

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**  
**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

**Мобільний додаток розпізнавання харчових продуктів**

---

---

---

Освітній ступінь «бакалавр»  
Спеціальність 122 – «комп'ютерні науки»

Керівник проекту:

\_\_\_\_\_

(підпис)

**Барбарук В.М.**

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

**Критська Я.О.**

\_\_\_\_\_

(ініціали, прізвище)

Здобувач вищої освіти:

\_\_\_\_\_

(підпис)

**Панченко К.О.**

\_\_\_\_\_

(ініціали, прізвище)

Група:

**КН-15з**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітній ступінь бакалавр  
Напрямок підготовки \_\_\_\_\_  
(шифр і назва)  
Спеціальність 122 – комп'ютерні науки  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри КНІ  
І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Панченко Катерині Олександрівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Мобільний додаток розпізнавання харчових продуктів

керівник проекту (роботи) Барбарук Віктор Миколайович, к.т.н., доц.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "13" 05 2019 р. № 83/15.15

2. Термін подання студентом роботи 16.06.2019

3. Вихідні дані до роботи Математичні моделі обробки зображень, програмна реалізація додатку за допомогою мови програмування Java для операційної системи Android

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Огляд основних методів виділення та аналізу ознак для пошуку овочів та фруктів на зображенні, реалізація моделі методу віолі-джонса, розробка архітектури та реалізація комплексу, охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст. викл. Критська Я.О.		

7. Дата видачі завдання 30.04.2019

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Аналіз завдання та робота з літературою	05.05.2019 - 13.05.2019	
2	Аналіз технічних засобів	14.05.2019 - 22.05.2019	
3	Розробка алгоритму	22.05.2019 - 02.06.2019	
4	Програмна реалізація	02.06 .2019- 11.06.2019	
5	Оформлення пояснювальної записки	11.06.2019 - 14.06.2019	
6	Підготовка презентації та доповіді	14.06.2019 - 16.06.2019	

Здобувач вищої освіти

\_\_\_\_\_ ( підпис )

Панченко К.О.

\_\_\_\_\_ (прізвище та ініціали)

Керівник

\_\_\_\_\_ ( підпис )

Барбарук В.М.

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка дипломної роботи: 74с., 20 рис., 2 додатки, 22 джерела.

Робота присвячена вирішенню проблеми розпізнавання зображень. Реалізовано метод Віоли-Джонса з використанням каскадів Хаара. Він полягає в аналізі інтенсивностей суміжних пікселів в прямокутних областях зображення. Розроблені методи дозволяють знаходити відповідні перетини інтенсивностей вхідного зображення з еталонними значеннями, які були отримані аналізом позитивних та негативних характеристик класу, одержувати області таких перетинів, що дозволяє з великою ймовірністю віднести знайдений об'єкт до певного класу.

Розроблені алгоритми реалізації запропонованих методів використано для створення програмного комплексу по розпізнаванню овочів і фруктів в продуктивній корзині.

**Ключові слова:** ШТУЧНИЙ ІНТЕЛЕКТ, РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ, КАСКАД, ЕТАЛОН, МІРА ПОДІБНОСТІ.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400с.

## ЗМІСТ

1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ВИДІЛЕННЯ ТА АНАЛІЗУ ОЗНАК ДЛЯ ПОШУКУ ОВОЧІВ ТА ФРУКТІВ НА ЗОБРАЖЕННІ .....	8
1.1 Аналіз предметної області розпізнавання образів .....	8
1.2 Методи сегментації зображення за кольором фруктів або овочів .....	13
1.2.1 Визначення границь кластеру .....	13
1.2.2 Моделювання кольору об'єктів непараметричними методами .....	14
1.3 Штучна нейронна мережа .....	15
1.4 Перцептрон .....	18
1.5 Штучний нейрон .....	19
1.6 Методи дистантного перетворення .....	21
1.7 Постановка задачі .....	22
2 РЕАЛІЗАЦІЯ МОДЕЛІ МЕТОДУ ВІОЛИ-ДЖОНСА .....	24
2.1 Інтегральне представлення в методі Віоли-Джонса .....	24
2.2 Обґрунтування вибору середовища програмної реалізації .....	29
2.2 Базова інформація про мову програмування Java .....	30
2.3 Тип бази даних для розробленого додатку .....	33
3 РОЗРОБКА АРХІТЕКТУРИ ТА РЕАЛІЗАЦІЯ КОМПЛЕКСУ .....	35
3.1 Проектування програмного комплексу .....	35
3.2 Activity і його стандартна реалізація .....	37
3.2 Життєвий цикл додатку на Android .....	38
3.3 Архітектура бази даних в контексті SQLite .....	40
3.4 Тестування розробленого додатку .....	42
4 ОХОРОНА ПРАЦІ .....	49
4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал .....	49
4.2 Заходи щодо техніки безпеки .....	50
4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці .....	53
4.4 Рекомендації по пожежній безпеці .....	57

	5
ВИСНОВКИ .....	62
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	64
ДОДАТОК А. Тестові зображення.....	67
ДОДАТОК Б. Електронні плакати .....	70

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СТЗ – система технічного зору;

ШНС – штучна нейронна мережа;

API – application programming interface;

ОС – операційна система;

IDE – Integrated Development Environment;

БД – база даних.

## ВСТУП

Із завданням розпізнавання образів живі системи, у тому числі і людина, стикаються постійно з моменту своєї появи. Зокрема, інформація, що надходить з органів почуттів, обробляється мозком, який у свою чергу сортує інформацію, забезпечує прийняття рішення, а далі за допомогою електрохімічних імпульсів передає необхідний сигнал далі, наприклад, органам руху, які реалізують необхідні дії. Потім відбувається зміна навколишнього оточення, і вищевказані явища відбуваються заново. І якщо розібратися, то кожен етап супроводжується розпізнаванням.

З розвитком обчислювальної техніки стало можливим вирішити ряд завдань, що виникають в процесі життєдіяльності, полегшити, прискорити, підвищити якість результату. Наприклад, робота різних систем життєзабезпечення, взаємодія людини з комп'ютером, поява роботизованих систем та ін. Тим не менш, відзначимо, що забезпечити задовільний результат в деяких завданнях (розпізнавання подібних об'єктів, що швидко рухаються, рукописного тексту) на даний час не вдається.

Нині існує величезна різноманітність підходів для виявлення об'єктів на фотографіях. Найбільш досконалим методом розпізнавання на сьогоднішній день є нейронні мережі. Але вони досить складні у розробці, тому конкурентно спроможні рішення розробляються великими корпораціями. У той же час існують алгоритми і рішення, які є сучасними і ефективними та можуть бути реалізовані без значних зусиль. У даній роботі досліджено той метод, який можливо реалізувати у мобільних додатках при невеликих обсягах пам'яті. Ми будемо використовувати алгоритм Віоли-Джонса для пошуку фруктів та овочів на зображенні та ознаки Хаара для опису кожного об'єкту.



# **1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ВИДІЛЕННЯ ТА АНАЛІЗУ ОЗНАК ДЛЯ ПОШУКУ ОВОЧІВ ТА ФРУКТІВ НА ЗОБРАЖЕННІ**

## **1.1 Аналіз предметної області розпізнавання образів.**

Розпізнавання образів – це розділ теорії штучного інтелекту, що вивчає методи класифікації об'єктів. За традицією об'єкт, що піддається класифікації, називається образом.

Образом може бути цифрова фотографія (розпізнавання зображень), буква або цифра (розпізнавання символів), запис мови (розпізнавання мови) тощо. В межах теорії штучного інтелекту розпізнавання образів включається в більш широку наукову дисципліну – теорію машинного навчання, метою якої є розробка методів побудови алгоритмів, що здатні навчатися. Існує два підходи до навчання: індуктивне і дедуктивне.

Індуктивне навчання, або навчання за прецедентами, засноване на виявленні загальних властивостей об'єктів на підставі неповної інформації, отриманих емпіричним шляхом.

Дедуктивне навчання передбачає формалізацію знань експертів у вигляді баз знань (експертних систем тощо). Слід зауважити, що, як кожна математична дисципліна, розпізнавання образів має власний математичний апарат, який включає математичну статистику, методи оптимізації, дискретну математику, алгебру і геометрію. Розпізнавання образів має широке застосування і використовується при створенні усіх комп'ютерних систем, на які покладаються інтелектуальні функції, тобто функції, пов'язані із прийняттям рішень замість людини: медична діагностика, криміналістична експертиза, пошук інформації та інтелектуальний аналіз даних тощо.

Прецедент — це об'єкт, приналежність якого до заданого класу визначена заздалегідь. Прецедентом може бути, наприклад, набір ознак пацієнта із відомим діагнозом, з яким слід порівнювати набір ознак людини, діагноз якої ще невідомий. Кожний образ являє собою набір чисел, що

описують його властивості і називаються ознаками. Упорядкований набір ознак об'єкта називається вектором ознак.

Вектор ознак – це точка в просторі ознак. Класифікатор, або вирішальне правило – це функція, яка ставить у відповідність вектору ознак образу клас, до якого він належить. Задачу розпізнавання образів можна розділити на ряд підзадач:

- генерування ознак (вимірювання або обчислення числових ознак, що характеризують об'єкт);

- вибір ознак (визначення найбільш інформативних ознак для класифікації, в цей набір можуть входити не лише первинні ознаки, але й функції від них);

- побудова класифікатора – конструювання вирішального правила, на підставі якого здійснюється класифікація;

- оцінка якості класифікації – обчислення показників правильності класифікації (точність, чутливість, специфічність, помилки першого та другого роду).

Проектування системи розпізнавання образів є важливим завданням у межах розвитку як ІТ-галузі, так і промисловості загалом, оскільки немає жодного підприємства чи суб'єкта господарювання, в арсеналі якого не застосовується обладнання, що сприяє необхідності розвитку відповідного математико-алгоритмічного апарату для його реалізації. Внаслідок неординарності поставленої задачі немає загального її розв'язання для різних типів зображень, тому з певними допущеннями можна застосовувати як методи розпізнавання зображень, так і технології штучного інтелекту. Побудова нових методів опрацювання сприятиме розв'язанню задачі розпізнавання та забезпечить істотне здешевлення обладнання.

Метод, запропонований Віолою і Джонсом в 2001 році, став справжнім проривом в області розпізнавання образів. Цей метод набув великої популярності завдяки високій точності і серйозній теоретичній основі [1], [2]. Основні принципи, на яких базується метод:

- використовуються зображення в інтегральному уявленні, що дозволяє швидко обчислювати необхідні об'єкти;
- використовуються ознаки Хаара, за допомогою яких відбувається пошук потрібного об'єкта (в цьому контексті, овочів і фруктів);
- використовується “boosting” для вибору найбільш підходящих ознак для шуканого об'єкта на певній частині зображення;
- всі ознаки надходять на вхід класифікатора, який дає результат «так» або «ні»;
- використовуються каскади ознак для швидкого відкидання ділянок зображень, де не знайдено певний фрукт або овоч.

Найбільш популярний цей метод для пошуку об'єктів у відео потоці. Також цей детектор має вкрай низьку ймовірність помилкового виявлення об'єкту.

Для того, щоб проводити будь-які дії з даними в методі Віоли-Джонса, використовується інтегральне представлення зображень. Однією з корисних особливостей інтегрального представлення є можливість дуже швидко вирахувати суму пікселів довільного прямокутника (або будь-який інший фігури, яку можна апроксимувати декількома прямокутниками).

Інтегральне представлення зображення – це матриця, що збігається за розмірами з вихідним зображенням. У кожному елементі її зберігається сума інтенсивності яскравості всіх пікселів, що знаходяться лівіше і вище даного елемента. Елементи матриці розраховуються за формулою:

$$L(x, y) = \sum_{i=0, j=0}^{i \leq x, j \leq y} I(i, j). \quad (1.1)$$

де  $I(i, j)$  – яскравість пікселя вихідного зображення. Кожен елемент матриці  $L(x, y)$  є сумою пікселів в прямокутнику від  $(0,0)$  до  $(x,y)$ , тобто значення кожного пікселя  $(x, y)$  дорівнює сумі значень всіх пікселів лівіше і вище даного пікселя  $(x, y)$ . Розрахунок матриці займає лінійний час, пропорційне числу пікселів в зображенні, тому інтегральне зображення прораховується за один прохід.

Розрахунок матриці можливий за формулою:

$$L(x, y) = I(x, y) - L(x - 1, y - 1) + L(x, y - 1) + L(x - 1, y). \quad (1.2)$$

Один з важливих моментів в методі Віола-Джонса — це використання каскадної моделі або виродженого дерева прийняття рішень: в кожному вузлі дерева за допомогою каскаду приймається рішення чи може на зображенні містяться об'єкт чи ні. Якщо об'єкт не міститься, то алгоритм закінчує свою роботу, якщо він може містяться, то ми переходимо до наступного вузла. Навчання побудовано таким чином, щоб на початкових рівнях з найменшими витратами відкидати велику частину областей зображення, в яких не може міститися об'єкт.

Недоліком методу є те, що для виявлення об'єктів різних типів нам необхідно зберігати велику базу еталонних об'єктів в різних положеннях, масштабах і, можливо навіть для різних рівнів чутливості. При додаванні в базу еталонів нового виявленого об'єкту, необхідно занесення його «копій» у всіх допустимих масштабах і поворотах, що істотно збільшує витрати пам'яті, а також часу виявлення пропорційно кількості об'єктів, що вводяться.

Додаткові обмеження застосування привносить те, що така зміна форми об'єкта як, наприклад, масштабування або перекривання, робить даний метод непридатним для виявлення більшості об'єктів в реальних ситуаціях з великим рівнем невизначеності.

Інтегральне представлення зображення використовується для обчислення ознак Хаара. Ознаки Хаара — ознаки цифрового зображення, використовувані в розпізнаванні образів. Своєю назву вони зобов'язані інтуїтивним схожістю з вейвлетами Хаара. Ознаки Хаара використовуються в детекторі фруктів та овочів, що працює в режимі розпізнавання зі зробленої фотографії продуктового кошика. Ознака Хаара складається з суміжних прямокутних областей. Вони позиціонуються на зображенні, далі підсумовується інтенсивності пікселів в областях, після чого обчислюється

різниця між сумами. Ця різниця і буде значенням певної ознаки, визначеного розміру, певним чином розташованій на зображенні.

Перетворення Хаара використовується для стиснення вхідних сигналів, компресії зображень, здебільшого кольорових і чорно-білих з плавними переходами. Даний вид архівації відомий досить давно і безпосередньо виходить з ідеї використання когерентності областей. Ступінь стиснення задається і варіюється в межах 5-100. При спробі задати більший коефіцієнт на різких межах, особливо проходять по діагоналі, проявляється «ефект сходів» - сходинки різної яскравості розміром в кілька пікселів.

Вейвлет-перетворення широко використовується для аналізу нестационарних процесів. Воно показало свою ефективність для вирішення широкого класу задач, пов'язаних з обробкою зображення. Коефіцієнти вейвлет-перетворення містять інформацію про аналізованому процесі і використовуюваному вейвлет. Тому вибір аналізує вейвлета визначається тим, яку інформацію необхідно витягти з процесу. Кожен вейвлет має характерні особливості в тимчасовій і частотній областях, тому іноді за допомогою різних вейвлетів можна повніше виявити і підкреслити ті чи інші властивості аналізованого процесу. У даній роботі використовуються вейвлет-перетворення Хаара.

Процес аналізу зображення з використанням ознак Хаара методом Віоли-Джонса представлений на рисунку 1.1. Детектор аналізує області зображення, розраховує матрицю та порівнює інтенсивність пікселів в областях з ознаками.

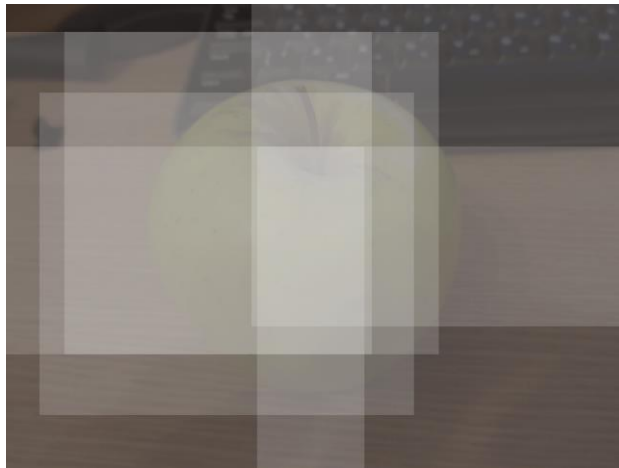


Рисунок 1.1 – Процес аналізу областей зображення

## 1.2 Методи сегментації зображення за кольором фруктів або овочів

### 1.2.1 Визначення границь кластеру

Одним з методів побудови класифікатора кольору шуканих об'єктів – це побудова границь кольорового кластеру за допомогою заздалегідь вибраних правил для кожного класу об'єктів.

Для прикладу можна привести зелене яблуко, для нього використовувалися такі правила границь кольорового кластеру в системі RGB:

$$R \in [85, 205]; G \in [70, 210]; B \in [20, 100] \quad (1.3)$$

Слід сказати, що такі правила являються елементарними, тому класифікатор, що працюватиме з ними – буде аналізувати їх швидко та з мінімальними обчислювальними потребами. Але слід брати до уваги об'єкти, що підходять для таких правил, не будучи шуканими. У статті [3] було запропоновано підхід з використанням алгоритмів машинного вивчення, який сам знаходить відповідні правила для об'єкту, після цього зберігає їх, використовуючи для детектування, такий підхід веде до високого рівня детектування об'єктів різних класів.

У контексті алгоритму, є множина об'єктів (зображень), розділених деяким чином на класи. Визначена скінченна множина зображень, для яких відомо, до якого класу вони відносяться (наприклад, це може бути клас «яблуко»). Ця множина називається навчальною вибіркою. Класова приналежність інших об'єктів невідома. Потрібно побудувати алгоритм, здатний класифікувати довільний об'єкт з початкової множини.

Класифікувати об'єкт - це вказати номер (або найменування класу), до якого належить даний об'єкт. Класифікація об'єкта - номер або найменування класу, що видаються алгоритмом класифікації в результаті його застосування до даного конкретного об'єкту. Класифікатор (classifier) – в задачах класифікації це апроксимуюча функція, що виносить рішення, до якого саме класу даний об'єкт належить. Навчальна вибірка - скінченне число даних. У машинному навчанні завдання класифікації відноситься до розділу навчання з учителем, коли класи роздільні та не перетинаються. Розпізнавання образів по суті і є класифікація зображень і сигналів.

У випадку алгоритму Віюлі- Джонса для ідентифікації та розпізнавання фруктів або овочів класифікація є багатокласовою – об'єкт може бути яблуком, бананом чи капустою.

### **1.2.2 Моделювання кольору об'єктів непараметричними методами**

Ідея непараметричних методів моделювання кольору фруктів або овочів – визначення правил розподілу кольорів об'єктів з використанням тренувальних даних, але без наявності явної моделі кольору фрукта або овочу. Такі методів використовують принципи теорії імовірності для розрахунку значень для кожної точки кольорового простору. Декілька методів з описаних у використаних ресурсах реалізували гістограмний метод для розділення зображення на сегменти. Кольоровий простір розбивається на деяку кількість біт, відповідних за деяку смугу значень кольорових компонент пар (2D) або тріад (3D). Ці біти формують гістограми. Кожен біт

містить інформацію про кількість випадків коли цей колір зустрічався в тренувальному процесі обробки кольорового зображення. Після тренування вміст гістограми нормалізується перетворюючи значення гістограм до дискретного розподілу ймовірностей.

### 1.3 Штучна нейронна мережа

Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних різним чином шарів нейронів, які організовують загальну активну структуру і функціонально впливають на роботу всієї системи. У більшості архітектур ШНМ активність нейрона визначається сумою впливу інших нейронів на даний нейрон [6]).

З моменту свого зародження технології штучних нейронних мереж розвивалися досить відокремлено від класичних методів шляхом, нерідко докорінно змінюючи уявлення про предмет в сукупної проблематики теорій машинного навчання і розпізнавання образів. З цього часу в науковому спільноті відбулося кілька спадів і підйомів зацікавленості до цього напрямку, але, завдяки деяким проривам в теорії ШНМ, широке практичне застосування нейромережеві технології отримали порівняно недавно.

У більшості архітектур ШНМ передавальні функції активації нейронів фіксовані, а ваги є параметрами мережі. Деякі входи нейронів є зовнішніми входами сукупної мережі, а деякі виходи нейронів - виходами сукупної мережі. Робота нейромережі полягає в перетворенні вхідного вектору у вихідний вектор, що здійснюється вагами і топологією мережі. Існують різні класифікації ШНМ за рядом ознак. По одному з топологічних ознак ШНМ можна класифікувати як:

- повно зв'язні ШНМ – кожен нейрон пов'язаний з іншими нейронами в мережі, включаючи себе самого;
- шаруваті ШНМ – нейрони об'єднуються в шари, нейрони попереднього шару пов'язані з нейронами наступного шару;



– слабо зв'язані ШНМ – нейрони знаходяться в вузлах прямокутної або гексогональної решітки.

Відмінності обчислювальних процесів в нейронних мережах часто обумовлені способом взаємозв'язків нейронів. За сукупністю критеріїв на сьогоднішній день багатопов'язані архітектури ШНМ можна розділити на статичні і динамічні. Кожен з класів архітектури нейронних мереж може включати безліч підкласів, реалізуючи різні підходи, нижче будуть наведені основні з них.

До статичних архітектур відносять мережі прямого поширення, в яких реалізован односторонній зв'язок між шарами, відсутні динамічні елементи і зворотний зв'язок, а вихід навченої ШНМ однозначно визначається входом і не залежить від попередніх станів мережі [7]).

Статичні ШНМ прямого поширення:

- Перцептрон;
- нейронна мережа Кохонена;
- Когнітрон;
- Неокогнітрон.

На противагу статичним архітектурам, існують динамічні архітектури ШНМ, що реалізують рекурентну структуру з використанням обернених зв'язків, завдяки чому стан мережі в кожен момент часу залежить від попереднього стану. Рекурентні ШНМ як правило базуються на перцептроні. Динамічні рекурентні ШНМ зі зворотними зв'язками:

- нейронна мережа Хопфілда;
- нейронна мережа Коско;
- нейронна мережа Джордана;
- нейронна мережа Елмана.

Найважливішою властивістю нейронних мереж є їх здатність навчатися на основі даних навколишнього середовища і в результаті навчання підвищувати свою продуктивність. Підвищення продуктивності відбувається з часом у відповідності з певними правилами.

Навчання нейронної мережі відбувається за допомогою інтерактивного процесу корегування синаптичних ваг і порогів. В ідеальному випадку нейронна мережа отримує знання про навколишнє середовище на кожній ітерації процесу навчання. З поняттям навчання асоціюється досить багато видів діяльності, тому складно дати цьому процесу однозначне визначення. Більше того, процес навчання залежить від точки зору на нього. Саме це робить практично неможливим появу будь-якого точного визначення цього поняття. Наприклад, процес навчання з точки зору психолога в корені відрізняється від навчання з точки зору шкільного вчителя.

З позицій нейронної мережі, ймовірно, можна використовувати наступне визначення: Навчання – це процес, у якому вільні параметри нейронної мережі настроюються за допомогою моделювання середовища, у яке ця мережа вбудована.

Навчання нейронної мережі з учителем припускає, що для кожного вхідного вектору з навчальної множини існує необхідне значення вихідного вектору, так званого цільовим. Ці вектору утворюють навчальну пару. Ваги мережі змінюють до тих пір, поки для кожного вхідного вектору не буде отриманий прийнятний рівень відхилення вихідного вектору від цільового. Нейронна мережа має у своєму розпорядженні правильними відповідями (виходами мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді, як можна більш близькі до відомих правильних відповідей.

Посилений варіант навчання з учителем припускає, що відома тільки критична оцінка правильності виходу нейронної мережі, але не самі правильні значення виходу. Навчання нейронної мережі без вчителя є набагато більш правдоподібною моделлю навчання з точки зору біологічних коренів штучних нейронних мереж. Навчальна множина складається лише з вхідних векторів. Алгоритм навчання нейронної мережі підлаштовує ваги мережі так, щоб виходили узгоджені вихідні вектори, тобто щоб пред'явлення досить близьких вхідних векторів давало однакові виходи.

Навчання без вчителя не вимагає знання правильних відповідей на кожний приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних, або кореляції між зразками в системі даних, що дозволяє розподілити зразки за категоріями. При змішаному навчанні частина ваг визначається за допомогою навчання з учителем, у той час як інша виходить за допомогою самонавчання.

#### **1.4 Перцептрон**

Елементарний перцептрон організовується на основі сенсорних даних на вході – S-елементів, асоціативних елементів – A – елементів, і реагують елементів на виході – R – елементів. Набір S – елементів, пов'язаний з A – елементом утворює асоціацію, і A елемент активізується після того як досягнуто певне число сигналів від S-елементів. A – елемент передає зважений сигнал на фінальний R – елемент, і в залежності від того, чи перевищує зважена сума деякий поріг, R – елемент видає результат роботи перцептрону, таку систему можна побачити на рисунку 1.2.

Багатошаровий перцептрон організовується з додатковими прихованими шарами A-елементів, розташованими між S-елементами і R-елементами. Принципова складність завдань, що вирішуються багатошаровим перцептроном, є найвищою для класу перцептронів. Навчання елементарного і багатошарового перцептрона полягає в зміні вагових коефіцієнт зв'язків A – R. Перцептрон здатний працювати в режимі розпізнавання або узагальнення

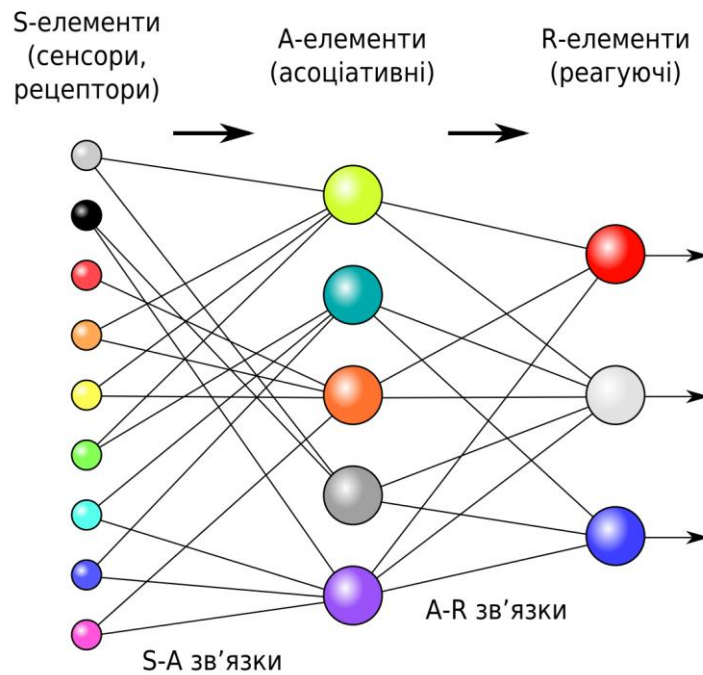


Рисунок 1.2 – Перцептрон

### 1.5 Штучний нейрон

Прототипом для створення нейрона став біологічний нейрон головного мозку. Біологічний нейрон має тіло, сукупність відростків – дендритів, за якими в нейрон надходять вхідні сигнали, і аксонів, що передають вихідні сигнали нейронів іншим клітинам. Точка з'єднання дендрита і аксона називається синапсом [7].

Спрощено функціонування нейрона можна представити наступним чином: Нейрон отримує від дендритів набір (вектор) вхідних сигналів; У тілі нейрона оцінюється сумарне значення вхідних сигналів. Однак входи нейрона нерівнозначні. Кожен вхід характеризується деяким ваговим коефіцієнтом, що визначає важливість інформації переданої ним. Таким чином, нейрон не просто підсумовує значення вхідних сигналів, а обчислює скалярний добуток вектору вхідних сигналів і вектору вагових коефіцієнтів;

Нейрон формує вихідний сигнал, інтенсивність якого залежить від значення обчисленого скалярного перемноження. Якщо воно не перевищує деякого заданого порогу, то вихідний сигнал не формується зовсім – нейрон

«не спрацьовує»; Вихідний сигнал надходить на аксон і передається дендриту інших нейронів. Нейрон – це складова частина нейронної мережі. Структуру штучного нейрона зображено на рисунку 1.3.

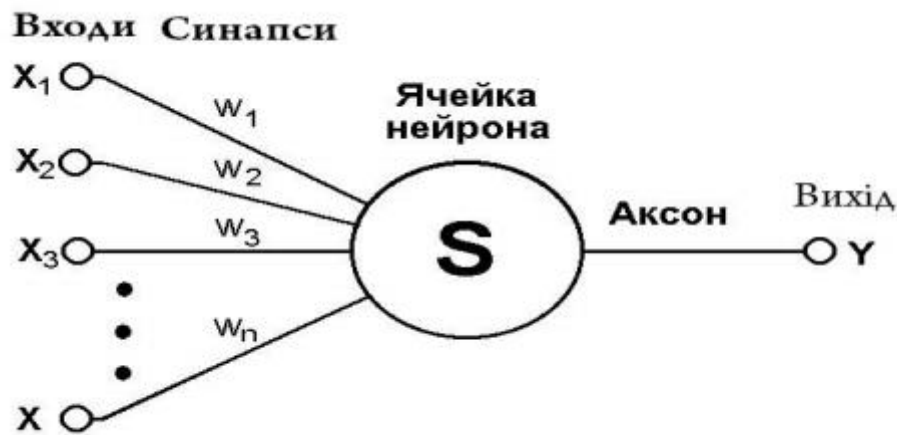


Рисунок 1.3 – Структура штучного нейрона

Не менш важливим є завдання побудова мережі. Це питання вирішується в два етапи:

- вибір типу (архітектури) мережі;
- підбір ваг (навчання) мережі.

На першому етапі слід вирішити наступні питання:

- які нейрони ми хочемо використовувати (число входів, передавальні функції);
- яким чином слід з'єднати їх між собою;
- що взяти в якості входів і виходів мережі.

Це завдання на перший погляд здається складним, але необов'язково придумувати нейромережу – існує кілька десятків різних нейромережових архітектур, причому ефективність багатьох з них доведена математичною статистикою. Найбільш популярні і вивчені архітектури – це багат шаровий перцептрон, нейромережа із загальною регресією, мережі Кохонена, мережі Хопфілда, мережі Хеммінга та інші. На другому етапі слід навчити обрану мережу, тобто підібрати такі значення її ваг, щоб мережа працювала

належним чином. У використовуваних на практиці нейромережах кількість ваг може становити кілька десятків тисяч, тому навчання це дійсно складний процес.

Для багатьох архітектур розроблені спеціальні алгоритми навчання, які дозволяють налаштувати ваги мережі певним чином. У залежності від функцій, виконуваних нейронами в мережі, можна виділити три їх типи: - вхідні нейрони – це нейрони, на які подається вхідний вектор, що кодує вхідний вплив чи образ зовнішнього середовища; в них зазвичай не здійснюється обчислювальні процедури, інформація передається з входу на вихід нейрона шляхом трансформаційних змін його активації; - вихідні нейрони – це нейрони, вихідні значення яких представляють вихід мережі; - проміжні нейрони – ці нейрони складають основу штучних нейронних мереж.

У більшості нейронних моделей тип нейрона пов'язаний з його розташуванням у мережі. Якщо нейрон має тільки вихідні зв'язки, то це вхідний нейрон, якщо навпаки – вихідний нейрон. Однак може зустрітися випадок, коли вихід внутрішнього нейрона розглядається як частина виходу мережі. У процесі функціонування мережі здійснюється перетворення вхідного вектору у вихідний.

## 1.6 Методи дистантного перетворення

Припустимо, що необхідно виявити об'єкт  $S_0(x, y, i)$  з відомим контуром  $B_0(i)$  і невідомими координатами точки прив'язки  $Ap_0(x, y, i)$  на двовимірному сигналі  $S_0(x, y)$ . Априорі відомо безліч еталонних об'єктів  $S_e(x_e, y_e, j)$  з відомими кордонами  $B_e(j)$  і малюнками інтенсивностей.

Уявімо сигнал скануючого пристрою у вигляді упорядкованого в тимчасовому і просторовому інтервалі двовимірної множини значень

$\{S_s(x, y)\}$ , в якому можна виділити кілька підмножин значень: безліч значень інтенсивності об'єкта і фону.

У разі апіорної невизначеності значення точки прив'язки об'єкта, рішення щодо його виявлення приймається за допомогою аналізу міри подібності еталонного об'єкта  $S_e(x_e, y_e, j)$  і аналізованого об'єкта  $S_0(x, y, i)$ .

Міра подібності двох об'єктів в найпростішому випадку може бути середньоквадратичною помилкою

$$Dis(i, j) = \max\left\{ \sum_{\forall(x,y) \in (B_e \cap B_x)} [S_e(x, y, j) - S_0(x, y, i)]^2 \right\} \quad (0.1)$$

Точний збіг еталона з якою-небудь частиною малюнка інтенсивностей в реальній ситуації практично неможливо, внаслідок наявності шуму вихідного сигналу, а також, внаслідок наявності апіорної невизначеності щодо точної форми, розмірів і структури локалізованого об'єкта.

У такому випадку висновок про присутність шуканого об'єкта робиться на підставі перевищення наведених вище заходів подібності деякого порога виявлення  $S_d$ , який зазвичай вибирається однаковим для всіх точок.

## 1.7 Постановка задачі

Кількість вирішених задач з розпізнавання образів росте з кожним днем, але, незважаючи на це – їх недостатньо для покриття всіх потреб в таких задачах.

Метою даної дипломної роботи є розробка моделі та програмного комплексу для розпізнавання об'єктів у продуктовому кошику (фруктів та овочів), шляхом використання сегментації зображень за кольором, ознак Хаара, методу Віоли-Джонса. Також буде використовуватися тренувальні дані за метою отримання певної кількості ознак. Для цього необхідно виконати наступні завдання:

- дослідити методи сегментації зображень за ознаками;
- дослідити методи опису об'єктів за допомогою ознак;
- розбити тренувальні зображення на позитивні та негативні;
- використати бібліотеку для тренування аналізатору;
- програмно використати алгоритм Віоли-Джонса;
- розробити й програмно реалізувати алгоритм роботи Android додатку для отримання вхідних зображень з камери.

Для демонстрації роботи розробленого комплексу були вибрані тестові фотографії, які містять овочі та фрукти для розпізнавання. Обрані кольорові зображення мають довільний розмір в діапазоні від 200x200 пікселів, і до 2050x2050 пікселів. Приклади тестових зображень та зображень для тренування наведено в додатку А.



## 2 РЕАЛІЗАЦІЯ МОДЕЛІ МЕТОДУ ВІОЛІ-ДЖОНСА

### 2.1 Інтегральне представлення в методі Віолі-Джонса

Однією з корисних особливостей інтегрального представлення в методі Віолі-Джонса є можливість дуже швидко вичислити суму пікселів довільного прямокутника [5]. Як задувалося вище – елементи інтегральної матриці розраховуються по формулі (1.1).

По такій інтегральній матриці можна дуже швидко вичислити суму пікселів довільного прямокутника, довільної площі. На етапі виявлення в методі Віолі-Джонса вікно встановленого розміру рухається по зображенню, і для кожної області зображення, над якою проходить вікно, розраховується ознака Хаару. Наявність або відсутність предмета у вікні визначається різницею між значенням ознаки і навчаним порогом, у нашому модернізованому методі Віолі-Джонса, використовуються основні ознаки предмету, вони знаходяться на рисунку 2.1 та додаткові ознаки, які представлені на рисунку 2.1.

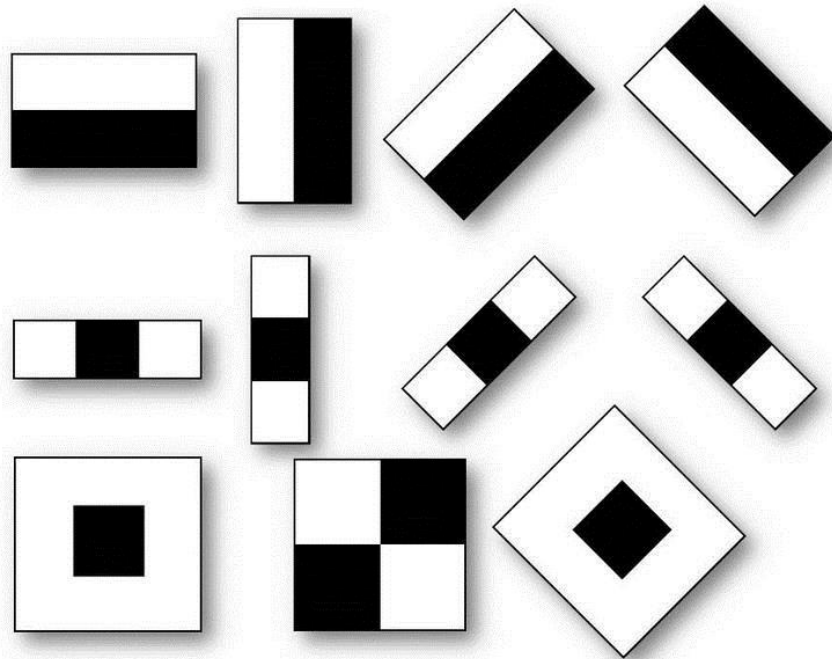


Рисунок 2.1 – Основні ознаки Хаару

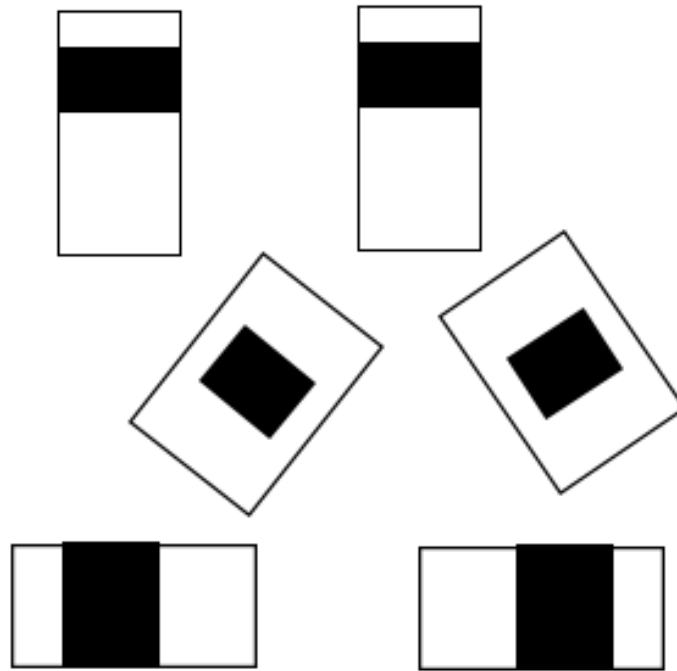


Рисунок 2.2 – Додаткові ознаки Хаару

Обчислюваним значенням такої ознаки буде:

$$F = X - Y, \quad (2.1)$$

де  $X$  - сума значень яркостей точок що закриваються світлою частиною ознаки;  $Y$  - сума значень яркостей точок що закриваються темною частиною ознаки.

В процесі пошуку обчислювати усі ознаки просто нереально, отже, класифікатор повинен реагувати тільки на певну, потрібну підмножину усіх ознак. Потрібно навчити класифікатор знаходженню пішохода на дорозі по цій певній підмножині, це можна зробити, навчаючи алгоритм автоматично [4]) для вирішення проблеми такого складного навчання існує технологія бустинг.

Математично бустинг пояснюється так: спочатку обчислюються оцінки приналежності об'єкту класам, потім вирішальне правило переводить ці оцінки в номер класу. Значення оцінки, як правило, характеризує міру упевненості класифікації. Алгоритмічна композиція алгоритму  $X \rightarrow Y$ :

$$a = C(F(b1(x), \dots, bT(x)), x \in X. \quad (2.2)$$

де складений з алгоритмічних операторів

$$bt : X \rightarrow R, t = 1, \dots, T. \quad (2.3)$$

операції  $F$ , що коригує:

$$F : RT \rightarrow R. \quad (2.4)$$

і вирішального правила

$$C : R \rightarrow Y. \quad (2.5)$$

У завданнях класифікації на два класи, що не перетинаються, як простір оцінок зазвичай використовується безліч дійсних чисел. Вирішальні правила можуть мати параметри, що настроюються. Так, в алгоритмі Віюлі-Джонса використовується порогове вирішальне правило, де, як правило, спочатку будується оператор при нульовому значенні, а потім підбирається значення оптимальне.

Бустинг може призводити до побудови громіздких композицій, що складаються з сотень алгоритмів. Підхід посилення простих класифікаторів є найбільш ефективним методом класифікації за рахунок високої швидкості і ефективності роботи і відносної простоти реалізації.

При об'єднанні слабких анізотропних класифікаторів в один великий сильний класифікатор детектувальна здатність підвищується, але при цьому підвищується і час ухвалення рішення.

В деяких випадках час, окрім якості розпізнавання, є ключовим чинником. Для цього випадку пропонується побудувати каскадну модель з невеликих сильних класифікаторів, що дозволяє підвищити швидкість детектування, не погіршуючи якість [6]. Каскадна модель сильних

класифікаторів - це дерево ухвалення рішень, де кожен вузол дерева побудований так, щоб детектувати майже усі образи, що цікавлять, і відхиляти деяку частину регіонів, що не є образами.

Окрім цього, вузли дерева розміщені таким чином, що чим ближче вузол знаходиться до кореня дерева, тим з меншої кількості анізотропних примітивів гаусів він полягає і тим самим вимагає меншого часу на ухвалення рішення.

Перцептивні хеш-функції обчислюють схожі хеш-значення для схожих зображень. В алгоритмі використовується функція порівняння, відстань Хемінга. Для полегшення порівняння відстань Хемінга можна нормувати за допомогою довжини векторів:

$$\Delta(x, y) = \frac{1}{n} \sum_{x_i \neq y_i} 1, i = 1, \dots, n. \quad (2.6)$$

Відстань Хеммінга визначимо як

$$\delta(x, y) \leq n. \quad (2.7)$$

Ідея м'якого виходу полягає в нечіткому результаті, на основі аналізу якого можна прийняти інше рішення. Розглядатимемо відношення

$$H_S(x) = \frac{2 \sum_{t=1}^T a^{(t)} h_j^{(t)}}{\sum_{t=1}^T a^{(t)}}. \quad (2.8)$$

як самостійний числовий критерій, який називатимемо м'який вихід.

Одним з можливих застосувань м'якого виходячи являється ухвалення однозначного рішення про знайдений об'єкт, виходячи зі значення  $H_s(x)$ . Проте застосування такого критерію прямо має ряд недоліків.

Видно, що знайдений алгоритмом об'єкт в реальності не має найбільшого значення м'якого виходу. У завданні кластеризації даних можна враховувати додаткову інформацію, якою є значення м'якого виходу. Крім того, використовуючи цю інформацію можна штучно понизити поріг розпізнавання і враховувати отримані регіони на етапі кластеризації даних.

При зниженні порогу значно збільшується кількість знайдених регіонів, ідея запропонованого методу полягає в перевірці гіпотези: при зменшенні порогу спрацьовування алгоритму розпізнавання і застосовуючи алгоритм кластеризації об'єктів можливо поліпшити ефективність розпізнавання.

При цьому слід зауважити, що параметри в алгоритмі кластеризації підбираються емпірично. Алгоритм AdaBoost (скор. Від adaptive boosting) – алгоритм машинного навчання, запропонований Йоав Фройнд (Yoav Freund) і Робертом Шапіро (Robert Schapire). Являє собою мета-алгоритм, в процесі навчання будує композицію з базових алгоритмів навчання для поліпшення їх ефективності.

AdaBoost є алгоритмом адаптивного бустінга в тому сенсі, що кожен наступний класифікатор будується по об'єктах, які погано класифікуються попередніми класифікаторами.

AdaBoost викликає слабкий класифікатор в циклі. Після кожного виклику оновлюється розподіл ваг, які відповідають важливості кожного з об'єктів навчальної вибірки для класифікації. На кожній ітерації ваги кожного невірно класифікованого об'єкта зростають, таким чином новий класифікатор «фокусує свою увагу» на цих об'єктах.

Переваги алгоритму можна записати наступним списком:

– добра узагальнююча здатність. В реальних задачах вдається будувати композиції, що перевершують за якістю базові алгоритми. Узагальнююча здатність може поліпшуватися у міру збільшення числа базових алгоритмів;

- простота реалізації;
- власні накладні витрати бустінга невеликі. Час побудови композиції практично повністю визначається часом навчання базових алгоритмів;
- Можливість ідентифікувати об'єкти, які є шумовими викидами.

Недоліками ж є:

- AdaBoost схильний до перенавчання при наявності значного рівня шуму в даних. Експоненціальна функція втрат занадто сильно збільшує ваги найбільш важких об'єктів, на яких помиляються багато базові алгоритми.

Однак саме ці об'єкти найчастіше виявляються шумовими викидами. В результаті AdaBoost починає налаштовуватися на шум, що веде до перенавчання. Проблема вирішується шляхом видалення викидів або застосування менш агресивних функцій втрат;

- AdaBoost вимагає досить довгих навчальних вибірок. Інші методи лінійної корекції, зокрема, беггінг, здатні будувати алгоритми порівнянної якості за меншими вибірках даних;

– жадібна стратегія послідовного додавання призводить до побудови неоптимального набору базових алгоритмів. Для поліпшення композиції можна періодично повертатися до раніше побудованим алгоритмам і навчати їх заново. Для поліпшення коефіцієнтів можна оптимізувати їх ще раз після закінчення процесу бустінга за допомогою якого-небудь стандартного методу побудови лінійної розділяє поверхні. Рекомендується використовувати для цієї мети SVM (машини опорних векторів).

## **2.2 Обґрунтування вибору середовища програмної реалізації**

Існує кілька найпоширеніших операційних систем для смартфонів, такі як, IOS, Android, WindowsPhone, BlackBerry, Symbian. В даній роботі програмний комплекс буде створено для операційної системи Android, тому що вона, на нашу думку, має ряд переваг перед іншими ОС для смартфонів:

- Android – операційна система з відкритим програмним кодом, що дозволяє нам отримати актуальну інформацію про будь-які внутрішні функції та API;

- це найпоширеніша система серед користувачів мобільних пристроїв;

- більшість додатків для Android розробляються за допомогою мови програмування Java, яка на даний момент є найпоширенішою у світі;

- безкоштовні IDE для використання як в учбових цілях, так і в комерційних.

Для створення програми на ОС Android, можуть бути обрані різні середовища розробки, такі як, Eclipse, Embarcadero, JBuilder, JDeveloper і т.д, але для роботи обрана AndroidStudio від компанії Google. Причиною вибору стала зручність інтерфейсу, швидкодія програми, добре підготовлена документація від авторів, велика кількість шаблонів та бібліотек, інтегрована система контролю версій та здатність тестувати розроблений комплекс на емуляторі, або встановити його на реальний смартфон.

Android Studio - це інтегроване середовище розробки (IDE) для роботи з платформою Android, анонсована 16 травня 2013-го року на конференції Google. Ця IDE перебувала у вільному доступі починаючи з першої нестабільної версії, опублікованій в травні 2013-го, а потім перейшла в стадію бета-тестування у червні 2014-го. Перша стабільна версія побачила світ у грудні 2014-го року, тоді ж припинилася підтримка плагіна Android Development Tools (ADT) для Eclipse. AndroidStudio, заснована на програмному забезпеченні IntelliJ IDEA від компанії JetBrains, зараз ця IDE є офіційним засобом розробки Android додатків. Дане середовище розробки доступна для Windows, OS X і Linux.

## **2.2 Базова інформація про мову програмування Java**

Java - об'єктно-орієнтована мова програмування, яку поставляє компанія Oracle. Програми на Java транслюються в байт-код, що виконується

віртуальною машиною Java (JVM) - програмою, що обробляє байтовий код і передає інструкції пристрою як інтерпретатор. Перевагою подібного способу виконання програм є повна незалежність байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки, в рамках якої виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером), викликають негайне переривання.

Часто до недоліків концепції віртуальної машини відносять зниження продуктивності. Ряд удосконалень кілька збільшив швидкість виконання програм на Java: застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) з можливістю збереження версій класу в машинному коді, широке використання переносних орієнтованого коду (Native-код) в стандартних бібліотеках, апаратні засоби, що забезпечують прискорену обробку байт коду (наприклад, технологія Jazelle, підтримувана деякими процесорами фірми ARM). За даними інтернет ресурсів для семи різних завдань час виконання на Java становить в середньому в півтора-два рази більше, ніж для C / C ++, в деяких випадках Java швидше, а в окремих випадках в 7 разів повільніше. З іншого боку, для більшості з них споживання пам'яті Java- машиною було в 10–30 разів більше, ніж програмою на C / C ++.

OpenCV – бібліотека алгоритмів комп'ютерного зору, обробки зображень та чисельних алгоритмів загального призначення з відкритим кодом. Реалізована на C / C ++, також розробляється для Python, Java, Ruby, Matlab, Lua та інших мов. Може вільно використовуватися в академічних і комерційних цілях - поширюється в умовах ліцензії BSD.

Застосовується для затвердження загального стандартного інтерфейсу комп'ютерного зору для додатків в цій області. Для сприяння зростанню



числа таких додатків і створення нових моделей використання персонального комп'ютера.

Основні модулі бібліотеки:

- sxcogе–ядро містить базові структури даних і алгоритми;
- базові операції над багатовимірними числовими масивами;
- матрична алгебра, математичні функції, генератори випадкових чисел;
- запис та відновлення структур даних XML;
- базові функції 2D графіки.
- CV - модуль обробки зображень і комп'ютерного зору;
- базові операції над зображеннями (фільтрація, геометричні перетворення, перетворення колірних просторів і т. д.);
- аналіз зображень (вибір відмінних ознак, морфологія, пошук контурів, гістограми);
- спостереження за об'єктами;
- виявлення об'єктів, зокрема номерів;
- калібрування камер, елементи відновлення просторової структури.

Highgui - модуль для введення та виведення зображень і відео, створення призначеного для користувача інтерфейсу:

- захоплення відео з камер і з відео файлів, читання/запис статичних зображень;
- функції для організації простого UI (всі демо додатки використовують HighGUI).

Svauх - експериментальні та застарілі функції:

- просторовий. зір: стерео калібрації, саме калібрації;
- пошук стерео-відповідності, кліки в графах;
- знаходження і опис рис обличчя.

SvCam - захоплення відео:

– дозволяє здійснювати захоплення відео з цифрових відеокамер (підтримка припинена і в останніх версіях цей модуль відсутній).

Ця бібліотека була обрана для використання при розробці програмного продукту, так як задовольняє всім вимогам і задачам та значно спрощує розробку.

### **2.3 Тип бази даних для розробленого додатку**

База даних (БД) – упорядкований набір логічно взаємопов'язаних даних, що використовується спільно, та призначений для задоволення інформаційних потреб користувачів. У технічному розумінні включно й система управління БД. Система управління базами даних (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації. Централізований характер управління даними в базі даних передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі.

Головним завданням БД є гарантоване збереження значних обсягів інформації та надання доступу до неї користувачеві або ж прикладній програмі. Таким чином БД складається з двох частин: збереженої інформації та системи управління нею. З метою забезпечення ефективності доступу записи даних організовують як множину фактів (елемент даних).

Класифікація БД за моделлю даних:

- ієрархічні;
- мережеві;
- реляційні;
- об'єктні;
- об'єктно-орієнтовані;
- об'єктно-реляційні.

Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- кожен елемент таблиці – один елемент даних;
- всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символний тощо);
- кожен стовпчик має унікальне ім'я;
- однакові рядки в таблиці відсутні;
- порядок проходження рядків і стовпчиків може бути довільним.

Для зберігання інформації про середню кількість калорій для певного фрукту або овочу, мобільний додаток на платформі Android використовує базу даних SQLite.

SQLite – це компактна вбудована реляційна база даних. Вона не використовує парадигму клієнт-сервер, тобто не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якої програма компонується. Таким чином, в якості протоколу обміну використовуються виклики функцій (API) бібліотеки SQLite.

Такий підхід зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (Включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому пристрої, на якому виконується програма. Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, який зберігає базу даних – блокується на запис інформації. SQLite можливо використовувати як на вбудованих системах, так і на виділених машинах з гігабайтними масивами даних. Вона і була обрана для реалізації зберігання даних в мобільному додатку, як стандартне рішення, пропоноване і підтримуване платформою Android.

## **3 РОЗРОБКА АРХІТЕКТУРИ ТА РЕАЛІЗАЦІЯ КОМПЛЕКСУ**

### **3.1 Проектування програмного комплексу**

Перед безпосереднім початком програмної реалізації комплексу розпізнавання овочів та фруктів необхідно розробити архітектуру проекту, якої під час написання Java коду будемо притримуватись.

Слід сказати, що в нашій системі буде використовуватися об'єктно-орієнтований підхід програмування. З цього можна зробити висновок про основну одиницю комплексу – клас. Для початку спроектуємо структуру директорій в майбутній реалізації.

Основна директорію матиме назву `app` (`application` з англійської), вона буде містити усі класи, стилі, представлення. В ній ми і будемо розробляти усю логіку розпізнавання овочів і фруктів. Для цієї цілі використовується бібліотека `OpenCV`, яка представлена двома частинами: Java частина відповідає за API для додатків та виклику його з наших класів і C/C++ частина, так названі “JNI” бібліотеки. JNI бібліотеки було вирішено винести в сторонню директорію і збирати додаток разом з нею. В ній не буде зроблено ніяких змін після створення. Інші директорію необхідні для технічних цілей компіляції та розгортання нашого комплексу, про них ми згадувати не будемо, так як такі директорії є стандартними для всіх Android додатків та формуються середовищем розробки. Структура проекту представлена на рисунку 3.1. Вона відображається в спеціальному вікні `Android Studio`.

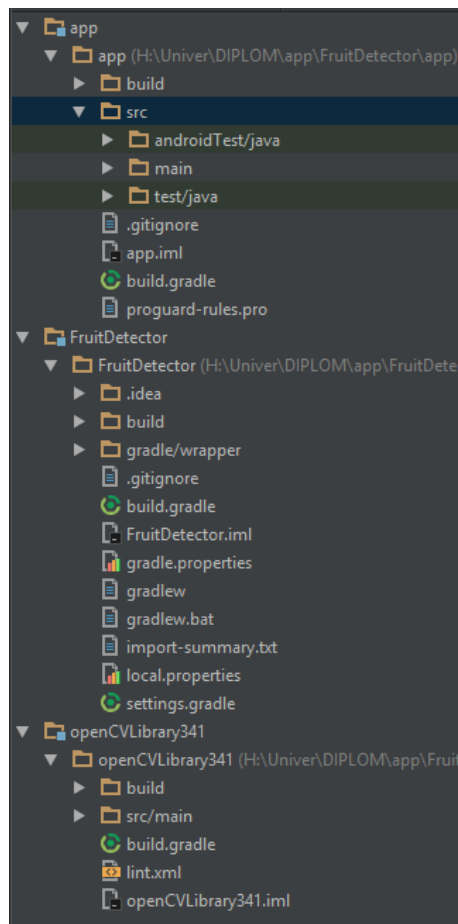


Рисунок 3.1 – Структура проекту в Android Studio

Наступним кроком необхідно розробити принципи використання методу Віоли-Джонса та каскади Хаара. Основною ідеєю використання OpenCV є:

– по-перше треба згенерувати каскади Хаара за допомогою бібліотеки Cascade-Trainer. Для кожного фрукту або овочу буде використовуватися свій каскад. На виході (після тренування системи) ми матимемо кілька файлів з розширенням xml, приклад таких файлів для банану та яблуку представлено на рисунку 3.2;

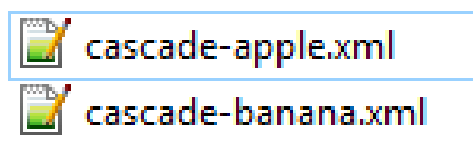


Рисунок 3.2 – Файли каскадів для яблуку і банана

– після формування повної структури необхідних каскадів додаємо всі файли до ресурсів комплексу, що розробляємо. Для обробки цих файлів буде використовуватись клас `FrameProcessor`. Основна його функція – пропускати отримане зображення з камери через кожен з каскадів, шукаючи при цьому необхідний об'єкт. Якщо його знайдено – повертати тип об'єкту (яблуко, банан, і т.д.) та область зображення, де він знайдений. Для зчитування файлів-каскадів створимо клас `CascadeUtils`.

Для легшої структуризації об'єктів будемо використовувати наступну архітектуру: створимо Java перерахування з назвою «Тип». Він буде містити назву об'єкту та файл-каскад, який відповідає назві. В такому випадку ми зможемо програмно редагувати об'єкти, які повинні бути знайдені.

Для відображення основного екрану додатку будемо використовувати клас `CameraView`. А для відображення окна результатів `ResultView`. Для кожного з них буде створено дизайн в зручному редакторі вікон `Android Studio`.

### **3.2 Activity і його стандартна реалізація**

Ключовим компонентом для створення візуального інтерфейсу в додатку `Android` є `activity` (активність). Нерідко `activity` асоціюється з окремим екраном або вікном додатка, а перемикання між вікнами буде відбуватися як переміщення від однієї `activity` до іншої. Додаток може мати одну або кілька `activity`. Приклад такої базової активності представлено нижче, на рисунку 3.3. Вона створюється при формуванні базового проекту

Всі об'єкти `activity` є об'єктами класу `android.app.Activity`, який містить базову функціональність для всіх `activity`. У нашому додатку для розпізнавання об'єктів в корзині супермаркету `MainActivity` представлено класом `CameraView`, що теж є похідним від класу `AppCompatActivity`. Однак сам клас `AppCompatActivity`, хоч і не прямо, успадковується від базового класу `Activity`.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
    }  
}
```

Рисунок 3.3 – Базовий код Activity

### 3.2 Життєвий цикл додатку на Android

Всі додатки Android мають строго певний системою життєвий цикл. При запуску користувачем система дає цьому додатку високий пріоритет. Кожна програма запускається у вигляді окремого процесу, що дозволяє системі давати одним процесам вищий пріоритет, на відміну від інших. Завдяки цьому, наприклад, при роботі з одними додатками не блокувати вхідні дзвінки, програвання музики чи отримання листа. Після припинення роботи з додатком, система звільняє всі пов'язані ресурси, переводить додаток у розряд низькопріоритетних і закриває його.

Всі об'єкти activity, які є в додатку, управляються системою у вигляді стека activity, який називається back stack. При запуску нової activity вона поміщається поверх стека і виводиться на екран пристрою, поки не з'явиться нова activity. Коли поточна activity закінчує свою роботу (наприклад, користувач йде з програми), то вона видаляється з стека, і відновлює роботу та activity, яка раніше була другою в стеці.

Після запуску activity проходить через ряд подій, які обробляються системою і для обробки яких існує ряд зворотних викликів, які представлені на рисунку 3.4

```

protected void onCreate(Bundle savedInstanceState) {}
protected void onStart() {}
protected void onRestoreInstanceState(Bundle savedInstanceState) {}
protected void onRestart() {}
protected void onResume() {}
protected void onPause() {}
protected void onSaveInstanceState(Bundle savedInstanceState) {}
protected void onStop() {}
protected void onDestroy() {}

```

Рисунок 3.4 – Події класу Activity

Схематично взаємозв'язок між усіма цими зворотними викликами можна представити рисунком 3.5.

Розберемо основні метода цього списку для розуміння принципу дії:

- onCreate – перший метод, з якого починається виконання activity. У цьому методі activity переходить в стан Created. Цей метод обов'язково повинен бути визначений в класі. У ньому проводиться первісне налаштування activity. Зокрема, створюються об'єкти візуального інтерфейсу. Цей метод отримує об'єкт Bundle, який містить попередній стан, якщо він був збережений. Якщо activity заново створюється, то даний об'єкт має значення null. Якщо ж activity вже раніше була створена, але перебувала в загальмованому стані, то bundle містить пов'язану з activity інформацію;

- після завершення методу onStart() викликається метод onRestoreInstanceState, який покликаний відновлювати збережений стан з об'єкта Bundle, який передається в якості параметра. Але слід враховувати, що цей метод викликається тільки тоді, коли Bundle НЕ дорівнює null і містить раніше збережений стан. Так, при першому запуску програми цей об'єкт Bundle матиме значення null, тому і метод onRestoreInstanceState не викликатиметься;

- а при виклику методу onResume activity переходить в стан Resumed, а користувач може з нею взаємодіяти. І власне activity залишається в цьому стані, поки вона не втратить фокус, наприклад, внаслідок перемикання на іншу activity або просто через виключення екрану пристрою;



– якщо користувач вирішить перейти до іншої activity, то система викликає метод `onPause`. У цьому методі можна звільняти використовувані ресурси, припиняти процеси, наприклад, відтворення аудіо, анімацій, зупиняти роботу камери (якщо вона використовується) і т.д., щоб вони менше позначалися на продуктивності системи.

Також слід зазначити, що при зміні орієнтації екрану система завершує activity і потім створює її заново, викликаючи метод `onCreate`.

В цілому перехід між станами activity можна виразити наступною схемою (Рис. 3.5):

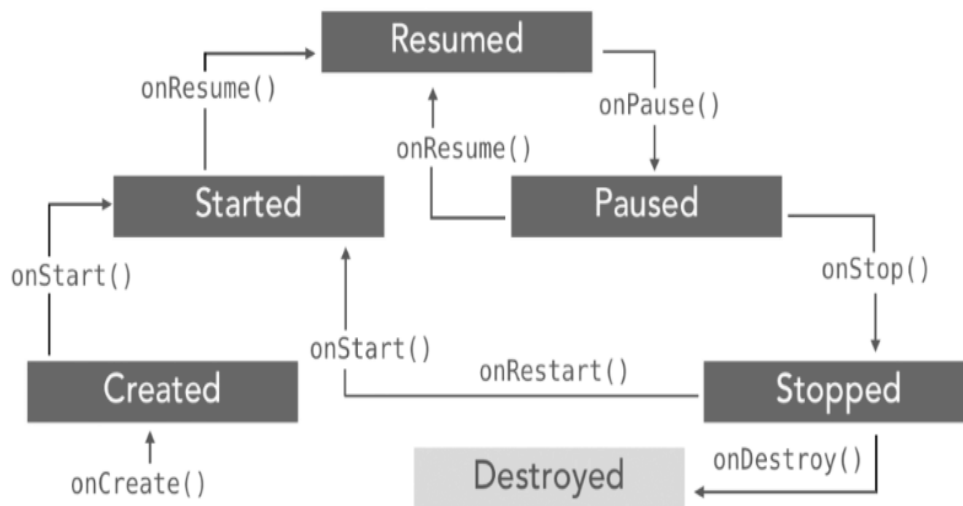


Рисунок 3.5 – Схема життєвого циклу Activity

### 3.3 Архітектура бази даних в контексті SQLite

SQLite — полегшена реляційна система керування базами даних. Втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92.

Особливістю SQLite є те, що вона не використовує парадигму клієнт-сервер, тобто рушій SQLite не є окремим процесом, з яким взаємодіє застосунок, а надає бібліотеку, з якою програма компілюється і рушій стає складовою частиною програми. Таким чином, як протокол обміну використовуються виклики функцій (API) бібліотеки SQLite. Такий підхід

зменшує накладні витрати, час відгуку і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, індекси і дані) в єдиному стандартному файлі на тому комп'ютері, на якому виконується застосунок.

Простота реалізації досягається за рахунок того, що перед початком виконання транзакції весь файл, що зберігає базу даних, блокується; ACID-функції досягаються зокрема за рахунок створення файлу-журналу.

Кілька процесів або потоків можуть одночасно без жодних проблем читати дані з однієї бази. Запис в базу можна здійснити тільки в тому випадку, коли жодних інших запитів у цей час не обслуговується; інакше спроба запису закінчується невдачею, і в програму повертається код помилки. Іншим варіантом розвитку подій є автоматичне повторення спроб запису протягом заданого інтервалу часу.

У комплекті постачання йде також функціональна клієнтська частина у вигляді виконуваного файлу `sqlite3`, за допомогою якого демонструється реалізація функцій основної бібліотеки. Клієнтська частина працює з командного рядка, і дозволяє звертатися до файлу БД на основі типових функцій ОС.

Завдяки архітектурі рушія можливо використовувати SQLite як на вбудовуваних (*embedded*) системах, так і на виділених машинах з гігабайтними масивами даних.

В роботі буде використовуватись саме ця база даних для зберігання інформації про калорійність того чи іншого продукту. Також в база даних буде зберігати назву, опис, шлях до файлу-каскаду. Модель даних представлена наступною схемою (Рис. 3.6):

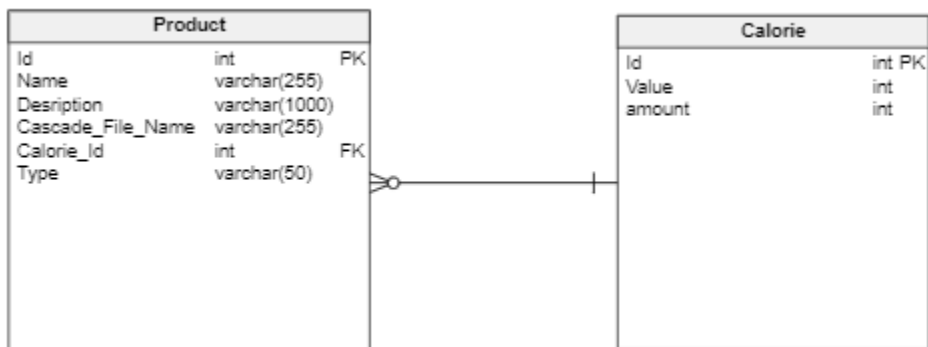


Рисунок 3.6 – Схема бази даних

Таблиця продукт містить необхідні дані про фрукт чи овоч (колонка «Type» дозволяє різнити їх між собою, та зовнішній ключ на таблицю калорій. Вона містить значення у калоріях (колонка «Value») та об'єм, до якого це значення припустиме (колонка «Amount»). Ця модель реалізована на стороні додатку Android та використовується у класі DatabaseHelper.

### 3.4 Тестування розробленого додатку

Спочатку проаналізуємо вихідний інтерфейс Android додатку. Головний екран повністю представляє собою область камери, з якої логіка розпізнавання буде брати інформацію, або вхідні зображення. Для початку аналізу вхідного зображення необхідно натиснути на прозору кнопку внизу екрану. Тільки після цього додаток почне аналіз та підрахунок калорійної цінності фруктів та овочів (Рис. 3.7).

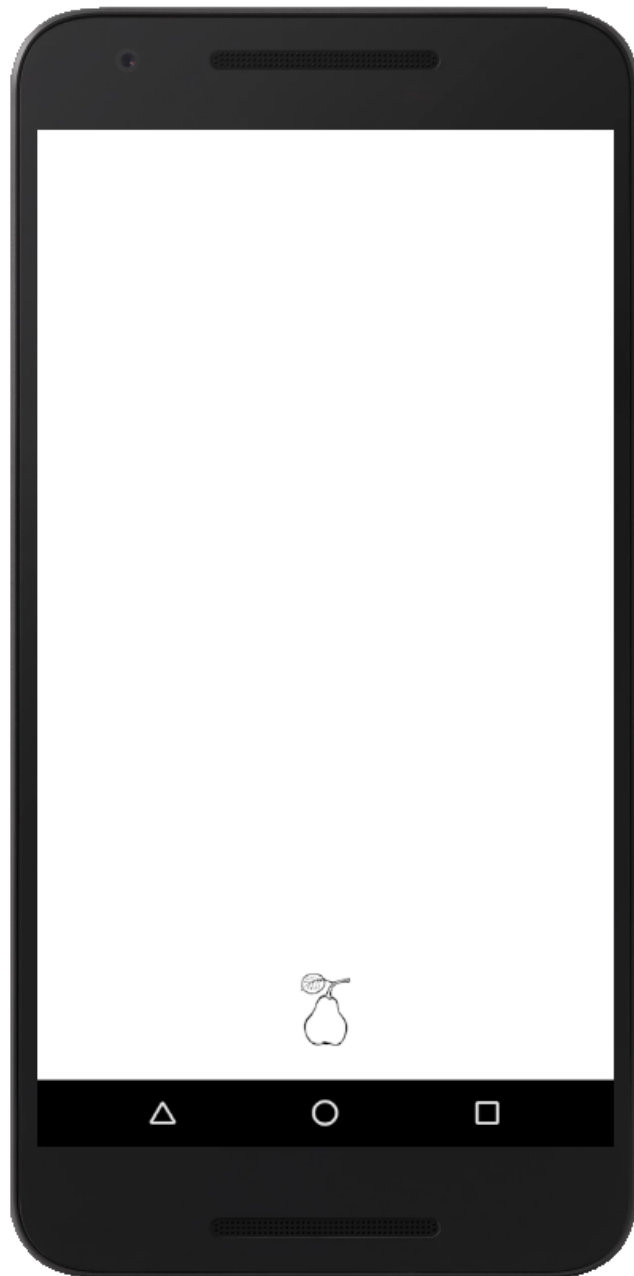


Рисунок 3.7 – Головне вікно розробленого додатку

Перш за все треба протестувати працездатність головного вікна та перехід до вікна результатів після натискання кнопки. При наведенні телефону на певну поверхню головне вікно повинно відобразити зображення з цією поверхнею (Рис. 3.8).



Рисунок 3.8 – Тестування працездатності модулю камери

Після натискання на кнопку з'являється вікно результатів без обведених прямокутників, що повідомляє нам про відсутність шуканих об'єктів на знімку (Рис. 3.9).



Рисунок 3.9 – Вікно результатів без фруктів та овочів

Тепер перевіримо процес роботи розпізнавання фрукту. Для тесту використаємо вхідне зображення тієї ж самої поверхні, але з яблуком. Додаток повинен розпізнати об'єкт, обвести його прямокутною областю та написати кількість калорій на сто грам продукту. Процес отримання вхідного зображення представлено на рисунку 3.10, а вікно отриманих результатів на рисунку 3.11.

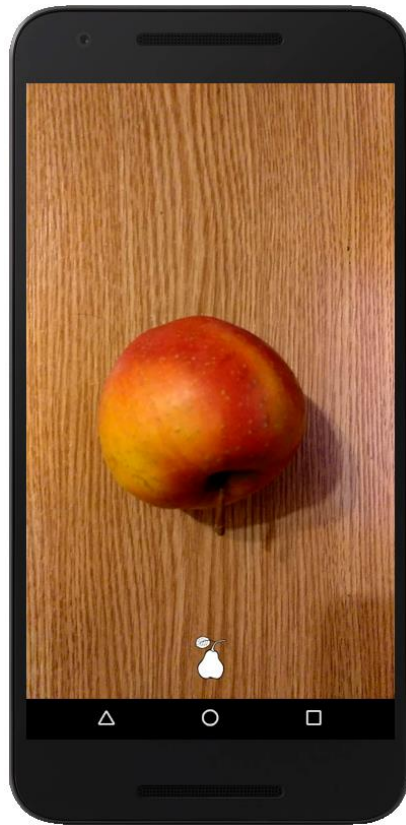


Рисунок 3.10 – Отримання вхідного зображення яблука



Рисунок 3.11 – Отримання результатів для яблука

Розглянемо роботу додатку на прикладі інших фруктів. Для тестування виберемо банан та інший фон зображення. Додаток так само повинен розпізнати об'єкт, обвести його прямокутною областю та написати кількість калорій на сто грам продукту. На Рисунку 3.12 представлено вихідне зображення, а на рисунку 3.13 маємо вікно з отриманими результатами.



Рисунок 3.12 – Отримання вхідного зображення банана





Рисунок 3.13 – Отримання результатів для банана

## 4 ОХОРОНА ПРАЦІ

### 4.1 Аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкту, що мають вплив на персонал

У даному дипломному проєкті розробляється програмне забезпечення.

Розроблене програмне забезпечення орієнтоване на роботу з персональним комп'ютером. Експлуатовані для вирішення внутрішньовиробничих завдань ПЕОМ типу IBM PC мають наступні характеристики:

споживана потужність	220 Вт;
робоча напруга	220 В;
напруга джерел живлення	+12 В; - 12 В; +5 В;
робоча частота	50 Гц.

Відповідно до [12] до легкої фізичної роботи відносяться всі види діяльності, виконувані сидячи і ті, що не потребують фізичної напруги. Робота користувача ПК відноситься до категорії 1а.

При роботі на ПЕОМ користувач піддається ряду потенційних небезпек. Унаслідок недотримання правил техніки безпеки при роботі з машиною (невиконання огляду відкритих частин ПЕОМ, що знаходяться під напругою або знятих для ремонту вузлів) для користувача існує небезпека поразки електричним струмом.

Джерелами підвищеної небезпеки можуть служити наступні елементи:

- розподільний щит;
- джерела живлення;
- блоки ПЕОМ і друку, що знаходяться в ремонті.

Ще одна проблема полягає у тому, що спектр випромінювання комп'ютерного монітора включає рентгенівську, ультрафіолетову і інфрачервону області, а також широкий діапазон хвиль інших частот. Небезпека рентгенівського проміння мала, оскільки цей вид випромінювання

поглинається речовиною екрану. Проте велику увагу слід приділяти біологічним ефектам низькочастотних електромагнітних полів (аж до порушення ДНК).

Відповідно до [13], при обслуговуванні ПЕОМ мають місце фізичні і психофізичні небезпечні, а також шкідливі виробничі чинники:

- підвищене значення напруги в електричному ланцюзі, замикання
- якої може відбутися через тіло людини;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітних випромінювань;
- підвищена або знижена температура повітря робочої зони;
- підвищений або знижений рух повітря;
- підвищена або знижена вологість повітря;
- відсутність або недостатність природного світла;
- підвищена пульсація світлового потоку;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- розумове перенапруження;
- емоційні навантаження;
- монотонність праці.

#### **4.2 Заходи щодо техніки безпеки**

Основним небезпечним чинником при роботі з ЕОМ є небезпека поразки людини електричним струмом, яка посилюється тим, що органи чуття людини не можуть на відстані знайти наявності електричної напруги на устаткуванні.

Проходячи через тіло людини, електричний струм чинить на нього складну дію, що є сукупністю термічної (нагрів тканин і біологічних середовищ), електролітичної (розкладання крові і плазми) і біологічної

(роздратування і збудження нервових волокон і інших органів тканин організму) дій.

Тяжкість поразки людини електричним струмом залежить від цілого ряду чинників:

- значення сили струму;
- електричного опору тіла людини і тривалості протікання через нього струму;
- роду і частоти струму;
- індивідуальних властивостей людини і навколишнього середовища.

Розроблений дипломний проект передбачає наступні технічні способи і засоби, що застерігають людину від ураження електричним струмом [14]:

- заземлення електроустановок;
- занулення;
- захисне відключення;
- електричне розділення ятерів;
- використання малої напруги;
- ізоляція частин, що проводять струм;
- огорожа електроустановок.

Занулення зменшує напругу дотику і обмежує година, протягом якого людина, ткнувшись до корпусу, може потрапити під дію напруги.

Струм однофазного короткого замикання визначається по наближеній формулі:

$$I_k = \frac{U_\phi}{Z_\Pi + \frac{Z_\Gamma}{3}}, \quad (4.1)$$

де  $U_\phi$  - номінальна фазна напруга мережі, В;

$Z_\Pi$  - повний опір петлі, створене фазними і нульовими дротами, Ом;

$Z_\Gamma$  - повний опір струму короткого замикання на корпус, Ом.

Згідно таблиці 4 [4]:  $Z_T / 3 = 0,1 \text{ Ом}$ .

Для провідників і жил кабелю для розрахунку повного опору петлі використовуємо формулу(4.2.) :

$$Z_{\Pi} = \sqrt{R_{\Pi}^2 + X_{\Pi}^2}, \quad (5.2)$$

де  $R_{\Pi} = R_{\phi} + R_0$  - сумарний активний опір фазного  $R_{\phi}$  і нульового  $R_0$  дротів, Ом;

$X_{\Pi}$  - індуктивний опір паяння дротів, Ом.

Перетин 1 км мідного дроту  $S = 2.5 \text{ мм}$ , тоді згідно таблицям 5 і 6 [15], має такий опір:

$$X_{\Pi} = 0,11 \text{ Ом};$$

$$R_{\phi} = 7,55 \text{ Ом};$$

$$R_0 = 7,55 \text{ Ом}.$$

$$\text{Отже, } R_{\Pi} = 7,55 + 7,55 = 15,1 \text{ Ом}.$$

Тоді по формулі(4.2) знаходимо повний опір петлі :

$$Z_{\Pi} = \sqrt{15,1^2 + 0,11^2} \approx 15,1 \text{ (Ом)}.$$

Струм однофазного короткого замикання рівний:

$$I_k = \frac{220}{15,1 + 0,1} = 14,47 \text{ (А)}.$$

Дія плавкої вставки на ПЕОМ забезпечується, якщо виконується співвідношення:

$$I_k \geq k * I_n, \quad (4.3)$$

де  $I_n$  - номінальний струм спрацьовування плавкої вставки, А;

$k$  - коефіцієнт кратності нелінійного струму  $I_n$ , А.

Коефіцієнт кратності нелінійного струму  $I_n$  розраховується по формулі(4.4) :

$$I_n = P / U, \quad (4.4)$$

де  $P = 220$  Вт - споживана потужність;

$U = 220$  В - робоча напруга;

$k = 3$  А - для плавких вставок.

Отже,  $I_n = 220 / 220 = 1$  А.

Підставивши значення у вираз(4.3), одержимо:

$$14,47 > 3 * 1.$$

Таким чином, доведено, що апарат забезпечить спрацьовування (і захист) при підвищенні номінального струму.

### **4.3 Заходи, що забезпечують виробничу санітарію і гігієну праці**

Вимоги до виробничих приміщень встановлюються [16], СНіП, відповідними ГОСТами і ОСТами з урахуванням небезпечних і шкідливих чинників, що утворюються в процесі експлуатації електроустаткування.

Підвищення працездатності людини і збереження її здоров'я забезпечується стабільними метеорологічними умовами. Мікроклімат виробничих приміщень [1] визначається діючими на організм людини поєднаннями температури, вологості і швидкості руху повітря, а також температури навколишніх поверхонь. Значне коливання параметрів мікроклімату приводить до порушення систем кровообігу, нервової і потовидільної, що може викликати підвищення або пониження температури тіла, слабкість, запаморочення і навіть непритомність.

Відповідно до [12] встановлюють оптимальну і допустиму температуру, відносну вологість і швидкість руху повітря в робочій зоні. За відсутності надмірного тепла, вологи, шкідливих речовин в приміщенні досить природної вентиляції.

У приміщенні для виконання робіт операторського типу (категорія 1а), пов'язаних з нервово-емоційною напругою, проектом передбачається дотримання наступних нормованих величин параметрів мікроклімату (табл. 4.1).

Таблиця 4.1 - Санітарні норми мікроклімату робочої зони приміщень для робіт категорії 1а.

Пора року	Температура, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодна	22...24	40...60	0,1
Тепло	23...25	40...60	0,1

У приміщенні, де знаходиться ПЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (з пристроєм вентиляційних каналів в перекриттях будівлі і вертикальних шахт) й устанавленого промислового кондиціонера фірми Mitsubishi, який дозволяє вирішити переважну більшість завдань по створенню та підтримці необхідних параметрів повітряного середовища. Цей метод забезпечує приток потрібної кількості свіжого повітря, визначеного в СНіП (30 м<sup>3</sup> в годину на одного працівника).

Шум на виробництві має шкідливу дію на організм людини. Стомлення операторів через шум збільшує число помилок при роботі, призводить до виникнення травм. Для оператора ПЕОМ джерелом шуму є робота принтера. Щоб усунути це джерело шуму, використовують наступні методи. При покупці принтера слід вибирати найбільш шумозахисні матричні принтери або з великою швидкістю роботи (струменеві, лазерні). Рекомендується

принтер поміщати в найбільш віддалене місце від персоналу, або застосувати звукоізоляцію та звукопоглинання (під принтер підкладають демпфуючі підкладки з пористих звукопоглинальних матеріалів з листів тонкої повсті, поролону, пеноплону).

При роботі на ПЕОМ, проектом передбачені наступні методи захисту від електромагнітного випромінювання: обмеження часом, відстанню, властивостями екрану.

Обмеження годині роботи на ПЕОМ складає 3,5-4,5 години. Захист відстанню передбачає розміщення монітора на відстані 0,4-0,5 м від оператора. Передбачений монітор 20" TFT, Samsung 2043BW відповідає вимогам стандарту [17].

Стандарт [17] пред'являє жорсткі вимоги в таких областях: ергономіка (фізична, візуальна і зручність користування), енергія, випромінювання(електричних і магнітних полів), навколишнє середовище і екологія, а також пожежна та електрична безпека, які відповідають всім вимогам [13].

Для зниження стомлюваності та підвищення продуктивності праці обслуговуючого персоналу в колірній композиції інтер'єру приміщень для ПЕОМ дипломним проектом пропонується використовувати спокійні колірні поєднання і покриття, що не дають відблисків.

У проекті передбачається використання сумісного освітлення. У світлий час доби приміщення освітлюватиметься через віконні отвори, в решту часу використовуватиметься штучне освітлення.

Як штучне освітлення необхідно використовувати штучне робоче загальне освітлення. Для загального освітлення необхідно використовувати люмінесцентні лампи. Вони володіють наступними перевагами: високою світловою віддачею, тривалим терміном служби, хоча мають і недоліки: високу пульсацію світлового потоку.



При експлуатації ПЕОМ виробляється зорова робота. Відповідно до [18] ця робота відноситься до розряду 5а. При цьому нормоване освітлення на робочому місці( $E_n$ ) при загальному освітленні дорівнює 200 лк.

Приміщення завдовжки 12 м, шириною 10 м, заввишки 4 м обладнується світильниками типу ЛП02П, оснащеними лампами типу ЛБ зі світловим потоком 3120 лм кожна.

Виконаємо розрахунок кількості світильників в робочому приміщенні завдовжки  $a=12$  м, шириною  $b=10$  м, заввишки  $z=4$  м, використовуючи формулу (4.5) розрахунку штучного освітлення при горизонтальній робочій поверхні методом світлового потоку:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.5)$$

де  $F$  - світловий потік = 3120 лм;

$E$  - максимально допустима освітленість робочих поверхонь = 200 лк;

$S$  - площа підлоги = 120 м<sup>2</sup>;

$Z$  - поправочний коефіцієнт світильника = 1,2;

$k$  - коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації світильників = 1,5;

$n$  - кількість світильників;

$U$  - коефіцієнт використання освітлювальної установки = 0,6;

$M$  - кількість ламп у світильнику = 2.

З формули(4.5) виразимо  $n$ (4.6) і визначимо кількість світильників для даного приміщення:

$$n = (E \cdot S \cdot Z \cdot k) / (F \cdot U \cdot M), \quad (4.6)$$

Отже,  $n = (200 \cdot 120 \cdot 1,2 \cdot 1,5) / (3120 \cdot 0,6 \cdot 2) = 12$  .

Виходячи з цього, рекомендується використовувати 12 світильників. Світильники слід розміщувати рядами, бажано паралельно стіні з вікнами. Схема розташування світильників зображена на рис. 4.1.

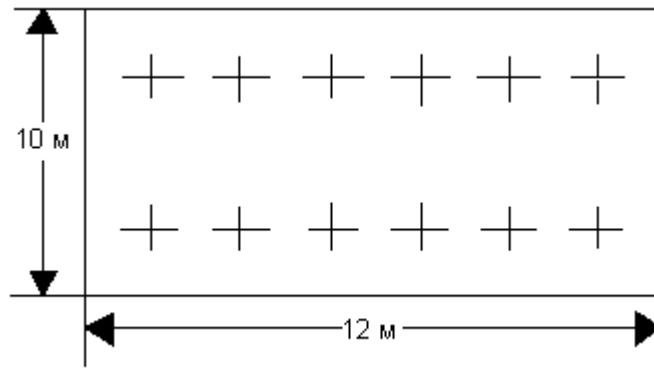


Рисунок 4.1 - Схема розташування світильників

#### 4.4 Рекомендації по пожежній безпеці

Пожежі в приміщеннях, де встановлена обчислювальна техніка, представляють небезпеку для життя людини. Пожежі також пов'язані як з матеріальними втратами, так і з відмовою засобів обчислювальної техніки, що у свою чергу спричиняє за собою порушення ходу технологічного процесу.

Пожежа може виникнути при наявності горючої речовини та внесення джерела запалювання в горюче середовище. Пальними матеріалами в приміщеннях, де розташовані ПЕОМ, є:

- поліамід - матеріал корпусу мікросхеми, горюча речовина, температура самозаймання аерогелю 420 З ;

- полівінілхлорид - ізоляційний матеріал, горюча речовина, температура запалювання 335 З, температура самозаймання 530 З, кількість енергії, що виділяється при згоранні - 18000 - 20700 кДж/кг;

– стеклотекстоліт ДЦ - матеріал друкарських плат, важкозаймистий матеріал, показник горючості 1.74, не схильний до температурного самозаймання;

– пластика кабельний №489 - матеріал ізоляції кабелю, горючий матеріал, показник горючості більш 2.1;

– деревина - будівельний і обробний матеріал, матеріал з якого виготовлені меблі, горючий матеріал, показник горючості більше 2.1, теплота згорання 18731 - 20853 кДж/кг, температура запалювання 399 З, схильна до самозаймання [15].

Згідно [19] приміщення відносяться до категорії В (пожежовибухонебезпечним) і згідно правилам побудови електроустановок простір усередині приміщення відноситься до вогнебезпечної зони класу П - Па (зони, розташовані в приміщеннях, в яких зберігаються тверді горючі речовини).

Потенційними джерелами запалення при роботі ПЕОМ є:

- іскри при замиканні і розмиканні ланцюгів;
- іскри і дуги коротких замикань;
- перегріву від тривалого перевантаження і наявності перехідного опору.

Продуктами згорання, що виділяються при пожежі, є: оксид вуглецю, сірчистий газ, оксид азоту, синильна кислота, акропеїн, фосген, хлор та ін. При горінні пластмас, окрім звичайних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол та ін., що шкідливо впливають на організм людини.

Для захисту персоналу від дії небезпечних і шкідливих чинників пожежі проектом передбачається застосування промислового протигаза з коробкою марки В (жовта).

Пожежна безпека об'єктів народного господарства регламентується [9] і забезпечується системами запобігання пожежам і протипожежному захисту.

Для успішного гасіння пожеж вирішальне значення має швидке виявлення пожежі і своєчасний виклик пожежних підрозділів до місця пожежі.

Зменшити горюче навантаження не представляється можливим, тому проектом передбачається застосувати наступні способи і їх комбінації для запобігання утворенню(внесення) джерел запалення :

- застосування устаткування, що задовольняє вимогам електростатичної безпеки;
- застосування в конструкції швидкодіючих засобів захисного відключення можливих джерел запалення;
- виключення можливості появи іскрового заряду статичної електрики в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення;
- підтримка температури нагріву поверхні машин, механізмів, устаткування, пристроїв, речовин і матеріалів, які можуть увійти до контакту з палим середовищем, нижче гранично допустимої, становить 80% якнайменшої температури самозаймання пального.
- заміна небезпечних технологічних операцій більш безпечними;
- ізольоване розташування небезпечних технологічних установок і устаткування;
- зменшення кількості палих і вибухонебезпечних речовин, що знаходяться у виробничих приміщеннях;
- запобігання можливості утворення палих сумішей на лінії, вентиляційних системах і ін.;
- механізація, автоматизація та справність(потоківа) виробництва;
- суворе дотримання стандартів і точне виконання встановленого технологічного режиму;
- запобігання можливості появи в небезпечних місцях джерел запалення;
- запобігання розповсюдженню пожеж і вибухів;

- використання устаткування і пристроїв, при роботі яких не виникає джерел запалення;
- виконання вимог сумісного зберігання речовин і матеріалів;
- наявність громовідводу;
- організація автоматичного контролю параметрів, що визначають джерела запалення;
- ліквідація можливості самозаймання речовин і матеріалів.

Для запобігання пожежі в обчислювальних центрах проектом пропонується виконання наступних вимог:

- електроживлення ЕОМ повинно мати автоматичне блокування відключення електроенергії на випадок зупинки системи охолодження і кондиціонування;
- система вентиляції обчислювальних центрів повинна бути обладнана блокуючими пристроями, що забезпечують її відключення на випадок пожежі;
- робочі місця повинні бути оснащені пожежними щитами, сигналізацією, засобами для сповіщення про пожежну небезпеку (телефонами), медичними аптечками для надання першої медичної допомоги, розробленим планом евакуації.

Для зниження пожежної небезпеки в приміщеннях використовуються первинні засоби гасіння пожеж, а також система автоматичної пожежної сигналізації, яка дозволяє знайти початкову стадію загоряння, швидко і точно оповістити службу пожежної охорони про час і місце виникнення пожежі.

Відповідно до [21] приміщення категорії В підлягають устаткуванню системами автоматичної пожежної сигналізації. Проектом передбачається застосування датчика типу ІДФ - 1(димовий фотоелектричний датчик), оскільки специфікою пожеж обчислювальної техніки і радіоапаратури є, в першу чергу, виділення диму, а потім - підвищення температури.

При виникненні пожежі в робочому приміщенні обслуговуючий персонал зобов'язаний негайно вжити заходи по ліквідації пожежі. Для

ліквідації пожежі використовують вогнегасники (хімічно-пінні, пінні для повітря ОП-5, ОП-6, ОП-9, вуглекислотні ОУ-5), пісок, пожежний інвентар(сокири, ломи, багри, шерстяну або азбестову ковдри) [22]. Як засіб індивідуального захисту проектом передбачається використання промислового протигаза з маскою, фільтруючої коробки В.

В якості організаційно-технічних заходів рекомендується проводити навчання робочого персоналу правилам пожежної безпеки.

## ВИСНОВКИ

У рамках дипломної роботи був розроблений і реалізований програмний комплекс розпізнавання фруктів та овочів за допомогою методу Віюли-Джонса та ознак Хаара.

Завдання, поставлені перед бакалаврською дипломною роботою виконанні в повному об'ємі, а саме:

- 1) Вибраний метод розпізнавання.
- 2) Поставлена задача для реалізації .
- 3) Розроблений алгоритм функціонування комплексу розпізнавання овочів і фруктів на фотокартці.
- 4) Спроектований концепт Android - додатку.
- 5) Проведено тренування системи для виділення ознак Хаара.
- 6) Розроблена програмна реалізація концепту та методу розпізнавання.
- 7) Створена інструкція користувача
- 8) Проведено тестування вихідного комплексу на тестових зображеннях.

Програмний комплекс виконаний у середовищі Android Studio з використанням мови програмування Java та бібліотеки OpenCV.

Модуль дозволяє знаходити шукані овочі та фрукти на знімку продуктової корзини в супермаркеті, після чого виводить інформацію про калорійну цінність кожного з них. Метод виявлення побудований з використанням boosting-алгоритму навчання, тренований каскад сильних класифікаторів, в яких в якості простих класифікаторів використані хаароподібні ознаки.

Під час четвертого розділу було проаналізовано умови праці, виявлені причини травматизму і захворювань, можливі небезпечні й шкідливі виробничі фактори. Також було проведено ряд розрахунків щодо виконання вимог охорони праці в приміщенні відділу програмного забезпечення.

Дотримання цих вимог є важливим для збереження працездатності та здоров'я працівників.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

- 1) Viola, P. Fast and Robust Classification Using Symmetric AdaBoost and a Detector Cascade [Текст] / P. Viola, M. Jones. – 2001. – P. 1311–1318
- 2) Viola, P. Rapid object detection using a boosted cascade of simple features [Текст] / P. Viola, M.J. Jones // IEEE Conf. On Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, 2001. – V. 1. – P. 511–518.
- 3) Gomez G., Morales E. Automatic feature construction and a simple rule induction algorithm for skin detection [Текст] // In Proc. of the ICML Workshop on Machine Learning in Computer Vision, 2002. – 31-38.
- 4) Ту, Дж. Принципы распознавания образов [Текст] / Дж. Ту, Р. Гонсалес. – М.: Мир, 2011. – 403 с.
- 5) Фуканага, К. Введение в статистическую теорию распознавания образов [Текст] / К. Фуканага. – М.: Наука, 1999. – 368 с.
- 6) Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир [Текст] – 1992. – 184 с.
- 7) Хайкин С. Нейронные сети: полный курс, 2-е издание. [Текст] / М. Вильямс, 2008. – 1103 с.
- 8) Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. [Текст] / А.Н. Горбань, Д.А. Россиев – Н.: Наука, 2006. – 276с.
- 9) Fisher, B., Perkins S., Walker A. HyperMedia Image Processing Reference [Электронный ресурс]. - 1994. – Режим доступа до ресурсу: [http://www.cee.hw.ac.uk/hipr/html/hipr\\_top.html](http://www.cee.hw.ac.uk/hipr/html/hipr_top.html), вільний.
- 10) Шапиро, Л. Компьютерное зрение [Текст] / Л. Шапиро, Дж. Стокман; перевод с англ. – М.: БИНОМ. Лаборатория знаний, 2006. – 752 с.
- 11) Вікіпедія – вільна енциклопедія [Електронний ресурс]: – Режим доступу: <http://wikipedia.org>..

12) Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. Постанова N 42 від 01.12.99. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/va042282-99](http://www.url:https://zakon.rada.gov.ua/rada/show/va042282-99)

13) Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПН 3.3.2.007-98. Затверджено Постановою Головного державного санітарного лікаря України 10 грудня 1998 р. N 7. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98](http://www.url:https://zakon.rada.gov.ua/rada/show/v0007282-98)

14) НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів». *Наказ від 09.01.98 №4*. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0093-98](http://www.url:https://zakon.rada.gov.ua/laws/show/z0093-98)

15) ГОСТ 12.1.044-89 «Система стандартів безпеки праці. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения». Постанова від 12.12.1989 № 3683. Режим доступу: [www. URL: http://online.budstandart.com/ru/catalog/doc-page?id\\_doc=51048](http://www.url:http://online.budstandart.com/ru/catalog/doc-page?id_doc=51048)

16) ДСП 173-96 «Державні санітарні правила планування та забудови населених пунктів». *Наказ від 19.06.1996 №173*. Режим доступу: [www. URL: https://zakon.rada.gov.ua/laws/show/z0379-96](http://www.url:https://zakon.rada.gov.ua/laws/show/z0379-96)

17) TCO'07 Certified Displays. © 2007 Copyright TCO Development AB. Режим доступу: [www. URL: https://tcocertified.com/files/2015/11/TCO-Certified-Displays-7.0.pdf](http://www.url:https://tcocertified.com/files/2015/11/TCO-Certified-Displays-7.0.pdf)

18) ДБН В.2.5-28:2018 «Природне і штучне освітлення». Режим доступу: [www. URL: http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf](http://www.url:http://www.minregion.gov.ua/wp-content/uploads/2018/12/V2528-1.pdf)

19) ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою». *Наказ від 15.06.2016 №158*. Режим доступу: [www. URL: https://zakon.rada.gov.ua/rada/show/v0158858-16](http://www.url:https://zakon.rada.gov.ua/rada/show/v0158858-16)

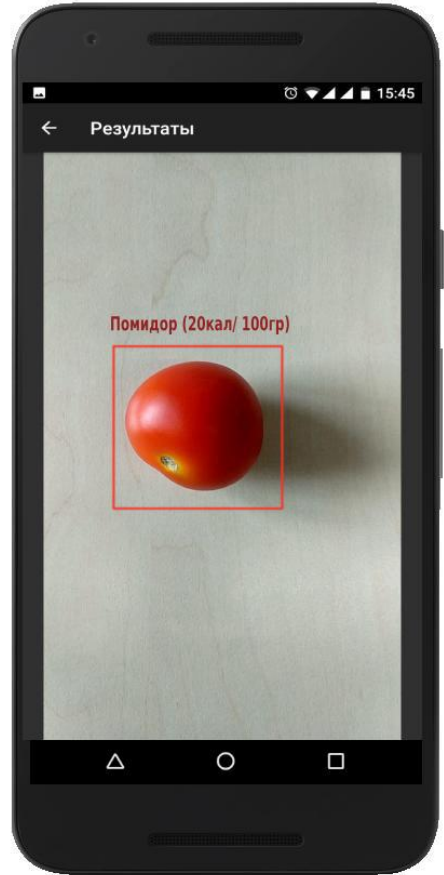
20) ГОСТ 12.1.004-91 «Система стандартов безопасности труда. Пожарная безопасность. Общие требования». Режим доступа: www. URL: [http://online.budstandart.com/ua/catalog/doc-page.html?id\\_doc=48679](http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=48679)

21) НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні». Наказ від 30.12.2014 №1417. Режим доступу: www. URL: <https://zakon.rada.gov.ua/laws/show/z0252-15>

22) НАПБ Б.01.008-2018 «Про затвердження правил експлуатації та типових норм належності вогнегасників». Наказ від 15.01.2018 №25. Режим доступу: www. URL: [http://search.ligazakon.ua/1\\_doc2.nsf/link1/RE31677.html](http://search.ligazakon.ua/1_doc2.nsf/link1/RE31677.html)

ДОДАТОК А.  
Тестові зображення





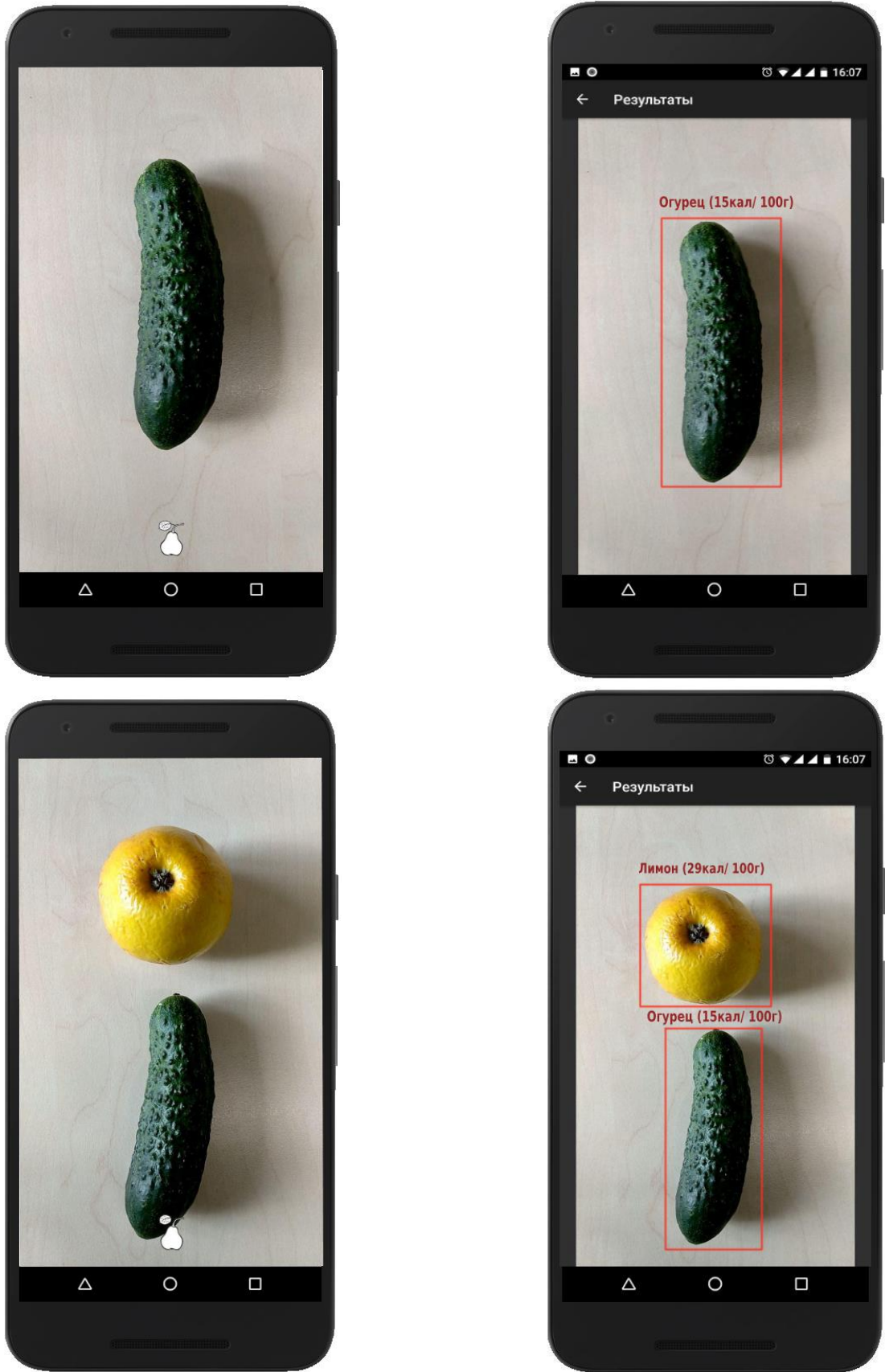


Рисунок А.1 – Приклади тестових зображень

## ДОДАТОК Б.

### Електронні плакати

Міністерство освіти і науки України  
Східноукраїнський національний університет ім.В.Дала

## ДИПЛОМНА РОБОТА БАКАЛАВРА

Мобільний додаток розпізнавання харчових продуктів

Виконав  
Ст. гр. КН-15з  
Панченко К.О.

Керівник  
Доц. Барбарук В.М.

## Постановка задачі

- Розглянути основні методи розпізнавання об'єктів на зображенні та постановка задачі
- Побудувати математичну модель методу Віолі-Джонса для знаходження образів
- Розробити та протестувати комп'ютерну модель розпізнавання фруктів

## Метод Віоли-Джонса

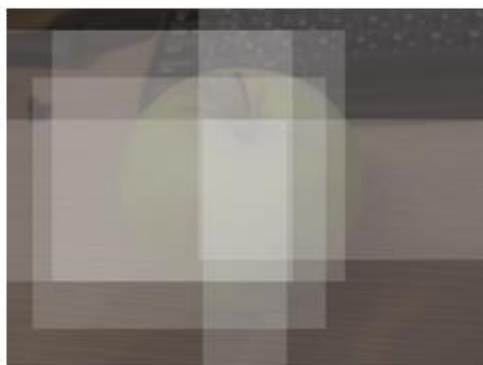
---

- Використання зображення в інтегральному уявленні
- Використання ознак Хаара
- Використання “boosting” для вибору найбільш підходящих ознак
- Надходження ознак до класифікатору
- Використання каскадів ознак

## Ознаки Хаара

---

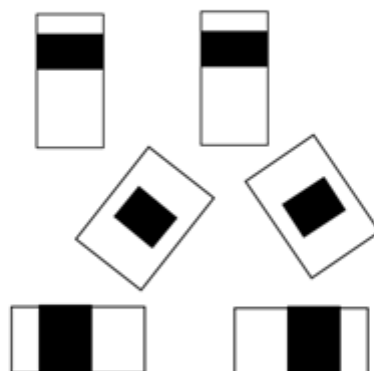
Процес аналізу зображення з використанням ознак Хаара методом Віоли-Джонса





## Ознаки Хаара

---



Додаткові ознаки Хаара

## Архітектура проекту

---



## Програмна реалізація



## Програмна реалізація



## Висновки

---

У рамках агестаційної роботи був розроблений і реалізований програмний комплекс розпізнавання фруктів та овочів за допомогою методу Віолі-Джонса та ознак Хаара.

Дякую за увагу!

---