору під час обчислення середнього значення. Часто використовуються ядра з квадратів 3×3 пікселі, хоча для більш жорсткого згладжування можна використовувати більш великі ядра, наприклад квадрати 5×5. Основна проблема mean фільтрації полягає в тому, що один піксель з дуже непредставленим значенням може суттєво вплинути на середнє значення всіх пікселів у його сусідстві. Приклад роботи різних фільтрів представлено на рис. 2.3.

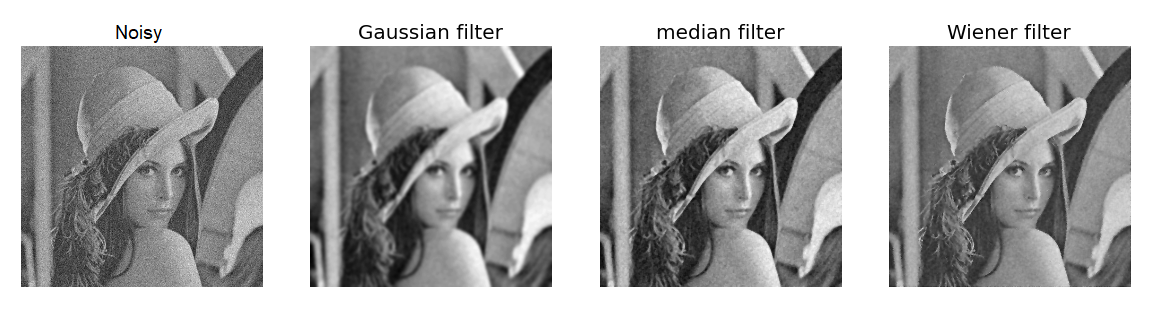


Рисунок 2.3 – Результати роботи різних фільтрів шуму

Медіанний фільтр часто робить кращу роботу, ніж mean фільтр, збереження корисної деталі на зображенні. Як mean фільтр, середній фільтр обмінюється кожним пікселем у зображенні по черзі і розглядає сусідів, що оточують їх, щоб вирішити, чи є вона репрезентативною оточення. Замість того, щоб просто замінити значення пікселя середнім значенням сусідніх значень пікселів, він замінює його медіаною цих значень.

2.2 Сегментація

Сегментація зображення - це процес розбиття цифрового зображення на кілька сегментів (наборів пікселів, також відомих як суперпікселі). Мета сегментації полягає у спрощенні та / або зміні представлення зображення на те, що є більш значущим і простішим для аналізу. Сегментація зображення зазвичай використовується для визначення об'єктів та меж (ліній, кривих тощо) у зображеннях. Точніше, сегментація зображення - це процес присвоєння мітки кожному пікселю у зображенні таким чином, щоб пікселі з однаковими мітками мали певні характеристики.

Результатом сегментації зображення є набір сегментів, які разом покривають все зображення або набір контурів, витягнутих із зображення. Кожен з пікселів у регіоні є подібним щодо певного характеристичного чи обчисленого властивості, наприклад, кольору, інтенсивності або текстури (див. рис. 2.4).

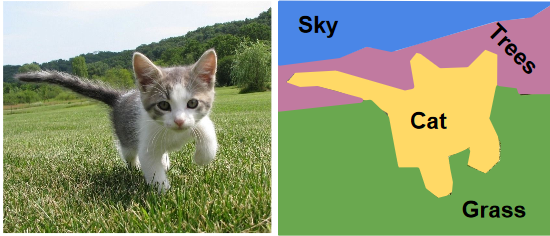


Рисунок 2.4 – Принцип сегментації зображень

Сусідні регіони значно відрізняються по відношенню до однакової характеристики. При застосуванні до стеків зображень, характерних для медичного зображення, результуючі контури після сегментації зображень можуть бути використані для створення 3D-реконструкції за допомогою алгоритмів інтерполяції, таких як marching cubes.

В промисловості використовуються кілька популярних методів, включаючи метод максимального ентропії, метод Оцу (максимальна дисперсія) та кластеризація k-means.

К-means кластеризація - це метод векторного квантування, що був створений для обробки сигналів, а тепер є популярним для кластерного аналізу при пошуку даних. Метод кластеризації k-means спрямований на розбиття n спостережень на k кластерів, в яких кожне спостереження належить до кластера з найближчим середнім значенням, служачи прототипом кластера. Це призводить до розбиття простору даних на клітини Вороного (див. рис. 2.5).

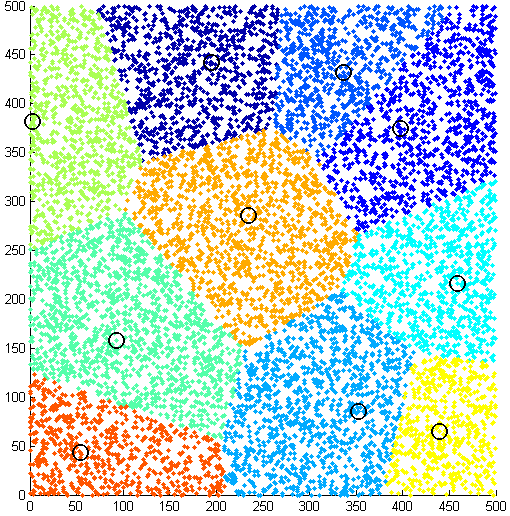


Рисунок 2.5 – Розбиття даних на кластери k-means

Проблема обчислювальної складності (NP-складна); однак існують ефективні евристичні алгоритми, які зазвичай використовуються і швидко сходяться до локального оптимуму. Вони, як правило, аналогічні алгоритму максимізації очікувань для сумішей гауссових розподілів за допомогою ітеративного методу удосконалення, що використовується як моделлю k-means, так і гауссової суміші. Крім того, вони обидва використовують центри кластерів для моделювання даних; однак, k-means кластеризація прагне знайти кластери порівняної просторової межі, тоді як механізм максимізації очікувань дозволяє кластерам мати різні форми.

Алгоритм має вільне відношення до класичного класифікатора k-nearest neighbors, популярної методики машинного навчання для класифікації, яка часто плутається з k-means через назву. Можна застосувати 1-найближчий класифікатор сусідів до кластерних центрів, отриманих за допомогою k-засобу для класифікації нових даних у існуючі кластери.

Найбільш поширеним евристичним алгоритмом є k-means++ - це алгоритм вибору початкових значень (або "насіння") для алгоритму кластеризації k-means. Це було запропоновано в 2007 році Девідом Артуром та Сергієм Васильовицьким, як алгоритм наближення для проблеми NP-hard-k-means - спосіб уникнути іноді поганих кластерингів, виявлених за допомогою стандартного алгоритму k-means [9].

Проблема k-means полягає в тому, щоб знайти кластерні центри, які мінімізують дисперсію всередині класу, тобто сума квадратних відстаней від кожної точки даних кластеризована до його кластерного центру (найближчого до нього центру). Хоча пошук точного рішення задачі k-засобу для довільного введення NP-складний, широко використовується стандартний підхід до пошуку наближеного рішення (часто його називають алгоритмом Ллойда або алгоритмом k-means), який часто знаходить розумні рішення швидко.

Однак алгоритм k-means має принаймні два основні теоретичні недоліки: по-перше, було показано, що найгірший час роботи алгоритму є супер-поліноміальним у розмірі входу; по-друге, знайдене наближення може бути як завгодно поганим стосовно об'єктивної функції порівняно з оптимальним кластеризацією.

Алгоритм k-means++ вирішує другу з цих перешкод, дозволяючи вказати процедуру ініціалізації кластерних центрів, перш ніж продовжувати стандартні ітерації оптимізації k-means. За допомогою ініціалізації k-means++ алгоритм гарантує пошук рішення, яке є O (log k), конкурентоспроможним для оптимального рішення k-means.

2.3 Виявлення ознак

Виявлення ознак включає в себе методи для обчислення абстракцій візуальних даних та прийняття локальних рішень на кожному зображенні, вказує на те, чи присутня на ньому ознака зображення певного типу чи ні. Отримані ознаки будуть підмножинами домену зображення, часто у вигляді ізольованих точок, безперервних кривих або з'єднаних областей (див. рис. 2.6).



Рисунок 2.6 – Виявлення ознак різних типів

Немає універсального чи точного визначення того, що являє собою ознаку, і точне визначення часто залежить від проблеми або типу програми. З огляду на це, ознака визначається як "цікава" частина зображення, і ознаки використовуються як відправна точка для багатьох алгоритмів машинного зору. Оскільки ознаки використовуються як відправна точка та основні примітиви для наступних алгоритмів, загальний алгоритм буде часто настільки ефективний, як його детектор ознак. Отже, бажаною властивістю детектора ознак є повторюваність: чи буде виявлена ​​однакова ознака у двох або більше різних зображеннях однієї і тієї ж сцени.

Виявлення ознак - це обробка зображення на низькому рівні. Тобто, це зазвичай виконується як перша операція на зображенні, і розглядає кожен піксель, щоб побачити, чи присутня ознака на цьому пікселі. Якщо це частина більшого алгоритму, то алгоритм, як правило, розглядає лише зображення в регіоні ознак. В якості вбудованого попереднього визначення ознаки, вхідне зображення зазвичай згладжується у представленому масштабному просторі та обчислюється одне чи кілька зображень ознак, часто виражених у термінах операцій місцевих похідних.

Іноді, коли виявлення ознак має велику обчислювальну складність та існують часові обмеження, алгоритм вищого рівня може бути використаний для керування етапом виявлення ознак, тому для пошуку потрібні лише певні частини зображення. Багато алгоритмів машинного зору використовують виявлення функцій як початковий етап, тому в результаті розробляється дуже велика кількість детекторів. Вони широко розрізняються за видами виявленої ознаки, обчислювальною складністю та повторюваністю. Ознаки можна умовно поділити на кілька типів: ребра, кути або точки інтересу, регіони інтересу та хребти (див. рис. 2.7).

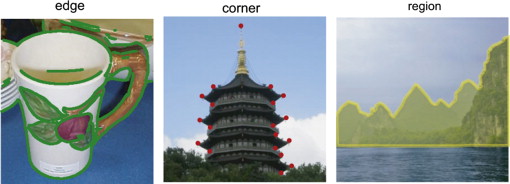


Рисунок 2.7 – Види ознак зображень

Ребра – це точки, де є межа (або край) між двома областями зображення. Загалом, ребро може мати майже довільну форму і може включати в себе з'єднання. На практиці, ребра, як правило, визначаються як набори точок на зображенні, які мають сильну градієнтну величину. Крім того, деякі загальні алгоритми потім об'єднають високі градієнтні точки разом, щоб сформувати більш повний опис ребра. Ці алгоритми зазвичай ставлять певні обмеження на властивості ребра, такі як форма, гладкість та градієнтне значення. На місцевості ребра мають одномірну структуру.

Терміни "кути" та "точки інтересу" використовуються дещо взаємозамінно та стосуються точкових функцій зображення, які мають локальну двомірну структуру. Назва "кут" виникла з тих пір, коли ранні алгоритми вперше виконували виявлення краю, а потім проаналізували краї, щоб знайти швидкі зміни в напрямку. Потім ці алгоритми були розроблені таким чином, що для явного виявлення країв більше не потрібно, наприклад, шукати високий рівень кривизни градієнта зображення. Потім було помічено, що так звані кути також були виявлені на частинах зображення, які не були кутами в традиційному значенні (наприклад, невелика яскрава пляма на темному тлі). Ці моменти часто відомі як точки інтересу, але термін "кут" використовується за традицією.

Регіони інтересу надають додатковий опис структури зображення з точки зору великих фрагментів зображення, на відміну від кутів, які є більш точними (див. рис. 2.8).

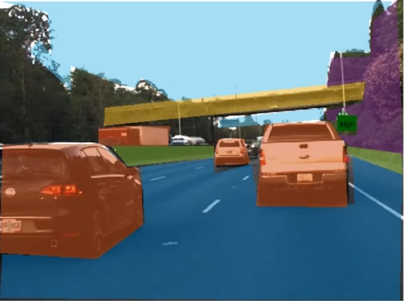


Рисунок 2.8 – Приклад виявлених регіонів інтересу

Тим не менше, дескриптори регіонів інтересу часто можуть містити переважну точку (місцевий максимум реакції оператора або центр тяжкості), що означає, що багато детекторів регіонів також можуть розглядатися як детектори точок інтересу. Детектори регіонів інтересу можуть виявляти ділянки на зображенні, які занадто гладкі, щоб бути виявлені детектором кутів.

Якщо зменшити зображення, а потім виконувати виявлення кутів, детектор буде реагувати на точки, які різкі у зменшеному зображенні, але можуть бути гладкими у вихідному зображенні. Саме в цьому випадку різниця між детектором кута та детектором регіонів стає дещо розпливчастою. значною мірою ці відмінності можна виправити шляхом включення відповідного поняття масштабу.

Для подовжених об'єктів поняття хребтів є природним інструментом. Дескриптор хребта, обчислений із зображення сірого рівня, можна розглядати як узагальнення медіальної осі. З практичної точки зору, хребет можна розглядати як одномірну криву, яка являє собою вісь симетрії, і, крім того, має атрибут локальної ширини хребта, пов'язаної з кожною точкою хребта. Однак, на жаль, алгоритмічно складніше витягувати властивості хребта з загальних класів зображень, крім краєвих, кутових чи чорно-білих. Тим не менше, дескриптори хребта часто використовуються для виявлення дороги на повітряних зображеннях та кровоносних судин у медичних зображеннях (див. рис. 2.9).



Рисунок 2.9 – виявлення хребтів за допомогою SIFT

Speeded Up Robust Features (SURF) - це запатентований детектор і дескриптор локальних ознак. Він може використовуватися для таких завдань, як розпізнавання об'єктів, реєстрація зображень, класифікація або 3D-реконструкція. Частково він надиханий дескриптором Scale-Invariant Feature Transform (SIFT). Стандартна версія SURF в кілька разів швидше, ніж SIFT, і її автори заявляють про більшу надійність проти різних перетворень зображення, ніж SIFT [10].

Для виявлення точок інтересу SURF використовує цілочисельне наближення детермінанта Гессенського блочного детектора, який можна обчислити за допомогою 3 цілочисельних операцій з використанням попередньо інтегрованого зображення. Його дескриптор опирається на суму значень вейвлета Хаара в точці інтересу. Вони також можуть бути обчислені за допомогою інтегрального зображення.

Алгоритм SURF базується на тих самих принципах та етапах, що і SIFT, але деталі на кожному кроці різні. Алгоритм має три основні частини: виявлення точок інтересу, опис локального середовища та підбір. SURF використовує детектор регіонів інтересу на основі матриці Гессе для пошуку цікавих об'єктів. Визначник матриці Гессе використовується як міра локальних змін навколо точки, і точки вибираються там, де цей детермінант має максимальне значення.

2.4 Модель Bag-of-words

Модель "Bag-of-words" (BoW) є спрощувальним представленням, що використовується при обробці природних мов та пошуку інформації. Також відомий як векторна просторова модель. У цій моделі текст (наприклад, речення або документ) представляється як сумка (мультисет) її слів, ігноруючи граматику і навіть порядок слів, але зберігаючи множинність. Модель BoW зазвичай використовується в методах класифікації документів, де частота появи кожного слова використовується як функція для навчання класифікатора.

У машинному зорі модель BoW може бути застосована до класифікації зображень, обробляючи ознаки зображення як слова (див. рис. 2.10). Тобто, гістограма над словником. У машинному зорі мішок візуальних слів є вектором розбіжностей у елементів словника локальних ознак зображення. Для представлення зображення за допомогою моделі BoW зображення можна розглядати як документ. Подібним чином, також слід визначити "слова" у зображеннях. Це, як правило, включає в себе три етапи: виявлення ознак, опис ознак та генерація словника. Визначення моделі BoW може бути "представлення у виді гістограми на основі незалежних ознак".

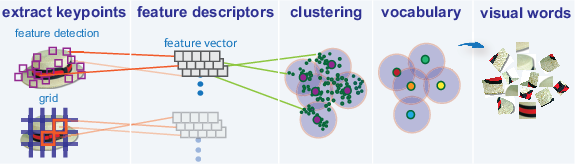


Рисунок 2.10 – Схема роботи BoW для візуальних даних

Після виявлення ознак кожне зображення абстрагується кількома фрагментами. Методи представлення ознак стосуються як представляти фрагменти як числові вектори. Ці вектори називаються дескрипторами ознак. Хороший дескриптор повинен мати здатність певною мірою керувати інтенсивністю, обертанням, масштабом і афінними варіаціями. Одним із дескрипторів, що добре підходять для такої задачі є SURF. Після цього кроку кожне зображення являє собою набір векторів одного і того ж розміру, де порядок різних векторів не має значення.

Останній крок для моделі BoW полягає у перетворенні векторних фрагментів на "кодові слова" (аналогічні словам у текстових документах), що також створює словник. Приклади фрагментів, що можуть вважатися за слова показано на рис. 2.11.

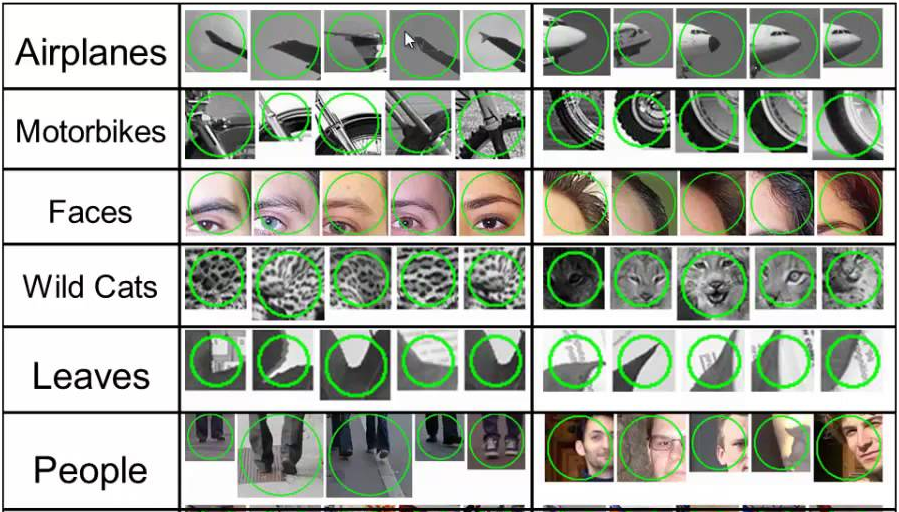


Рисунок 2.11 – приклади візуальних слів

Кодове слово можна розглядати як представник кількох аналогічних фрагментів. Одним із простих методів є виконання k-means кластеризації над усіма векторами. Кодові слова потім визначаються як центри вивчених кластерів. Кількість кластерів - це розмір словника. Таким чином, кожен фрагмент у зображенні відображається до певного кодового слова через процес групування, і зображення може бути представлено гістограмою кодових слів.

2.5 Машинне навчання

Машинне навчання – це підмножина штучного інтелекту в галузі інформатики, яка часто використовує статистичні методи, щоб дати комп'ютерам можливість "вчитися" (тобто поступово покращувати продуктивність для конкретного завдання) з даними, без явного програмування.   
Створене на базі дослідження розпізнавання образів та теорії обчислювального навчання в штучному інтелекті, машинне навчання досліджує вивчення та побудову алгоритмів, які можуть вивчати і робити прогнози щодо даних - такі алгоритми подолають строго статичні інструкції програми, створюючи прогнозовані дані чи рішення, шляхом створення моделі з вхідних даних вибірки. Машинне навчання використовується в ряді обчислювальних завдань, де проектування та програмування явних алгоритмів з високою продуктивністю важко або нездійсненно; Приклади застосування включають фільтрацію електронної пошти, виявлення шкідливих інсайдерів, що працюють над порушеннями даних, оптичне розпізнавання символів, навчання в ранзі та комп'ютерне бачення.

Є чіткі компроміси між традиційним комп'ютерним баченням та підходами з машинним навчанням[11]. Класичні алгоритми комп'ютерного зору є зрілими, перевіреними та оптимізованими для продуктивності та енергоефективності, а машинне навчання забезпечує більшу точність та універсальність - але вимагає великої кількості обчислювальних ресурсів. Тим, хто прагне впровадити високопродуктивні системи, швидко стає зрозуміло, що гібридні підходи, які поєднують традиційне комп'ютерне бачення та машинне навчання, можуть запропонувати найкраще з обох світів. Наприклад, в камері безпеки, алгоритм машинного зору дозволяє ефективно виявляти обличчя або переміщення об'єктів на сцені. Потім менший сегмент зображення, де виявлено обличчя або об'єкт, обробляється шляхом машинного навчання для підтвердження тотожності або класифікації об'єктів - заощаджуючи значні обчислювальні ресурси.

Штучні нейронні мережі (ANNs) є обчислювальними системами, у деякому сенсі натхненними біологічними нейронними мережами, які утворюють тваринний мозок. Такі системи "вчаться" виконувати завдання, розглядаючи приклади, як правило, без програмування з будь-якими правилами конкретних завдань [12]. ANN базується на сукупності пов'язаних одиниць або вузлів, які називаються штучними нейронами (див. рис. 2.12), які вільно моделюють нейрони у біологічному мозку. Кожне з'єднання, як синапси в біологічному мозку, може передавати сигнал від одного штучного нейрона до іншого. Штучний нейрон, який отримує сигнал, може обробити його, а потім сигналізувати додаткові штучні нейрони, пов'язані з ним.



Рисунок 2.12 – загальна схема нейронної мережі

У поширених реалізаціях ANN, сигнал на зв'язці між штучними нейронами є дійсним числом, а вихід кожного штучного нейрона обчислюється деякою нелінійною функцією суми його входів. З'єднання між штучними нейронами називають "краями". Штучні нейрони та ребра звичайно мають вагу, яка налаштовується у процесі навчання. Вага збільшує або зменшує сили сигналу при з'єднанні. Штучні нейрони можуть мати такий поріг, що сигнал надсилається лише тоді, коли сумарний сигнал перетинає цей поріг. Як правило, штучні нейрони агрегуються в шари. Різні шари можуть виконувати різні види перетворень на своїх вкладах. Сигнали проходять від першого шару (вхідного шару) до останнього шару (вихідний шар), можливо, після проходу по декількох шарах. Оригінальна мета методу ANN полягала в тому, щоб вирішити проблеми так само, як і людський мозок, проте з часом увага перейшла до виконання конкретних завдань, що призвело до відхилень від біології.

Багатошаровий персептрон (MLP) – це клас нейронних мереж. MLP складається щонайменше з трьох шарів вузлів. Окрім вхідних вузлів, кожен вузол є нейроном, який використовує нелінійну функцію активації. MLP використовує контрольовану методику навчання, яка називається зворотне поширення. Його кілька шарів і нелінійна активація відрізняють MLP від лінійного персептрона. Він може розрізняти дані, які не є лінійно відокремлюваними [13].

Навчання під контролем - завдання машинного навчання, яке дозволяє вивчати функцію, яка відображає вхідний сигнал на виході на прикладі вхідних та вихідних пар. Він виконує функцію від помічених навчальних даних, що складаються з набору навчальних прикладів. У керованому навчанні кожен приклад - це пара, що складається з вхідного об'єкта (як правило, вектора) та бажаного вихідного значення (також називається контрольний сигнал). Наглядовий алгоритм навчання аналізує тренувальні дані та видає передбачувану функцію, яку можна використовувати для відображення нових прикладів. Оптимальний сценарій дозволить алгоритму правильно визначати мітки класів для нових даних без міток [14].

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Визначення напряму дослідження

Використання методів машинного зору вимагає значних зусиль з різних причин. По-перше, просто кажучи, існує величезна кількість даних для розгляду. Роздільна здатність кадру, що перевищує 1 Мпікс, повинна бути оброблена без стиснення у реальному часі, і кожен піксель кожного кадру вимагає уваги при обробці. Наприклад, типові алгоритми машинного зору для класифікації або відстеження включають комбінацію векторних та скалярних стадій обробки, які потребують інтенсивних обчислень та доступу до зовнішньої пам'яті у випадку традиційних архітектур обробки. Затримка проміжних результатів обробки між процесором і зовнішньою пам'яттю призводить до споживання енергії та збільшує загальну затримку обробки.

Крім того, надійний алгоритм машинного зору для класифікації об'єктів, наприклад, повинен досягати високого рівня продуктивності та надійності. Обережний дизайн алгоритму балансу повинен бути досягнутий між чутливістю (зменшення пропускної швидкості) та надійністю (зменшення "помилкових спрацьовувань"). В алгоритмі класифікації ця вимога вимагає додавання додаткових етапів класифікації для зменшення помилок. Крім того, це може призвести до попередньої обробки кадрів для зменшення шуму або, можливо, етапу нормалізації, щоб усунути нерівності в освітленні або орієнтації об'єкта. Оскільки ці функції бачення набувають переваги у прийнятті та розповсюдженні, толерантність користувачів до ненадійних або неефективних реалізацій зменшиться. Розпізнавання жестів, розпізнавання обличчя та інші функції на основі машинного зору також повинні стати більш витонченими та надійними для надійної роботи в широкому діапазоні середовищ. Ці вимоги переводяться в більшу кількість пікселів, більш складну обробку, а отже, навіть складнішу проблему для звичайних архітектур обробки.

На стороні обробки, на перший погляд, може здатися, що існує велика кількість процесорних можливостей у сучасних процесорів мобільних додатків, оскільки вони містять двоядерні або чотириядерні процесори, що працюють на частотах 1 ГГц і вище. Але процесори не оптимізовані для інтенсивних, паралельно розрядних завдань і критично для мобільних пристроїв – їх процесори не є найбільш енергоефективними для виконання завдань машинного зору. Сьогоднішні процесори мобільних додатків включають цілий ряд спеціалізованих співпроцесорів для таких завдань, як відео, 3D-графіка, аудіо та поліпшення зображення. Ці співпроцесори дозволяють смартфонам і планшетам доставляти вражаючі мультимедійні можливості із тривалим терміном служби акумулятора. Хоча деякі з цих же співпроцесорів можуть бути використані для завдань машинного зору, жоден з них не був спеціально розроблений для вирішення таких завдань. Оскільки попит на машинний зір збільшується на мобільних процесорах, спеціалізовані спільні процесори зору, швидше за все, будуть додані до ряду існуючих співпроцесорів, що забезпечить високу продуктивність і підвищує енергоефективність, але на сьогоднішній день таких процесорів не існує.

У розрізі дієтології, найбільш поширеним та важким з огляду на вищеописані проблеми кроком є класифікація. По-перше, існує велика кількість категорій продуктів харчування. Створення набору даних усіх категорій продуктів само по собі є складним завданням. По-друге, може існувати велика кількість внутрішньокласових варіацій в продуктах харчування, які ми спостерігаємо. Одні й ті самі продукти можуть мати дуже різний зовнішній вигляд. Нарешті, наявність оклюзій навколо продуктів харчування складає додаткову складність для її розпізнавання, однакова їжа може бути подана на тарілці або загорнута в паперову обкладинку, або якось по-іншому.

В рамках даної роботи було вирішено розробити алгоритм класифікації харчових продуктів, оптимальний для роботи на мобільному пристрої у реальному часі. Додаток з використанням цього алгоритму повинен займати мінімальний обсяг на диску та мати прийнятний рівень затримки при роботі.

3.2 Алгоритм класифікації

Першим кроком у розробці алгоритму класифікації харчових продуктів став пошук вибірки первинних даних для тестування та потенційного навчання нейронної мережі [15]. Ці дані повинні бути класифіковані людиною. У відкритому доступі було знайдено Food-11 – набір даних, що містить 16643 зображення продуктів харчування, згрупованих у 11 основних категорій продуктів харчування: хліб, молочний продукт, десерт, яйце, смажена їжа, м'ясо, локшина або макаронні вироби, рис, морепродукти, суп та овочі або фрукти. Використовується однакова конвенція імен, де ID 0-10 відноситься до 11 категорій продуктів відповідно. Розмір вибірки на диску складає близько 1.16GB [16].

Вважаючи на розміри вибірки, що покриває лише маленьку частину справжнього різноманіття харчових продуктів, очевидно що зберігати ці зображення та навіть їх дескриптори на диску не практично. У даній ситуації добре працює метод виявлення ознак, наприклад Bag of Words, де кількість мішків може бути підібрана мінімальною при задовільній точності роботи алгоритму, а словник візуальних слів можна побудувати за допомогою SURF. В результаті можемо зберігати на диску лише словник, що містить гістограми – векторні дані, розмір яких пропорційний лише кількості класів продуктів.

Для забезпечення мінімальної затримки при класифікації продуктів харчування у реальному часі має сенс використати методи машинного навчання. Для даної задачі добре підходить нейронна мережа MLP, тому що вона є глибокою мережею, де кількість шарів буде відповідна кількості слів у словнику BoW, а розмір шарів пропорційний кількості класів продуктів. Таким чином MLP може досягти високої точності класифікації в умовах великої кількості класів. Також ця мережа зручна для навчання під наглядом.

3.3 Практична реалізація

Для реалізації скористаємось бібліотекою OpenCV. Алгоритми та типи даних для реалізації алгоритму класифікації харчових продуктів знаходяться у наступних модулях: features2d, xfeatures2d, core, imgcodecs, ml [17].

Спочатку побудуємо словник візуальних слів для BoW. Для цього скористаємось детектором SURF та зберемо дескриптори всіх первинних даних. Далі створимо словник за допомогою k-means кластеризації з використанням k-menas++ та запишемо його у файл (див. рис. 3.1).



Рисунок 3.1 – Побудова словника BoW

Наступний крок – тренування нейронної мережі. Скористуємось реалізацією MLP у OpenCV – класом ANN\_MLP. Встановимо відповідну кількість та розмір шарів нейронів; у якості функції активації виберемо сигмоїду; у якості методу тренування – зворотне поширення (Backpropagation) [18]. Перед початком тренування треба сформувати тренувальні дані. Дана реалізація MLP вимагає подачу тренувальних даних у вигляді двох матриць – тренувальних даних та поміток. В даному випадку матриця тренувальних даних буде мати кількість стовпців відповідно кількості слів у словнику. Матриця поміток має містити по стовпцю для кожного класу даних, для кожного рядка стовпець класу відповідного по індексу екземпляру тренувальних даних має містити одиницю, інші стовпці – нуль. Перетворення тренувальних зображень на відповідні гістограми схоже на попередній крок: для кожного зображення виявимо ознаки за допомогою SURF детектора, а потім за допомогою алгоритму Approximate Nearest Neighbors (FLANN) спробуємо знайти відповідні ознаки у словнику [19]. В результаті отримаємо гістограму, яку помістимо в матрицю з тренувальними даними. Після підготовки даних тренуємо нейронну мережу і зберігаємо у файл (див. рис. 3.2).

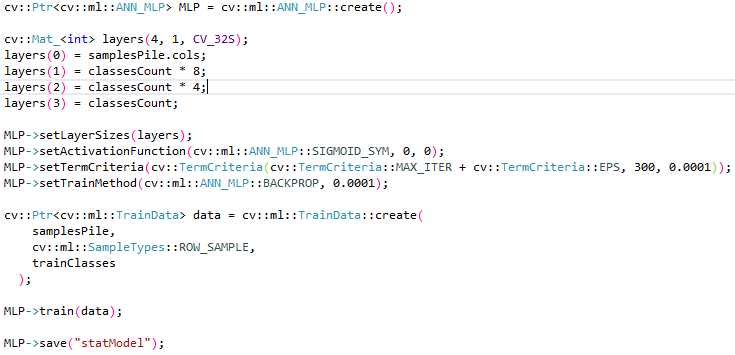


Рисунок 3.2 – Тренування нейронної мережі

У результаті тренування отримаємо відповідні вагові коефіцієнти нейронів, які будуть використані у наступному кроці при класифікації тестувальних даних [20].

Процес тестування аналогічний тренуванню, але без забезпечення мітками з правильним класом вхідних даних. Отримуючи гістограми для кожного тестового зображення, будемо передбачати його клас за допомогою нейронної мережі; заміримо час, необхідний на перетворення та передбачення (див. рис. 3.3).



Рисунок 3.3 – Тестування нейронної мережі

В результаті отримано середній час обробки та класифікації одного зображення 36.87мс, що є прийнятним для використання у реальному часі. У процесі побудови словника BoW та тренування нейронної мережі було використано 13191 зображення сумарним розміром 839MB, розмір відповідних файлів словника та MLP, необхідних для завантаження та повторного використання сумарно склав 1010KB.

4 UML МОДЕЛЮВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

4.1 UML – моделювання програмної системи

Моделювання розроблюваної системи проводиться з використанням мови UML для побудови діаграм, що допоможуть відобразити функціональність та внутрішню структуру системи. UML - мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення [21]. UML є мовою широкого профілю, це відкритий стандарт, який використовує графічні позначення для створення абстрактної моделі системи, що називається UML-моделлю.

Можна виділити наступні переваги UML:

– UML об'єктно-орієнтований, в результаті чого методи опису результатів аналізу і проектування семантично близькі до методів програмування на сучасних об'єктно-орієнтованих мовах;

– діаграми UML порівняно прості для читання після досить швидкого ознайомлення з його синтаксисом;

– UML розширює і дозволяє вводити власні текстові та графічні стереотипи.

При проектуванні системи були розроблені діаграми наступних типів: діаграма прецедентів та діаграма розгортання.

Діаграма прецедентів описує функціональне призначення системи або те, що система повинна робити [22]. Розробка діаграми має такі цілі:

– визначити контекст предметної області, яка моделюється;

– сформулювати загальні вимоги до функціонального поведінки проектованої системи;

– розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей.

Суть діаграми прецедентів полягає в наступному. Проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою варіантів використання. При цьому актором або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Діаграма прецедентів для даної системи відображена на рисунку 4.1.

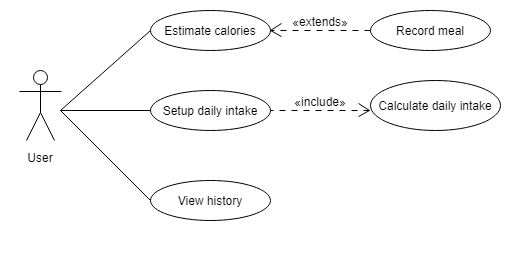


Рисунок 4.1 – Діаграма прецедентів

Діаграма прецедентів моделює систему з точки зору користувача і відображає доступні йому можливості. Діаграми цього типу допомагають при аналізі вимог до системи та проектуванні інтерфейсу користувача.

На рисунку 3.1 зображені основні функції системи, що доступні користувачу. Основний функціонал, доступний користувачу системи включає в себе наступні можливості:

– оцінка харчової цінності їжі за зображенням з камери;

– збереження даних з результату оцінки у історію споживання поточного дня;

– налаштування добової норми споживання калорій;

– вирахування рекомендованої добової норми споживання калорій за загальноприйнятими методами;

– перегляд історії споживання калорій у вигляді графіків та гістограм;

4.2 Аналіз та вибір технологій розробки

Методи та алгоритми машинного зору як правило потребують великі об’єми обчислень і є достатньо вимогливими до продуктивності. Зважуючи на це, для розробки рішень, що включають в себе методи машинного зору було обрано мову програмування C++. Вона має імперативні, об'єктно-орієнтовані та загальні функції програмування, а також забезпечує можливості для маніпулювання пам'яттю на низькому рівні.

Ця мова програмування була розроблена з упередженням до системного програмування та вбудованих, обмежених ресурсами та великих систем, з продуктивністю, ефективністю та гнучкістю використання, як його основні моменти [23]. C++ також виявився корисним у багатьох інших контекстах, де його ключовими сильними сторонами є інфраструктура програмного забезпечення та обмежені ресурсами програми, включаючи настільні додатки, сервери (наприклад, електронна комерція, веб-пошук або SQL-сервери) та додатки, де продуктивність обчислень критична ( наприклад, телефонні перемикачі або космічні зонди).

Таким чином, мова програмування С++ добре підходить для вирішення будь-яких задач, які включають в себе реалізацію методів машинного зору, тому що такі задачі як правило включають в себе процеси високого рівня та потребують найбільшої можливої продуктивності.

OpenCV (Open Source Computer Vision Library) - це бібліотека програмного забезпечення з відкритим кодом для комп'ютерного зору та машинного навчання. OpenCV був побудований, щоб забезпечити загальну інфраструктуру для комп'ютерних програм зору та прискорити використання цих технологій у комерційних продуктах. OpenCV випускається під ліцензією BSD, що дозволяє підприємствам легко використовувати та змінювати код. Основна функціональність OpenCV полягає у вилученні та обробці значущих даних з зображень. Ці значущі дані можуть включати в себе пошук об'єктів або їх частин, розпізнавання об'єктів або їх частин, відстеження руху об'єктів в 2D або 3D між послідовними зображеннями, визначення 2D або 3D-форми об'єктів з одного або декількох зображень, асоціювання даних зображення із категорією тощо.

Бібліотека має понад 2500 оптимізованих алгоритмів, що включають в себе повний комплект як класичного, так і сучасного комп'ютерного бачення та алгоритмів машинного навчання. Ці алгоритми можуть бути використані для виявлення та розпізнавання облич, визначення об'єктів, класифікації людських дій у відео, рухання руху камери, відстеження рухомих об'єктів, витягування 3D-моделей об'єктів, створення 3D-точкових хмар із стереокамер, зшивання зображень разом для отримання високої роздільної здатності зображення цілої сцени, пошука схожих зображень з бази даних зображень, видалення червоних очей з фотографій, стеження за рухом очей тощо.

OpenCV має на меті зробити комп'ютерне бачення доступним програмістам і користувачам у галузі взаємодії людини та комп'ютера в режимі реального часу та мобільної робототехніки. Таким чином, бібліотека поставляється з вихідними кодами, оптимізованими так що користувачі можуть як вчитися в бібліотеці, так і використовувати її у промислових умовах. Під час створення та запуску додатка, що використовує OpenCV, у процесі запуску викликається вбудований DLL-перемикач, який автоматично визначає тип процесора та завантажує відповідну оптимізовану бібліотеку DLL для цього процесора. Якщо тип процесора не може бути визначений (або якщо відповідна бібліотека DLL недоступна), використовується оптимізована C-DLL.

Розгорнуті програми OpenCV охоплюють діапазон від відео спостереження в Ізраїлі, моніторингу обладнання шахти в Китаї, допомога роботам в навігації та зйомці об'єктів, перевірка злітно-посадкових смуг на сміття в Туреччині, перевірка етикеток на продуктах на заводах у всьому світі до швидкого розпізнавання обличчя в Японії.

Він має інтерфейси C++, Python, Java та MATLAB і підтримує Windows, Linux, Android та Mac OS. OpenCV спрямовується в основному на додатки в режимі реального часу та використовує інструкції MMX та SSE, коли це можливо. В даний час активно розвиваються повнофункціональні інтерфейси CUDA та OpenCL. Існує понад 500 алгоритмів і приблизно в 10 разів більше функцій, які складають або підтримують ці алгоритми. OpenCV написано на C++.

Для того, що б мати можливість випустити додаток на всі популярні мобільні платформи, доцільно використати кросплатформенний фреймворк [24], одним з яких і найбільш зручним у даній задачі є Qt.

Qt - це кросплатформенна технологія для додатків, яка використовується для розробки прикладного програмного забезпечення, яке може працювати на різних програмних та апаратних платформах, з незначними або ніякими змінами в основному коді, в той час як все ще є нативною програмою з повними можливостями і швидкістю [25].

Ключовими характеристиками Qt є повна абстракція графічного інтерфейсу, слот-сигнали та мета-компілятор. Коли вперше випустили, Qt використовував свій власний механізм малювання та елементи керування, наслідуючи зовнішній вигляд різних платформ, на яких він запускається, коли він малював свої віджети. Це полегшило роботу з перенесенням, оскільки дуже мало класів в Qt залежать від цільової платформи; однак це іноді призводило до незначних розбіжностей, коли така емуляція була недосконалою. Останні версії Qt використовують нативний стиль API різних платформ, на платформах, що мають власний набір віджетів, для запиту метрик та більшості елементів керування, і не страждають від таких проблем. На деяких платформах (наприклад, MeeGo і KDE) Qt є нативним API. Деякі інші портативні графічні набори інструментів приймають різні конструктивні рішення; Наприклад, wxWidgets використовує набір інструментів цільової платформи для реалізації. Слот-сигнали - мовна конструкція, введена в Qt для спілкування між об'єктами, що полегшує реалізацію схеми спостереження, одночасно уникаючи повторів коду. Концепція полягає в тому, що віджети графічного інтерфейсу можуть надсилати сигнали, що містять дані про події, які можуть бути отримані іншими елементами керування, використовуючи спеціальні функції, відомі як слоти. Компілятор метаоб'єктів, який називається moc, є інструментом, який запускається на вихідному коді програми Qt. Він інтерпретує певні макроси з коду C++ як анотації і використовує їх для створення додаткового C++ коду з мета-інформацією про класи, що використовуються в програмі. Ця мета-інформація використовується Qt для забезпечення функцій програмування, які не є вихідними в C++: сигнали та слоти, інтроспекції та асинхронні виклики функцій.

ВИСНОВКИ

У ході атестаційної роботи магістра було проведено дослідження сучасних підходів та методів машинного зору у застосуванні до завдань дієтології; проведено аналіз предметної області; запропоновано спосіб покращення існуючих систем за допомогою алгоритму класифікації продуктів; спроектоване програмне забезпечення для підтримки здорового харчування шляхом розпізнавання харчових продуктів, аналізу їх харчової цінності через фотографію; ведення журналу, раціону та статистики.

На початку роботи було проведено аналіз предметної області, в процесі якого були визначені основні проблеми, що існують в розглянутій галузі. Було проведено аналіз існуючих CRM систем, які можна застосовувати на навчальному центрі. Особливу увагу було приділено модулю розкладу. Було виявлено їх основні можливості та недоліки. На підставі аналізу предметної області була проведена постановка завдання.

У ході виконання поставленої задачі було:

* проведено аналіз та UML-моделювання предметної галузі;
* проведено аналіз можливостей існуючих систем;
* проведено дослідження підходів формування розкладу та розроблений алгоритм додавання заходів до розкладу центру;
* виконано програмну реалізацію веб-системи та мобільного додатку;
* проведено тестування розробленого програмного продукту.

Кінцевий продукт виконує функції, які були заплановані під час постановки завдання:

– реєстрація користувача у системі;

– отримання розкладу занять користувачами;

– надання можливостей додання пар до розкладу центру користувачем типу «викладач» , використовуючи критерії аудиторій, дати та часу початку та закінчення пари, вводу опису події, перевірки на накладання з розкладом груп студентів університету;

– надання можливості викладачу переглядати статус усіх своїх запитів на додання подій до календаря НВЦА;

– надання можливості адміністратору керувати заходами, шляхом можливості їхнього підтвердження чи відхилення;

– отримання звітів адміністратором.

Створений у процесі виконання роботи програмний продукт є актуальним для викладачів та студентів Навчального Науково-Виробничий Центр аутсорсингу, що діє на базі університету ХНУРЕ.

У перспективі розглядається вдосконалення даного продукту шляхом додавання нових можливостей таких, вдосконалення алгоритму додання заходів до розкладу навчального центру, впровадження інтеграції з персональним календарем викладача (наприклад з Google календарем) для запобігання накладань у заходах центр, університету та власних подіях.

По результатам роботи була зроблена презентація.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bishop, C. Pattern Recognition and Machine Learning [Текст] / C. Bishop – Springer, 2006. – 738 c. – ISBN 13: 9780387310732

2. Szeliski, R. Computer Vision: Algorithms and Applications [Текст] / R. Szeliski. – Springer, 2010. – 979 c. – ISBN 13: 978-1848829343.

3. SmartPlate [Електронний ресурс] / SmartPlate. – Режим доступу: www/ URL: https://www.getsmartplate.com – Загол. з екрану.

4. BiteSnap [Електронний ресурс] / BiteSnap. – Режим доступу: www/ URL: https://www.getbitesnap.com – Загол. з екрану.

5. Real-time Mobile Food Recognition System [Електронний ресурс] / Semantic Scholar, Режим доступу: www/ URL: https://pdfs.semanticscholar.org/92f4/2503f0e6cf98bf8b97a139002e66f4476849.pdf . – Загол. з екрану.

6. Computer Vision-based Food Calorie Estimation: Dataset, Method, and experiment [Електронний ресурс] / arXiv.org, Режим доступу: www/ URL : https://arxiv.org/pdf/1705.07632.pdf . – Загол. з екрану.

7. OpenCV [Електронний ресурс] / OpenCV. - Режим доступу: www/ URL: https://opencv.org/ - Загол. з екрану.

8. Gonzalez R. C., Woods R.E., Digital Image Processing (4th Edition) [Текст] / R. C. Gonzalez, R. E. Woods. - Pearson, 2018. - 1148 c. - ISBN 13: 9780133356724

9. k-means++: The Advantages of Careful Seeding [Електронний ресурс] / Stanford Library. - Режим доступу: www/ URL: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf - Загол. з екрану.

10. SURF: Speeded Up Robust Features [Електронний ресурс] / Computer Vision Laboratory, ETH Zurich. - Режим доступу: www/ URL: http://www.vision.ee.ethz.ch/~surf/eccv06.pdf - Загол. з екрану.

11. Bishop, C. Neural Networks for Pattern Recognition [Текст] / C. Bishop – Oxford University Press, 1995. – 576 c. – ISBN: 0198538642

12. Fausett, L. V., Fundamental of Neural Networks: Architectures, Algorithms, and Applications [Текст] / L. V. Fausett. - Prentice-Hall, 1993. - 463 c. - ISBN: 0133341860

13. Girshick R., Donahue J., Darrell T., Malik J., Rich feature hierarchies for accurate object detection and semantic segmentation [Текст] / R. Girshick, J. Donahue, T. Darrell, J. Malik , Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference, 2014.

14. Krizhevsky A., Sutskever I., Hinton G. E., ImageNet Classification with Deep Convolutional Neural Networks [Текст] / A. Krizhevsky, I. Sutskever, G. E. Hintonin, NIPS, 2012.

15. FooDD: Food Detection Dataset for Calorie Measurement Using Food Images [Електронний ресурс] / University of Ottawa. – Режим доступу: www/ URL: http://www.site.uottawa.ca/~shervin/pubs/FoodRecognitionDataset-MadiMa.pdf - Загол. з екрану.

16. Food image datasets [Електронний ресурс] / Ecole Polytechnique Fédérale de Lausanne. - Режим доступу: www/ URL: https://mmspg.epfl.ch/food-image-datasets - Загол. з екрану.

17. Suarez, O. D. et al. OpenCV Essentials [Текст] / O. D. Suarez et al. – Packt Publishing, 2014. – 214 c. – ISBN 13: 9781783984244.

18. Beyeler, M., Machine Learning for OpenCV [Текст] / M. Beyeler, Packt Publishing, 2017. - 382 c. - ISBN 13: 978-178398028-4

19. FLANN - Fast Library for Approximate Nearest Neighbors [Електронний ресурс] / University of British Columbia library . - Режим доступу: www/ URL: https://www.cs.ubc.ca/research/flann/ - Загол. з екрану.

20. Laganiere, R., OpenCV 3 Computer Vision Application Programming Cookbook - Third Edition [Текст] / R. Laganiere, Packt Publishing, 2017. - 474 c., - ISBN 13: 9781786469717

21. Фаулер, М. UML. Основи, 3-є видання: пер. з англ. [Текст] / М. Фаулер. – СПб: Символ-Плюс., 2004. – 192 с. – ISBN 5-93286-060-Х.

22. Діаграма варіантів використання (use case diagram) [Електронний ресурс] / UML Теорія – Режим доступу: http://www.info-system.ru/designing/methodology /uml/theory/use\_case\_diagram\_theory.html/ . – Загол. з екрану.

23. Meyers, S. Effective C++ [Текст] / S. Meyers. – Pearson Education, 2005. – 320 c. – ISBN: 0321334876.

24. Фаулер, М., Дейвид, Р. Архитектура корпоративных программных приложений [Текст] / М. Фаулер, Д. Райс; – М.: Вильямс, 2008. – 544 с.

25. Rischpater, R. Application Development with Qt Creator [Текст] / R. Rischpater. – Packt Publishing, 2013. – 138 c. – ISBN 13: 9781783282319.