

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

ВИПУСКНА РОБОТА

на тему:

**«Інформаційна система пошуку і розпізнавання об'єкта
зацікавленості»**

Завідувач

Випускаючої кафедри

Керівник роботи

Студента групи Інз-61с

Довбиш А.С.

Симоновський Ю.В.

Коропченко Ю.В.

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2020 г.

ЗАВДАННЯ

до випускної роботи

Студента п'ятого курсу, групи ІНЗ-61с спеціальності “Інформатика”
заочної форми навчання Коропченко С.В.

**Тема: “ Інформаційна система пошуку і розпізнавання об'єкта
зацікавленості ”**

Затверджена наказом по СумДУ

№ _____ от _____ 2013 г.

Зміст пояснювальної записки: 1) Літературний огляд актуальності теми;
2) постановка завдання й формування завдань дослідження; 3) Вибір
середовища розробки; 4) Розробка інформаційної системи.

Дата видачі завдання “ _____ ” _____ 2020 г.

Керівник випускної роботи _____ Симоновський Ю.В.

Завдання прийняв до виконання _____

РЕФЕРАТ

Записка: 34 стор., 13 рис., 1 додаток, 8 джерел.

Об'єкт дослідження — розпізнавання об'єктів зацікавленності.

Мета роботи — розробити інформаційну систему розпізнавання об'єктів на базі нейронної мережі.

Методи дослідження — методи використання комп'ютерного зору при розпізнаванні об'єктів

Результати — створено програмне забезпечення з використанням інтелектуальної системи класифікації об'єктів навколишнього середовища у вигляді мобільного додатку. В роботі проведено оптимізація роботи нейронної мережі на базі бібліотеки комп'ютерного зору OpenCV для пристроїв з операційною системою Android.

СИСТЕМА РОЗПІЗНАВАННЯ ОБ'ЄКТІВ, ШТУЧНА НЕЙРОННА МЕРЕЖА
РОЗПІЗНАВАННЯ ОБРАЗІВ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА

ЗМІСТ

ЗМІСТ	4
ВСТУП.....	5
1 ЛІТЕРАТУРНИЙ ОГЛЯД.....	6
1.1 Аналітичний огляд технологій обробки зображень.....	6
1.2 Підходи до локалізації об'єктів інтересу на зображенні.....	11
1.3 Формалізована постановка задачі.....	155
2 ТЕОРЕТИЧНІ ОСНОВИ.....	177
2.1 Штучні нейронні мережі	177
2.2 Архітектура згорткової нейронної мережі	18
2.2 Архітектура згорткової нейронної мережі.....	21
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	256
3.1 Формування вхідного математичного опису	256
3.2 Короткий опис програмної реалізації алгоритмів	27
3.3 Аналіз отриманих результатів моделювання	28
ВИСНОВКИ	322
СПИСОК ЛІТЕРАТУРИ	333
ДОДАТОК А.....	344

ВСТУП

Функція розпізнавання об'єктів зацікавленості є важливою задачею у сфері комп'ютерного зору. Щоб отримати певний результат потрібно виконати низку послідовних процесів: отримання, обробка, аналіз і класифікація об'єктів з цифрового зображення. Для цього необхідно використовувати ряд методів з таких основних наук як фізика, статистика, геометрія і теорії статичного навчання. Доступність новітніх телефонів з високоякісними камерами пришвидшило розвитк сучасних технологій автоматизації аналізу відео та зображень.

Вирішення завдання пошуку і розпізнавання об'єктів зацікавленості ускладнюється надлишковою інформацією на яку впливає навколишнє середовище та технічні характеристики телефона: освітлення, побічні предмети, якість зображення, зміна положення камери, оклюзія об'єктів.

Головними перевагами цього програмного додатку є її портативність та простота використання будь-якою аудиторією користувачами смартфонів.

Основним завданням бакалаврської роботи є реалізація програми пошуку та розпізнавання об'єкта зацікавленості з використанням інтелектуальної мережі для аналізу отриманих зображень котра здатна працювати на більшості пристроїв з операційною системою Android.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналітичний огляд технологій обробки зображень

В наш час фото і відео-зображення використовуються у багатьох галузях і для виконання необхідних цілей: військові, наукові, розважальні, промислові, цивільні та медичні за допомогою камери телефону, відеокамери, радарів, сканерів та електронних мікроскопів. Для того щоб машина змогла сприйняти інформацію з зображення воно має пройти цифрову обробку тому це дуже важливий етап без якого її подальше обробка буде неможливою. З боку обчислюваної машини аналіз зображення має зовсім інший вигляд від звичайної фотографії, пристрій бачить тільки набір чисел у різній послідовності в залежності від стану і різновидності об'єкта. Цей процес називається оцифровкою зображення тобто перетворення образотворчої форми у числові дані. Знімок розділений на дрібні сегменти, які називаються пікселями або елементами зображення (рисунок 1.1).

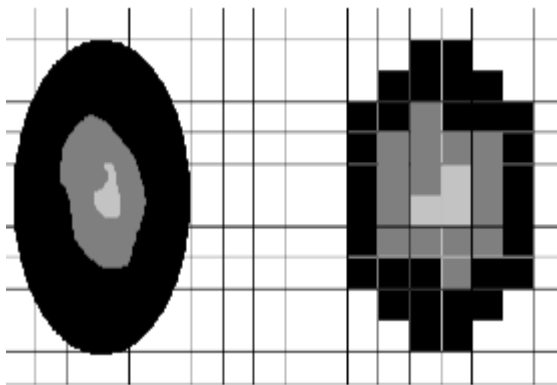
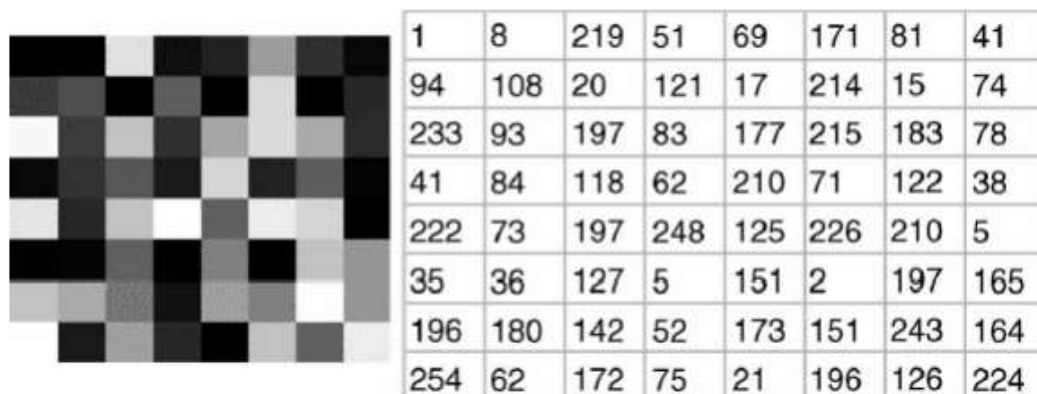


Рисунок 1.1 – Оцифровка зображення

Для правильного виконання цього процесу оцифровка відбувається на координатах (x, y) і на амплітуді яскравості. Знімок перетворився на матриці цілих чисел. Рівні яскравості та розміри зображення іноді бувають дуже великого розміру. Кожен номер у пікселі означає інтенсивність кольору в цій точці. Для кращого розуміння необхідно звернути увагу на рисунок 1.2, на ньому

намальовано перетворене зображення 8x8 з байтом на піксель, можна зробити висновок що 8 біт це 256 рівнів сірого.



Якість зображення має повну залежність від величини рівнів та зразків яскравості (тобто сірого кольору). Також цей параметр відображається на обсязі пам'яті яку займає велике зображення через розмір та кількість бітів. Якщо навпаки розмір буде малий то це може стати причиною ефекта зернистості. Коли зображення розміром 1920×1024 змінюється на 512×1024, різниця між ними не буде виділятися дуже сильно, але у випадку зміни до 128×256 та повторному збільшенні до 512×1024 за допомогою дублювання буде можливо помітити істотну зернистість. Обробка зображень класифікується на чотири види операцій:

1) Операції з пікселями: Цей процес відбувається не зважаючи на всі інші пікселі у представленому екземплярі тобто все залежить лише від даного пікселю який обрано як вхідний. Для цього використовуються функція порогування яка розподіляє їх в залежності від порогових рівнів білого та чорного. У деяких інших випадках вмикають інвертування зображення, підвищення та пониження яскравості, контрастне розтягнення.

2) Локальні (сусідні) операції: Цей процес базується на найближчих вхідних даних від обраного пікселя. В більшості випадків операція адаптивна, тому що все залежить від точних значень для пікселів які перебувають на всій області матриці. Найкращий приклад це фільтри які використовують для покращення текстури або кольорів, відтінків і тому подібне.

3) Геометричні операції: Процес має залежність від результатів геометричних перетворень вхідних даних для обраних пікселів. Це допомагає зменшити кількість необхідних сигментів на відміну від глобальних перетворень.

4) Глобальні операції: Перехід до пікселів залежить від усіх сигментів зображення при цьому здійснюється перетворення зображення зі зміщенням пікселів на мінімальну відстань до фонових сигментів.

Розпізнавання образів - наукова дисципліна, складовими якої є ідентифікація особливостей і властивостей з подальшою класифікацією образів або об'єктів у певній області. Техніки розпізнавання образів є одною з головних складових інтелектуальних систем і широко використовується в різних напрямках.

Образ - група об'єктів які об'єднанні спільними рисами і ознаками тим самим формуючи класифікацію

Розпізнавання образів - це процес розпізнавання об'єктів на зображенні за допомогою алгоритму машинного навчання. Розпізнавання образів може трактуватися як класифікація отриманих даних на основі вивчених знань або статистичної інформації. Усі знання для порівняння задаються для кожного з класів.

Причини використання технології розпізнавання образів:

- Розпізнавання місцевості;
- Класифікація змісту тексту;
- Розпізнавання відео-матеріалів;
- Система безпеки зі сканером ока.

У типовій системі всі отримані дані проходять перетворення у форму для коректного використання машиною. Для якісної роботи системи на зображенні виділяється необхідний об'єкт обмежений в певній області.

При знаходженні відповідності між об'єктом і знаннями, об'єкт отримує мітку класу тобто класифікація відбулася.

Розділення на частини допомагає при знаходженні рішення та має цілий розділ для цього. Кластеризація використовується у навчанні без нагляду.

Можливе предсавлення особливостей за допомогою дискретних, безперервних, дискретних бінарних змінних. Алгоритм розрахунків знаходження важливих параметрів об'єкта називається функцією особливості. На основі отриманих даних формується вектор властивостей.

Необхідні вимоги до системи:

- чітко та швидко знаходити вивчений образ;
- знаходити закономірності у незнайомих об'єктах і їх класифікація;
- якісно аналізувати форму об'єкта з різних кутів;
- ідентифікувати образи якщо одна з частин відсутня;
- автоматично виводити результат роботи.

Навчання - процес перетворення у адаптивну систему за допомогою аналізу тестових результатів. Це дуже важливий процес тому що від флгоритму залежить якість та точність результатів системи розпізнавання образів. До складу необхідних даних входить навчальний та тестовий набори, перший для підготовки системи та налагодження головної функції а другий для максимально наближеної до реальних умов перевірки яка і визначає якість системи. (рисунок 1.3).

Застосування навчального набору приймає участь у побудові моделі. Він має вигляд збірки зображень які аналізує система (тобто тренується). Результат навчання детермінований алгоритмом і правилами навчання розтлумачуючи зв'язок між вхідними даними та результатом роботи. Система виконує алгоритм тим самим удосконалючи вихідні дані на основі навчального набору даних.

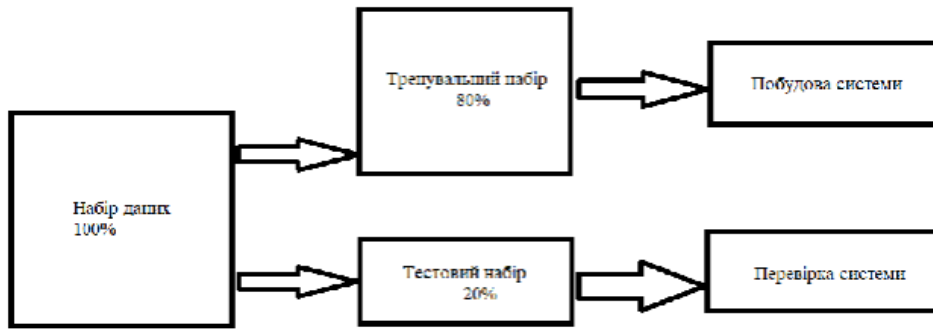


Рисунок 1.3 – Модель навчального набору інтелектуальної системи

Після цього система перевіряється за допомогою тестового набору, який орієнтований на перевірку вірності результатів. За звичай використовується п'ята частина з набору даних, цього має бути достатньо щоб виміряти точність.

Головні переваги системи:

- розподіляє вихідні результати по класам;
- знаходить більш точну інформацію біометричних показників;
- допомагає ідентифікувати походження джерела звуку;
- дає можливість знайти та розпізнати обраний об'єкт з різних ракурсів.

Недоліки:

- сповільнення швидкості роботи через підвищення складності синтаксичного образу.

- у більшості випадків необхідний великий набір даних.
- не конкретизоване пояснення розпізнавання обраного об'єкта.

Застосування:

- обробка зображень, сегментація та аналіз. Розпізнавання образу людини через передачу інформації аналізу до машини.

- комп'ютерний зір. Розпізнавання вже відомих об'єктів з бази даних та порівняння особливостей з новим об'єктом на фото або відео має дуже широке використання.

- сейсмічний аналіз. Виділення необхідних частин знімку з образами, інтерпретація тимчасових об'єктів у записах сейсмічного аналізу.

- класифікація / аналіз радіолокаційного сигналу. Розпізнавання збігу з вивченими даними та методи обробки вхідних сигналів має застосування у багатьох додатках класифікації радіолокаційного сигналу.

- розпізнавання мови. Найкращий результат було досягнуто за допомогою шаблонів розпізнавання образів. Він має широке коло використання у різних алгоритмах та трактує більші одиниці (тобто слова) .

- ідентифікація відбитків пальців. Зараз представляє собою одну з самих актуальних технологій на біометричному ринку. Для цього використовуються найкращі методи та підходи до розпізнавання образів.

1.2 Підходи до локалізації об'єктів інтересу на зображенні

Для вирішення задачі локалізації об'єктів на цифрових зображеннях і КСТ застосовуються різні методи цифрової обробки зображень. Під локалізацією розуміється процес визначення областей об'єктів на зображеннях за заданими параметрами.

Основними вимогами, що пред'являються до КСТ, є максимальні значення точності і швидкості обчислень при їх функціонування. З цього випливає, що вимога мінімального часу роботи в першу чергу пред'являється до алгоритму вилучення ознак. Точність, у свою чергу, визначається складністю поділу на класи отриманих ознак. У першому критеріями для порівняльного аналізу методів вилучення ознак можуть бути:

- час роботи алгоритму;
- складність у поділі ознак на класи.

Задача локалізації розуміється як задача віднесення певного прикладу до одного з декількох множин об'єктів, що попарно не перетинаються. В реальних ситуаціях при класифікації об'єктів виділяють три рівні складності:

- класи можна розділити прямими лініями (або гіперплощинами);
- лінійна подільність;

- класи можна розділити кількома гіперплощинами – нелінійна подільність;

- класи перетинаються, тому їх поділ можна трактувати тільки через ймовірнісне співвідношення.

Формалізувати і кількісно визначити ступінь складності в розділенні класів ознак практично неможливо. Тим не менш, зробити таку оцінку можна на підставі наступного міркування: чим простіше класи поділяються, тим нижче помилка класифікації. Чим нижче помилка класифікації, тим вищий коефіцієнт розпізнавання.

Оцінка коефіцієнта розпізнавання зазвичай проводиться на еталонних базах. Так, наприклад, для задачі пошуку облич на зображенні найбільш поширеними є бази MIT/CMU і BioID. Практика показує, що для бази MIT/CMU коефіцієнт розпізнавання вважається високим при його значенні 75% і вище, а для BioID – 90% і вище. Найбільш популярні бази для ідентифікації осіб – FERET, ORL, FACES, AR FACE. Для них значення якості розпізнавання, що починається зі значення 90% з використанням додаткової інформації про координати очей, вважається високим. Таким чином, для проведення порівняльного аналізу методів вилучення ознак виділені наступні критерії:

- час роботи алгоритму (вимірюється в мілісекундах і обмежено значенням 0,025 мс);

- коефіцієнт розпізнавання (приймає значення від 0% до 100%).

Методи обробки цифрових зображень, що застосовуються для локалізації об'єктів на зображеннях

Аналіз літератури, дозволив виділити декілька підходів до класифікації методів обробки зображень для локалізації об'єктів. Методи обробки різняться в залежності від типу зображення локалізованого об'єкта (об'єкт може бути представлений точковим зображенням, групою точок, у вигляді майданного або протяжного зображення).

У залежності від етапу процесу локалізації (методи первинного і вторинного пошуку), залежно від застосовуваної процедури пошуку (прогресивна сканування всього зображення, порівняння виділених областей з еталоном, обробка окремих фрагментів скануючим «вікном», знаходження опорних точок, геометричний пошук).

Існують дві основні групи ознак кольорових зображень: це ознаки пікселя зображення, зазвичай це характеристики кольору пікселя, і ознаки шуканого об'єкта, тобто безлічі пікселів, з яких і складається зображення об'єкта, зазвичай це геометричні характеристики.

При виборі значень параметрів локалізації, виходячи з змінності умов отримання зображень, з метою виключення можливості втрати області об'єкту, значення параметрів задаються максимальним діапазоном можливих значень. Виходячи з цього, не виключена ймовірність локалізації помилкових областей об'єктів.

На кожному етапі локалізації можливе використання різних параметрів об'єктів, тому методів локалізації по якомусь одному набору ознак у чистому вигляді не існує.

На рис. 1.4 представлена запропонована класифікація застосовуваних методів цифрової обробки для локалізації об'єктів на цифрових 3б зображеннях залежно від відомих апріорних відомостей про об'єкти. Для локалізації об'єктів по апріорним даними про ознаки пікселів зображень.

Методи кореляційного аналізу зводяться до обчислення кореляції зображення заданого об'єкта зі спостережуваним зображенням. Суть локалізації за кореляційним критерієм полягає в тому, щоб виявляти пікселі (фрагменти і цілі зображення) об'єктів, що цікавлять тільки в тому випадку, якщо між їх значеннями яскравості в заданих каналах спостерігається стійкий кореляційний зв'язок. Під час пошуку та локалізації за еталоном виконується кореляційна обробка ознак, отриманих від еталону і вхідного зображення як з

використанням порогів за величиною подібності, так і без встановлення порогу (коли шукається максимум подібності).



Рис. 1.4 – Класифікація методів обробки зображень для локалізації об'єктів

Кореляційні методи характеризується великою обчислювальною складністю. Пов'язано це з масштабуванням і поворотами шуканого об'єкта. Головний недолік методу зіставлення з еталоном полягає в необхідності використання величезної кількості еталонів для обліку змін об'єктів, що

виникають при їх повороті і збільшенні (зменшенні) розмірів, тому кореляція рідко використовується на практиці, якщо можливий довільний поворот шуканого об'єкта. З цієї причини при зіставленні з еталоном бажано обмежитися ознаками, які менше залежать від змін розміру і форми об'єкта, наприклад, структурні ознаки об'єкта.

Аналіз гістограми зображення полягає в побудові розподілу кольорових або статистичних властивостей пікселів на цифровому зображенні. Гістограма розподілу кольорових значень зображення показує процентний вміст точок певної яскравості. Пошук і локалізація областей інтересу проводиться методом установки порогових значень яскравості, і всі точки зображення, що лежать в межах даних значень, виділяються на зображенні, як шукані області.

Якщо контраст об'єкт-фон є достатнім, тоді локалізувати такий об'єкт не складно. Основою таких методів є уявлення про те, що будь-яке зображення можна розбити на кінцеве число сегментів - ділянок, які мають будь-які характеристики. Методи сегментації застосовуються для пошуку об'єктів на цифровому зображенні (пошук країв і зв'язок з ними ліній і кіл), пошуку людей (пошук сегментів тіла за ознаками та об'єднання їх в єдине ціле), пошуку обличчя.

Суть структурних методів полягає у формуванні на основі непохідних елементів зображення складових структурних елементів об'єкта. У якості складових структур служать елементи зображення об'єкта, складені з відрізків прямих ліній: проекції прямокутних паралелепіпедів на площину (у простішому випадку шукаються також паралельні лінії, прямокутники і паралелограми), перебування яких необхідно, наприклад, для виявлення будівель і доріг на зображеннях

1.3 Формалізована постановка задачі

Метою цього проекту є розробка системи пошуку та класифікації об'єктів за допомогою методів машинного навчання – розпізнавання образів.

Відповідно поставленій меті потрібно вирішити такі завдання:

1. Аналіз літературних джерел та інформації щодо застосування методів розпізнавання образів.
2. Створення алгоритму розпізнавання та класифікації образів для визначення об'єкта і його властивостей.
3. Розробка програмного додатку на базі Android який матиме змогу використовувати вище вказаний алгоритм реалізований у вигляді інтелектуальної системи.

2 ТЕОРЕТИЧНІ ОСНОВИ

2.1 Штучні нейронні мережі

Штучні нейронні мережі (ШНН) - це системи обчислювальної обробки, побудовані на прикладі функціонування людського або тваринного мозку, а саме архітектури нервової системи. ШНН складаються з багатьох шарів зв'язаних між собою “нейронів” тобто вузлів для вичислення результату на основі отриманої інформації а також колективного навчання на початку для покращення вихідних даних.

Для кращого розуміння структури нейронної мережі варто звернути увагу на рисунок 2.1. На першому етапі вхідний шар отримує завантажені дані у вигляді багатовимірного вектора після чого результат передається на прихований шар. На другому етапі приймається рішення як краще відкорегувати змінну шляхом зважування та порівняння з кінцевим результатом. У випадку якщо модель має декілька прихованих шарів розміщених один на одного це приклад глибокого навчання.

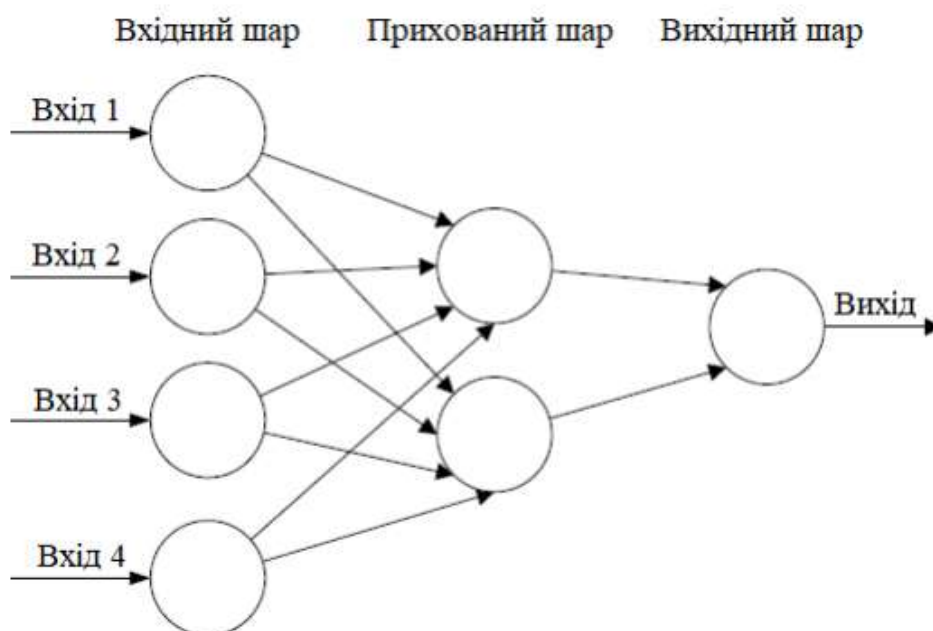


Рисунок 2.1 – Модель простої штучної нейронної мережі з трьома шарами.

Можно виділити два основних методів навчання для обробки зображення. Перший варіант з вчителем, другий без вчителя. Для першого випадку є обов'язкова вимога у вигляді числових позначок, які мають вигляд цілей для виконання. Це використовується для того щоб покращити точність результатів за рахунок зменшення похибки і правильного обчислення на ранніх етапах.

Для другого варіанту розмічені данні не застосовуються, тому успіх фіксується змінами функції витрат наскільки система може збільшити або зменшити її. В нашому випадку для розпізнавання об'єктів частіше всього використовують саме модель з вчителем.

Маємо декілька обмежень через які не має сенсу збільшувати кількість прихованих шарів. Одна з яких це обмеженість технічних засобів які не в змозі навчати дуже великі конструкції. Наступне обмеження полягає у порушенні дотримання заходів обережності що має наслідки надмірного пристосування. Що в свою чергу негативно впливає на здатність навчатися. Для вирішення цієї проблеми понижують складність ШНН, тому що чим менше параметрів необхідно для навчання тим менше шансів пристосування.

2.2 Архітектура згорткової нейронної мережі

Згорткові нейронні мережі (ЗНН) дуже подібні до ШНН, через те що вони однаково складаються з обчислювальних вузлів які самостійно налаштовуються завдяки навчання з функцією оцінки (вагами). В останньому шарі будуть показуватись такі самі результати, рекомендації і функції втрат.

Єдина істотна відмінність між цими двома видами мереж є те що ЗНН повністю пристосована для роботи з зображеннями а це в свою чергу зменшує обсяг необхідних параметрів для налаштування моделі. Ця модель більш вузькоспеціалізована що надає більшої зручності при експлуатації.

Одним з головних недоліків ШНН є те що у більшості випадків відбувається боротьба з обчислювальною складністю для аналізу даних картинки. Тому звичайні ШНН використовують тільки в тому випадку коли необхідно класифікувати відносно маленьке зображення.

В іншому випадку ШНН повинна бути адаптована для певної задачі. Фахівці у цій сфері не радять використовувати цей вид мереж для роботи з зображеннями.

Ще раз звертаю увагу на те що ЗНН сфокусована на початкових даних у вигляді зображень, тому її архітектура повинна відповідати умовам роботи з певним типом даних.

Одна з головних відмінностей ЗНН є порядок нейронів всередині шарів (висота, ширина, глибина) що впливає на меншу кількість зв'язків від стандартної ШНН. Зв'язок для об'єднання будується тільки з невеликою областю шару спереду.

Якщо розглянути попередній приклад параметри для першого шару будуть мати вигляд $X \times Y \times Z$ (X - висота, Y - ширина, Z - глибина). А для останнього шару параметр глибина зміниться на n - можлива кількість класів. Це дозволяє зменшити обсяг оцінок класів за допомогою глибинного виміру.

Для даної моделі можна виділити три види шарів. Це згорткові шари, шарів підвибірки і повністю з'єднаних шарів. За рахунок з'єднання формується дана архітектура згорткової нейронної мережі. На рисунку 2.2 є можливість побачити ілюстрацію спрощеної структури для класифікації MNIST. У цьому випадку головний функціонал можна розділити на чотири ключових блока.

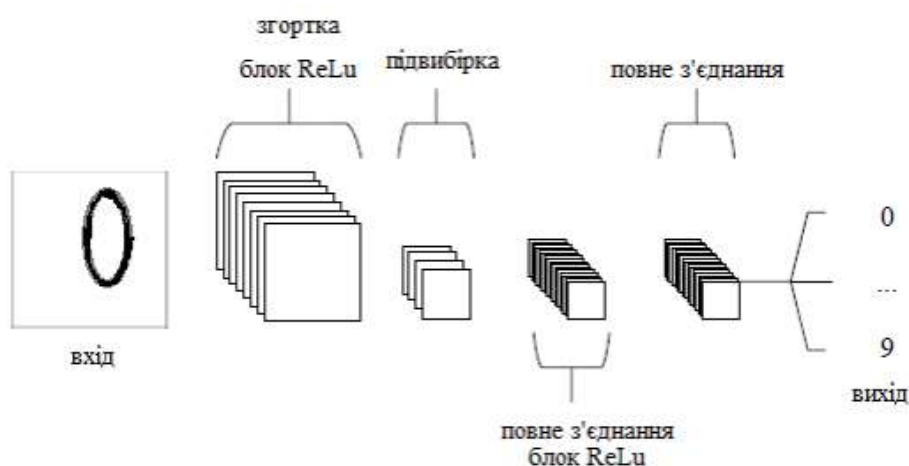


Рисунок 2.2 – Проста архітектура ЗНН для класифікації MNIST

1. Вхідний шар буде вміщати в собі дані пікселів зображень це звичайна практика для всіх ШНН.

2. Згортковий шар знаходитиме потік інформації з нейронів, які підключаються локальними областями введення за допомогою формули. Скалярного добутку між вагою та областю, підключеною до буфера вхідних даних. Блок лінійної ректифікації (зазвичай скорочують до ReLu) має на меті застосувати функцію активації "по елементах", таку як сигмоїд, до виходу активації попереднього шару.

3. Після цього виконується зменшення розмірів вздовж просторової розмірності заданого входу, тим самим ще більш знижуючи величину характеристик у межах цієї активації.

4. У цьому випадку всі об'єднані шари будуть виконувати звичайні функції ШНН а також запитувати оцінки класів для подальшої класифікації. Щоб підвищити швидкість буде розумно додати ReLu між шарами для їх кращої роботи.

Завдяки цьому простому методу перетворення, ЗНН можуть трансформувати початковий вхідні дані шар за шаром за допомогою згорткових методів та технік децимації для створення оцінки класів для цілей класифікації та регресії.

Результатом звичайного методу перетворення, ЗНН можуть перетворювати базові вхідні дані послідовно шар за шаром з використанням технік децимації і згорткових методів для створення оцінки класів з метою регресії і класифікації.

Треба розуміти що загальних знань про структуру архітектури CNN замало і для маніпуляцій з цими моделями потрібно буде витратити деякий час але треба також зважати на складність внесення змін.

2.3 Згортковий шар згорткової нейронної мережі

Можна вважати цей модуль є дуже важливою складовою ЗНН та її функціонуванні. Він налаштовує фокус навколо роботи ядер під час навчання.

Ці ядра можливо знайти на будь-якій глибині вхідної інформації, причиною можна вважати їх мала розмірність. Після потрапляння даних до шару, для отримання карти активації згортковий шар виконує стортання всіх фільтрів через просторову розмірність вхідного сигналу. Після входу скалярний добуток визначається всіх даних у обраному ядрі. (Рисунок 2.3). З цього моменту мережа вчить ядра, які «вистріляють», коли бачать певну особливість у заданому просторовому положенні вхідних даних. Після цього дефектні ядра навчаються мережею після детектування особливих ознак у обраному просторовому положенні вхідних даних. Їх звично називати активаціями.

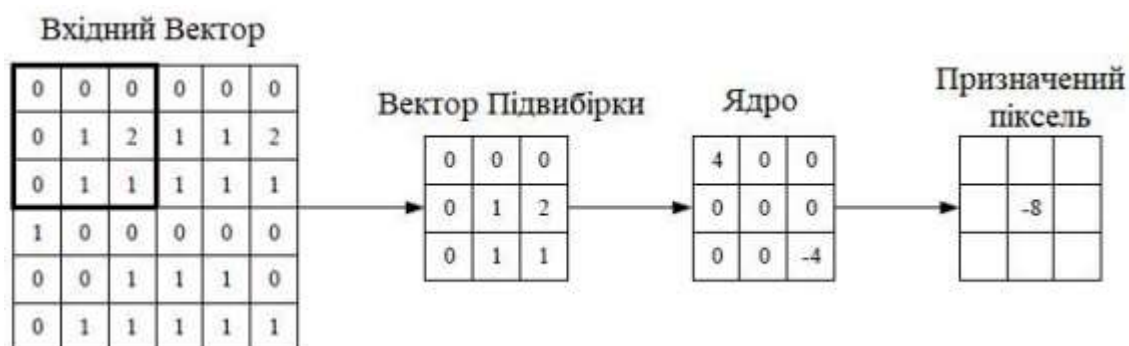


Рисунок 2.3 – Елементарний зразок згорткового шару. Перетворення значення центрального елемента ядра за допомогою обчислення на зважену суму самого себе і сусідніх сигментів.

Карта активації являє собою унікальний ключ для кожного ядра який знаходиться вздовж глибини, щоб вивести всі результуючі дані з шару.

Вже відомо що ШНН не є доцільним використовувати для опрацювання зображень через збільшення моделі до значних розмірів чим знижує якість та швидкість навчання. В наслідок цього використовується метод з максимальним підключенням стандартних нейронів ШНН між собою, для вирішення цієї проблеми була виконана мінімізація підключень до об'єму вхідних даних у згортковому нейроні. Ширина цього підключення прийнято іменувати розміром сприйнятливого поля нейрона. Значення з'єднання через глибину у більшості випадках дорівнює глибині вхідних даних.

Для прикладу візьмемо на вхід до мережі кольорову картинку розміром $64 \times 64 \times 3$ (RGB) і налаштуємо розмір сприйнятливої області як 6×6 , на виході отримаємо 108 ваг на кожен вузол у складі згорткового шару. Для порівняння у інших ШНН стандартний нейрон мав би ваги 12,288 на кожен.

За допомогою оптимізації вхідних даних згорткові шари мають властивість чутливо зменшити складність моделі. Гіперпараметри використані у процесі оптимізації: крок, глибина і встановлення нульових накладок.

Глибина об'єму вихідних даних це гіперпараметр який можливо змінювати вручну через визначення кількості вузлів у складі шару до раніше обраної області вхідних параметрів. Приклади використання можна знайти в інших формах ШНН де зв'язок налаштований безпосередньо між кожним нейроном у прихованому шарі. Зменшення величини цього гіперпараметра дозволяє сильно зменшити загальну кількість вузлів мережі, але це має прямий зв'язок з якістю і можливостями розпізнавання образів моделі.

Є можливість визначити другий гіперпараметр що відповідає за визначення глибини навколо просторової розмірності вхідних параметрів та розміщення сприйнятливої області. Наприклад, для того щоб отримати дуже великі активації та закрити сприйнятливе поле треба встановити значення кроку один. При збільшенні даної величини зведе до мінімуму кількість перекриттів і відобразиться у зменшенні отриманих просторових розмірів.

Нульова накладка - це простий процес прокладки межі вхідних параметрів і є ефективним методом для подальшого контролю розмірності об'єму вихідних даних.

Треба зазначити що просторова розмірність вихідних параметрів згорткових шарів зазнає змін після кожного використання наведених методів.

Для визначення нульової накладки необхідно скористатися нижче наведеною формулою:

$$\frac{(V - R) + 2Z}{S + 1}$$

де V - розмір об'єму вхідних параметрів (глибина×ширина×висота), R - розмір приймального поля, Z - кількість набору нульових накладок, а S - крок. У випадку отримання не цілого числа найчастіша помилка це невірно визначений крок, що призводить до нерівномірного і хибного розміщення вузлів до вказаного входу.

Нажаль на сьогоднішній день цих досягнень замало тому що зображення будь-якої реальної розмірності необхідно опрацьовувати величезною моделлю. В наслідок чого були винайдені методи для значного скорочення загальної кількості параметрів всередині згорткового шару.

У основі роботи обміну параметрами є припущення на рахунок якщо одна функція регіону дала позитивний результат для обчислення в заданій просторовій області то є велика ймовірність що позитивний результат буде відтворено в іншій області. У випадку встановлення обмеження для карт активації в межах однакової величини вагів та зсувів, окрема група параметрів втратить свою актуальність які були створенні у згортковому шарі.

Результатом стало отримання загальних градієнтів від кожного нейрона після стадії зворотного поширення які дають можливість підсумовувати по всій глибині при цьому оновлення буде виконуватися тільки для одного набору вагів а не для кожного окремо.

Ідея підходу ЗНМ полягає у належному базуванні функції моделі на основах локального розуміння зображення. Вона має особливість повторно використовувати один параметр декілька разів що зменшує загальну кількість параметрів у порівнянні зі стандартною повністю підключеною мережею. У випадку з стандартною мережею генерація вагів опрацьовується для кожного пікселя на зображенні, згорткова нейронна мережа створює і використовує тільки необхідну кількість ваг для сканування тільки певного фрагмента у будь-який час.

Це дуже позитивно відображається на навчальному процесі адже зі зменшенням параметрів використовуваних у мережі тим більше підвищується швидкість і якість.

На початку навчання ЗНМ аналізує базові елементи для пошуку функції. Наприклад, людина ідентифікує предмети за істотними особливостями а машина в свою чергу буде перевіряти кривизну меж цих особливостей. Для цього необхідно обрати шлях генерування обмежувальних блоків які вміщують в себе наявний об'єкт і клас у обмежувальних блоках. Також є інші способи на які дозволяють малювати контури об'єкта визначаючи форму: сегментація екземплярів, підмножина сегментації зображень,

Класифікація зображень наразі дуже актуальна тема тому що більшість застосунків використовують глибинні нейронні мережі. ЗНМ гарно реалізовані як автопілот у автомобілях який у постійному режимі аналізує величезні об'єми даних відео та фото файлів.

ЗНМ також дуже гарно використовується для сканування смаків своїх користувачів на фото у соціальних мережах, інтернет магазинах, розважально-торгівельних центрах.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Формування вхідного математичного опису

Спочатку необхідно здійснити попереднє навчання згорткової нейронної мережі на великому обсязі навчальних даних за допомогою технології перенесення знань (Transfer Learning). Для цього було використані COCO, KITTI, Open Images Dataset бази даних які містять мільйони тренувальних та сотні тисяч тестових зображень, розбитих на різні категорії (рис. 3.1).



Рисунок 3.1 – Колаж фотографій з бази COCO

У даному випадку навчальний набір з різних баз в першу чергу передаються до ЗНМ для її навчання, позначаючи самі прості і дуже обширні ознаки зображення, що дає більш якісний та швидший результат на відміну від пошуку серед голих градієнтів. Після виконання цієї операції згорткова нейронна мережа отримує властивість виділяти вектор ознак на кожному

зображенні для удосконалення детектора об'єктів через навчання. Всі вище вказані дії проілюстровані на схемі щоб краще зрозуміти роз'яснення.

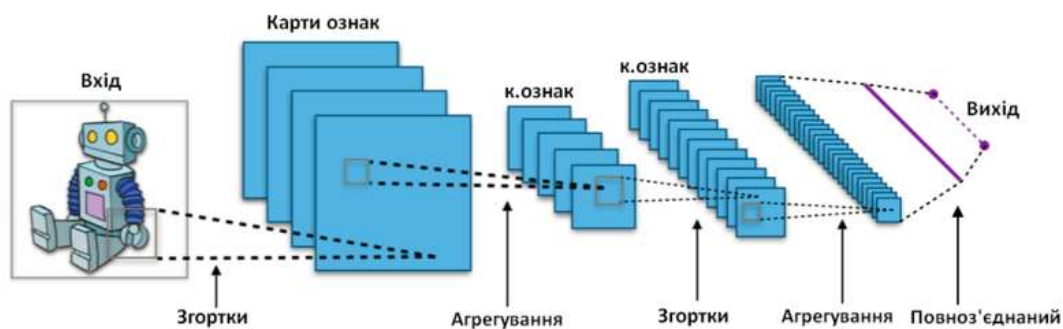


Рисунок 3.2 – Структура моделі навчання з особливими ознаками згорткової мережі

Деякі згорткові шари можуть бути відкореговані для отримання більш точного результату.

Судячи з ілюстрації початок створення навчальних матриць починається з передачі зразків для навчання з баз COCO, KITTI, Open Images у детектор образів для коректної роботи програми розпізнавання об'єктів зацікавленості та імпортування на пристрій.

3.2 Короткий опис програмної реалізації алгоритмів

У ході практичної реалізації мною був створений програмний додаток одного класу використовуючи синтаксис та принципи роботи мови програмування java. В одному з методів відбувається завантаження файлу з навченою моделлю MobileSSD до сховища і повертає шлях до нього. Другий метод виділяє фрейм зображення, постійно слідкуючи за зміною форми об'єкта і його розміру, і відсилає його до нейронної мережі яка у свою чергу ідентифікує і класифікує предмет дослідження. Розробка програмного додатку здійснювалась в середовищі AndroidStudio на мові Java. Для коректного функціонування програми були використані додаткова бібліотека OpenCV.

Один з головних елементів роботи даного алгоритму яка дає можливість працювати з зображеннями та загальними чисельними моделями Застосовується для використання глибокої згорткової нейронної мережі MobileSSD, роботи з камерою та знаходження необхідних параметрів об'єктів.

Основний клас програмного додатку для Android на мові Java приведено в Додатку А. Програма є об'єктно-орієнтованою і складається з одного основного файлу java-класу – `mainActivity.java`.

Клас `MainActivity` реалізовує інтерфейс `CvCameraViewListener2` для захоплення об'єктів під час відеозйомки, отримання параметрів об'єкта, завантаження нейромережі і для подальшої класифікації на основі даних отриманої після ідентифікації предмета.

Таким чином, програма дозволяє перевірити на практиці розроблений алгоритм за допомогою смартфона з метою реалізації в майбутньому повнофункціональної портативної системи пошуку і розпізнавання об'єктів зацікавленості.

3.3 Аналіз отриманих результатів моделювання

При створенні цього програмного проекту була задіяна модель згорткової нейронної мережі. Треба зазначити що обране рішення повністю адаптоване для вирішення поточних задач:

- локальне двохвимірне з'єднання вузлів шару реалізоване через потуповий вхідний сигнал до кожного з нейронів від попереднього локального рецептивного поля;

- інваріантність для зміни параметрів і кількості значень числа вагових коефіцієнтів системи обгрунтована тим що приховані шари мають безліч карт ознак

- мінімізація дозволу для карт ознак досягається за рахунок виконання підвибірки та локального зведення до середнього значення кожним обчислювальним шаром.

Дана модель у нормальному стані представляє собою чергування шарів згортки, шарів субдискретизації та повністю зв'язаних між собою шарів на виході. Вони можуть бути використані у будь-якому порядку з чергуванням.

Для кращого розуміння дамо відповідь що собою являє визначення згортка. У самому звичайному вигляді для цього потрібно дві матриці відповідних розмірів на основі яких буде їх скалярний добуток. Формула буде мати такий вигляд:

$$C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v}$$

де $B_{u,v}$ – значення елемента ядра згортки (u, v),
 $C_{i,j}$ – значення пікселя зображення, що отримаємо,
 $A_{i+u,j+v}$ – значення пікселя вхідного зображення,
 $m_x - 1, m_y - 1$ – розмір ядра згортки.

Рисунок 3.3 – Ілюстрація формули для визначення операції згортки

На представленому прикладі перша матриця порівнюється з другою матрицею так званим фільтром і на виході отримаємо результат їх подібності, тому що зміни значення елемента згортки мають позитивний вплив на коефіцієнт спільних ознак.

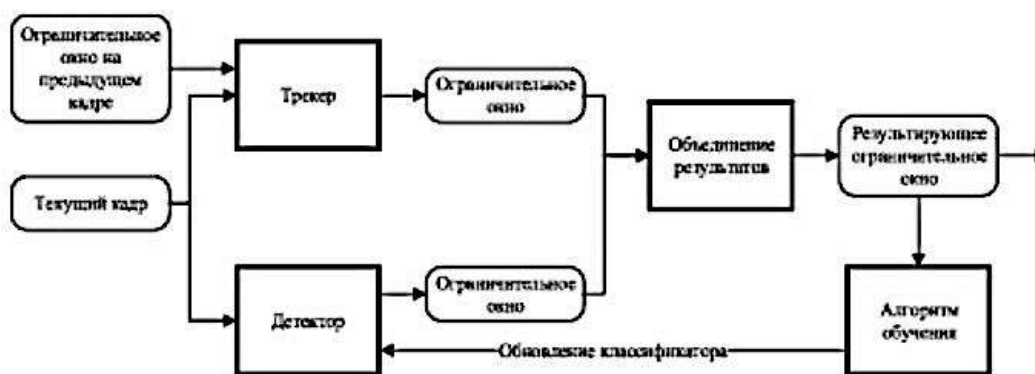


Рисунок 3.4 – Схема роботи алгоритма розпізнавання відеозображення

На рис. 3.4 можна бачити рух інформаційних даних після отримання зображення на вході. За допомогою трекеру та детектору виділяються головні

вектори спостереження за об'єктами зацікавленості після чого виконується аналіз виконаних математичних дій і порівняння з вже раніше відомими патернами.

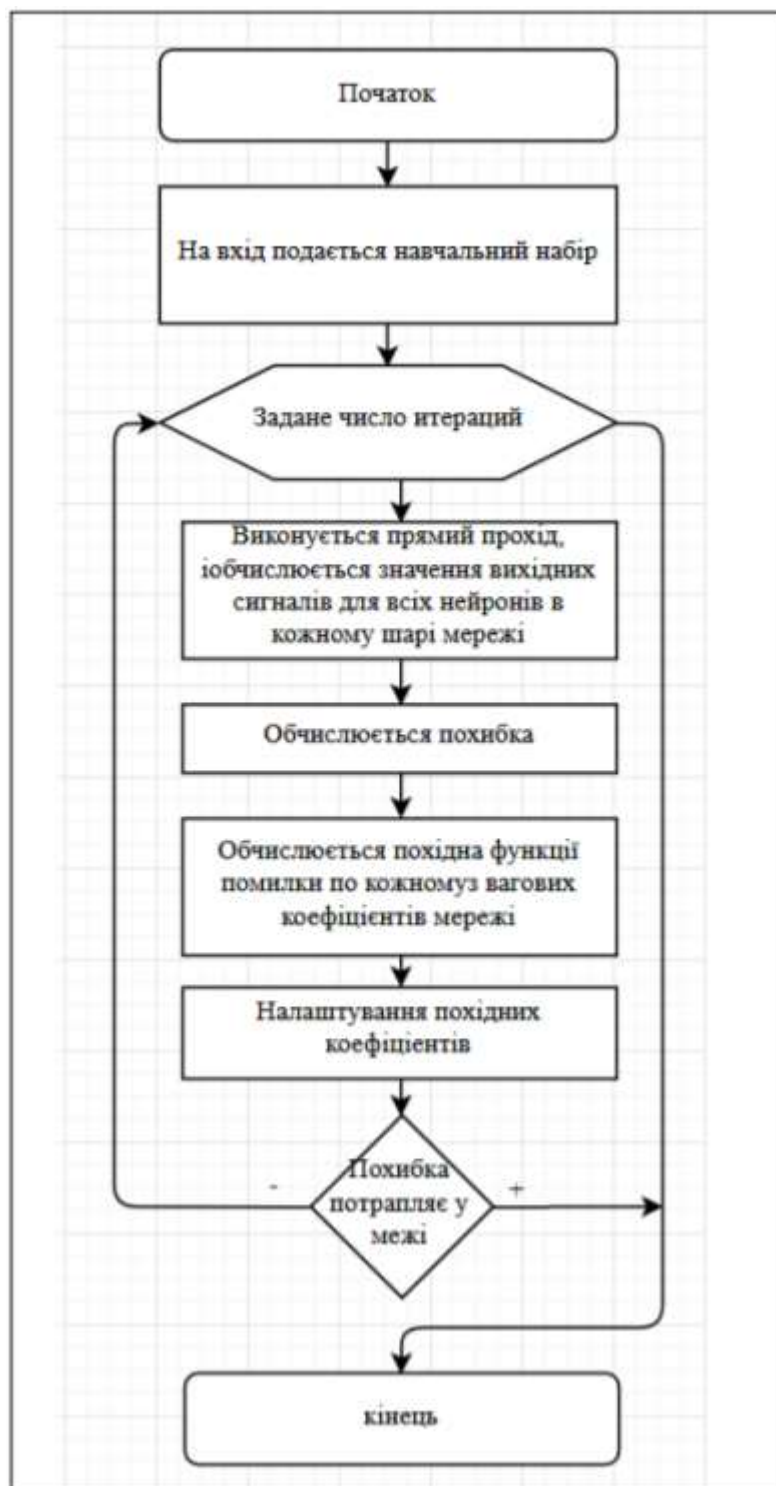


Рисунок 3.5 – Алгоритм зворотного поширення похибки

Важливо також зазначити що навчання були проведено з вчителем на основі алгоритму зворотного поширення похибки, схему якого можна бачити на рис. 3.5. Він представляє з себе ітераційний процес починаючи з самого останнього шару рух відбувається до початкового елементу через всі шари для виявлення всіх помилок та змін ваги якщо це необхідно. Особливістю є те що помилку можна помітити тільки у результаті тобто в кінцевому елементі що і зумовлює ітераційний підхід.



Рисунок 3.6 – Вибіркові зображення з результатом детектування об'єктів з різної відстані: а – стілець; б – телевізор; в – стіл; г – кіт;

Аналіз рис. 3.6 показує, що точність розпізнавання зменшується в залежності від відстані. Хибні результати не були зафіксовані.

Практичний дослід з пошуком та виявленням об'єктів зацікавленості з їх наступною ідентифікацією і класифікацією показав 100% результат що говорить про правильний вибір відповідних інструментів та алгоритмів.

Отже за допомогою ЗНМ було створено вдалий зразок використання комп'ютерного зору при пошуку і розпізнаванню предметів зацікавленості з використанням камери мобільного телефону.

ВИСНОВКИ

Розроблений алгоритм у вигляді програмного додатку для смартфонів на базі Android використовує згорткову нейронну мережу та має практичну реалізацію для ідентифікації та класифікації об'єктів людської праці або живих образів таких як людина та домашні тварини.

Під час практичної реалізації обраної у роботі системи пошуку і розпізнавання об'єктів зацікавленості було вирішено ряд завдань зі створення інформаційного та програмного забезпечення системи:

1) запропоновано вибір згорткової нейронної мережі MobileSSD на основі бази даних ImageNet;

2) обрано модель детектування Single Shot Detector з регресійним аналізом для виділення контурів області в якій вони перебувають на зображенні;

3) програмна реалізація алгоритмів пошуку і розпізнавання об'єктів зацікавленості з бази ImageNet;

4) оптимізовано роботу програмного додатку для роботи з мобільною камерою під час відеоспостереження за об'єктами зацікавленості.

Було зроблено аналіз результатів роботи ЗНМ після її навчання та перевірені усі 20 класів судячи з якого зроблено висновок про високу якість і точність пошуку і розпізнавання предметів та образів.

СПИСОК ЛІТЕРАТУРИ

1. Прэтт У. Цифровая обработка изображений/ Прэтт У – М.: Мир, Кн. 1 и 2, 2001. – 780с.
2. Дзюбаненко А. В. Аналіз методів розпізнавання інтелектуальних систем технічного зору / А. В.Дзюбаненко // Системний аналіз та інформаційні технології: IX міжнар. наук. конф., 26-30 травня 2009 р.: тези доп. – К., 2009. – С. 67.
3. Квасніков В.П., Методи автоматичного захоплення контурів обличчя особи на цифровому зображенні / В.П. Квасніков, Дзюбаненко А. В. // Авиационно-космическая техника и технология. – Харків. – 2009. – № 8(65). – С. 200-205.
4. Дзюбаненко А. В. Обробка зображень в системах технічного зору для засобів охоронного телеспостереження / А. В. Дзюбаненко // Вісник Інженерної академії України. – 2008. – №3-4. – С. 72-75.
5. Ch. M. Bishop, Pattern Recognition and Machine Learning. Springer. – 2006. – С.232-240
6. Y. LeCun and Y. Bengio “Word-level training of a handwritten word recognizer based on convolutional neural networks” in IAPR (Eds), Proc. of the International Conference on Pattern Recognition, II:88-92, IEEE, Jerusalem, October 1994. – С.1-3
7. Довбыш А. С. Информационно-экстремальный метод классификации наблюдений с категориальными признаками / А. С. Довбыш, В. В. Москаленко, А. С. Рыжова // Кибернетика и системный анализ. - 2016. - Т. 52, № 2. - С. 56-63. - Режим доступа: http://nbuv.gov.ua/UJRN/KSA_2016_52_2_9
8. Довбиш, А. С. Основи проектування інтелектуальних систем [Текст] : навч. посібн. / А. С. Довбиш. – Суми : Вид-во СумДУ, 2009. – 171 с.

ДОДАТОК А

```
// -----objCV.java-----
package com.example.objcv;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.res.AssetManager;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

import org.opencv.android.BaseLoaderCallback;
import org.opencv.android.CameraBridgeViewBase;
import org.opencv.android.CameraBridgeViewBase.CvCameraViewFrame;
import org.opencv.android.CameraBridgeViewBase.CvCameraViewListener2;
import org.opencv.android.JavaCameraView;
import org.opencv.android.LoaderCallbackInterface;
import org.opencv.android.OpenCVLoader;
import org.opencv.core.Core;
import org.opencv.core.Mat;
import org.opencv.core.Point;
import org.opencv.core.Scalar;
import org.opencv.core.Size;
import org.opencv.dnn.Net;
import org.opencv.dnn.Dnn;
import org.opencv.imgproc.Imgproc;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
public class MainActivity extends AppCompatActivity implements
CvCameraViewListener2 {

    private static final String[] classNames = "background",
        "plane", "bicycle", "bird", "boat",
        "bottle", "bus", "car", "cat", "chair",
        "caw", "table", "dog", "horse",
        "motorcycle", "human", "plant",
        "sheep`", "sofa", "train", "monitor"};

    private Net net;
    CameraBridgeViewBase mOpenCvCameraView;

    BaseLoaderCallback baseLoaderCallback;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Встановлюємо зчитувач
        mOpenCvCameraView = (JavaCameraView)findViewById(R.id.CamView);
        mOpenCvCameraView.setVisibility(CameraBridgeViewBase.VISIBLE);
        mOpenCvCameraView.setCvCameraViewListener(this);
    }
}
```

```

baseLoaderCallback = new BaseLoaderCallback(this) {
    @Override
    public void onManagerConnected(int status) {
        switch (status) {
            case LoaderCallbackInterface.SUCCESS: {
                Log.v("plan-log", "Loader interface success");
                mOpenCvCameraView.enableView();
                break;
            }
            default: {
                super.onManagerConnected(status);
                break;
            }
        }
    }
};

}
// Завантаження нейромережі
public void onCameraViewStarted(int width, int height) {
    String proto = getPath("MobileSSD.prototxt", this);
    String weights = getPath("MobileSSD.caffemodel", this);
    net = Dnn.readNetFromCaffe(proto, weights);
    Log.v("plan-log", "Network loaded successfully");
}

public Mat onCameraFrame(CvCameraViewFrame inputFrame) {
    final int IN_WIDTH = 299;
    final int IN_HEIGHT = 299;
    final float WH_RATIO = (float)IN_WIDTH / IN_HEIGHT;
    final double IN_SCALE_FACTOR = 0.007842;
    final double MEAN_VAL = 127.4;
    final double THRESHOLD = 0.1999;
    // Отримуємо новий фрейм
    Mat frame = inputFrame.rgba();
    Imgproc.cvtColor(frame, frame, Imgproc.COLOR_RGBA2RGB);
    // Відсилаємо зображення до нейромережі
    Mat blob = Dnn.blobFromImage(frame, IN_SCALE_FACTOR,
        new Size(IN_WIDTH, IN_HEIGHT),
        new Scalar(MEAN_VAL, MEAN_VAL, MEAN_VAL), false, false);
    net.setInput(blob);
    Mat detections = net.forward();
    int cols = frame.cols();
    int rows = frame.rows();
    // З'ясовуємо розмір об'єкта
    detections = detections.reshape(1, (int)detections.total() / 7);
    for (int i = 0; i < detections.rows(); ++i) {
        double confidence = detections.get(i, 2)[0];
        if (confidence > THRESHOLD) {
            int classId = (int)detections.get(i, 1)[0];
            int left = (int)(detections.get(i, 3)[0] * cols);
            int top = (int)(detections.get(i, 4)[0] * rows);
int right = (int)(detections.get(i, 5)[0] * cols);
            int bottom = (int)(detections.get(i, 6)[0] * rows);
            // Малюємо прямокутник навколо об'єкта
            Imgproc.rectangle(frame, new Point(left, top), new
Point(right, bottom),
                new Scalar(0, 255, 0));

```

```

        String label = classNames[classId] + ": " + confidence;
        int[] baseLine = new int[1];
        Size labelSize = Imgproc.getTextSize(label,
Core.FONT_HERSHEY_SIMPLEX, 0.5, 1, baseLine);
        // Малемо фон для поля з назвою об'єкта
        Imgproc.rectangle(frame, new Point(left, top -
labelSize.height),
            new Point(left + labelSize.width, top +
baseLine[0]),
            new Scalar(255, 255, 255),
Imgproc.COLOR_BGR2HLS_FULL);
        // Записуємо назву ідентифікованого класу об'єкта
        Imgproc.putText(frame, label, new Point(left, top),
            Core.FONT_HERSHEY_SIMPLEX, 0.5, new Scalar(0, 0,
0));
    }
}
return frame;
}
public void onCameraViewStopped() {}
// Завантаження файлу до сховища та повернення шляху до нього
private static String getPath(String file, Context context) {
    AssetManager assetManager = context.getAssets();
    BufferedInputStream inputStream = null;
    try {
        // Зчитування даних з assets
        inputStream =
BufferedInputStream(assetManager.open(file));
        byte[] data = new byte[inputStream.available()];
        inputStream.read(data);
        inputStream.close();
        // Створення копії файлу у сховищі
        File outFile = new File(context.getFilesDir(), file);
        FileOutputStream os = new FileOutputStream(outFile);
        os.write(data);
        os.close();
        // Повернення шляху до файлу для зчитування у звичайний
спосіб
        return outFile.getAbsolutePath();
    } catch (IOException ex) {
        Log.v("plan-log", "Failed to upload a file");
    }
    return "";
}
@Override
public void onResume() {
    super.onResume();
    if (!OpenCVLoader.initDebug()) {
        Toast.makeText(getApplicationContext(), "Error after Resume",
Toast.LENGTH_SHORT).show();
    } else {
baseLoaderCallback.onManagerConnected(BaseLoaderCallback.SUCCESS);
    }
}
}
}

```