**МIНIСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Західноукраїнський національний університет**

**Факультет комп’ютерних інформаційних технологій**

Кафедра комп’ютерної інженерії

**Жаборинська Ірина Василівна**

**«Алгоритм детекції облич на основі виділення ключових точок/Algorithm for facial detection based on the selection of key points»**

спеціальність: 123 - Комп’ютерна інженерія

освітньо-професійна програма - Комп’ютерна інженерія

Кваліфікаційна робота

Виконав студент групи КІм-21

І.В. Жаборинська

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Науковий керівник:

к.т.н., доц. Ю.М. Батько

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кваліфікаційну роботу допущено

до захисту:

"\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ р.

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Л. О. Дубчак

**Тернопіль – 2022**

ЗМІСТ

[Вступ 7](#_Toc120025779)

[1 Програмні системи аналізу та обробки цифрових зображень 10](#_Toc120025780)

[1.1 Цифрова обробка зображень 10](#_Toc120025781)

[1.2 Цифрові зображення, класифікація та основні характеристики 20](#_Toc120025782)

[1.3 Програмні засоби для написання програмних кодів 25](#_Toc120025783)

[1.4 Постановка задач дослідження 29](#_Toc120025784)

[1.5 Висновки до розділу 30](#_Toc120025785)

[2 Методи та алгоритми виділення об’єктів на цифрових зображеннях 31](#_Toc120025786)

[2.1 Методи та алгоритми виділення об’єктів на цифрових 31](#_Toc120025787)

[2.2 Алгоритми детекції людських облич на цифрових зображеннях 41](#_Toc120025788)

[2.3 Алгоритми детекції облич на основі виділення ключових точок 47](#_Toc120025789)

[2.4 Висновки до розділу 50](#_Toc120025790)

[3 Програмний додаток детекції людських облич на основі виділення ключових точок 51](#_Toc120025791)

[3.1 Структура програмного додатку детекції людських облич 51](#_Toc120025792)

[3.2 Програмні модулі додатку детекції облич 64](#_Toc120025793)

[3.3 Тестування та аналіз реалізованого програмного додатку 67](#_Toc120025794)

[3.4 Висновки до розділу 72](#_Toc120025795)

[Висновки 73](#_Toc120025796)

[Список використаної літератури 74](#_Toc120025797)

[Додаток А Лістинг програмного модуля детекції облич 77](#_Toc120025798)

[Додаток Б Світлокопії виданих публікацій 79](#_Toc120025799)

# Вступ

Актуальність роботи. Сьогодні зображення та відео повсюди. На онлайн-сайтах для обміну фотографіями та в соціальних мережах їх мільярди. У галузі досліджень зору домінували машинне навчання та статистика. Використання зображень і відео для виявлення, класифікації та відстеження об’єктів або подій, щоб «розуміти» сцену реального світу. Програмування комп’ютера та розробка алгоритмів для розуміння того, що на цих зображеннях, є сферою комп’ютерного зору. Комп’ютерне зір забезпечує такі програми, як пошук зображень, робота-навігація, аналіз медичних зображень, керування фотографіями та багато іншого. З точки зору комп’ютерного зору, зображення – це сцена, що складається з цікавих об’єктів і фону, представленого всім іншим на зображенні. Відносини та взаємодії між цими об’єктами є ключовими факторами для розуміння сцени.

Обробка зображення – це метод виконання деяких операцій із зображенням, щоб отримати покращене зображення або витягти з нього деяку корисну інформацію. Це тип обробки сигналу, у якому входом є зображення, а виходом може бути зображення або характеристики/функції, пов’язані з цим зображенням. Сьогодні обробка зображень є однією з технологій, що швидко розвиваються. Це також є основною дослідницькою сферою в інженерних та комп’ютерних дисциплінах.

Обробка зображення в основному включає наступні три етапи:

* імпортування зображення за допомогою інструментів отримання зображення;
* аналіз і обробка зображення;
* вихід, результатом якого може бути змінене зображення або звіт, який базується на аналізі зображення.

Існує два типи методів обробки зображень, а саме аналогова та цифрова обробка зображень. Для друкованих копій, таких як роздруківки та фотографії, можна використовувати аналогову обробку зображень. Аналітики зображень використовують різні основи інтерпретації, використовуючи ці візуальні методи. Методи обробки цифрових зображень допомагають маніпулювати цифровими зображеннями за допомогою комп’ютерів. Три загальні фази, які мають пройти всі типи даних під час використання цифрової техніки, це попередня обробка, покращення та відображення, вилучення інформації. Тому задача створення алгоритму детекції людських облич на основі виділення ключових точок є актуальною.

Метою роботи є розробка алгоритму детекції людських облич на основі виділення ключових точок.

Для досягнення даної мети ставились наступні завдання:

* провести класифікацію задач обробки цифрових зображень;
* проаналізувати існуючі формати зберігання цифрових зображень;
* провести аналітичний огляд наявних програмних засобів детекції об’єктів на цифрових зображеннях;
* проаналізувати існуючі методи та алгоритми аналізу та обробки цифрових зображень;
* розробити алгоритм детекції облич на основі виділення ключових точок;
* реалізувати програмний додаток детекції облич на цифрових зображеннях, провести її тестування та порівняти з програмами-аналогами.

Об’єкт дослідження – процес обробки цифрових зображень.

Предмет дослідження – методи і алгоритми детекції об’єктів на цифрових зображеннях.

Наукова новизна одержаних результатів визначається наступним чином:

* проведено комплексний аналіз та класифікацію алгоритмів виділення людських облич на цифрових зображеннях, що надало можливість підкреслити їх переваги та недоліки, а також розробити власний алгоритм детекції людських облич;
* розроблено алгоритм детекції людських облич на основі виділення ключових точок, що дозволило зменшити час опрацювання цифрових зображень та підвищити точність детекції облич на зображення зі складною та насиченою компоновкою сцени.

Практична цінність одержаних результатів полягає в тому, що:

* розроблено та проведено теоретичне дослідження програмного додатку виділення людських облич на основі знаходження ключових точок, що дозволило в подальшому програмно реалізувати та провести дослідження розроблених алгоритмів;
* реалізовано програмне забезпечення для детекції людських облич на основі виділення ключових точок та з використанням алгоритмів цифрової обробки зображень.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційні роботі, підготовлено тези доповіді «Аналіз алгоритмів попередньої обробки зображень систем автоматизованого моніторингу» обсягом 1 сторінка на VІ Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі», а також «Аналіз алгоритмів сегментації для систем автоматизованого аналізу зображень» обсягом 1 сторінка на VІ Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі».

1 Програмні системи аналізу та обробки цифрових зображень

## Цифрова обробка зображень

Обробка цифрових зображень – це клас методів обробки цифрових зображень за допомогою комп’ютерних алгоритмів. Це важливий етап попередньої обробки в багатьох програмах, таких як розпізнавання обличчя, виявлення об’єктів і стиснення зображень.

Обробка зображення виконується, щоб покращити наявне зображення або відсіяти з нього важливу інформацію. Це важливо в кількох додатках компютерного зору на основі глибокого навчання, де така попередня обробка може значно підвищити продуктивність моделі. Маніпулювання зображеннями, наприклад, додавання або видалення об’єктів до зображень, є іншим застосуванням, особливо в індустрії розваг.

Основні етапи будь-якого типового конвеєра цифрової обробки зображень містять такі етапи (дані етапи неє обов’язковими проте вони дозволяють описати усі можливі операції над цифровими зображеннями):

1. Отримання зображення. Зображення фіксується камерою та оцифровується (якщо вихідні дані камери не оцифровуються автоматично) за допомогою аналого-цифрового перетворювача для подальшої обробки в комп’ютері.

2. Покращення зображення. На цьому етапі отримане зображення змінюється відповідно до вимог конкретного завдання, для якого буде використано зображення. Ці методи в першу чергу спрямовані на виділення прихованих або важливих деталей зображення, як-от регулювання контрастності та яскравості тощо. Поліпшення зображення має дуже суб’єктивний характер.

3. Відновлення зображення. Цей крок пов’язаний із покращенням зовнішнього вигляду зображення та є об’єктивною операцією, оскільки погіршення якості зображення можна віднести до математичної або ймовірнісної моделі. Наприклад, видалення шумів або розмитості зображень.

4. Обробка кольорових зображень. Цей крок спрямований на обробку кольорових зображень (16-бітних зображень RGB або RGBA), наприклад, на виконання корекції кольорів або моделювання кольорів у зображеннях.

5. Вейвлети та обробка з різною роздільною здатністю. Вейвлети є будівельними блоками для представлення зображень із різним рівнем роздільної здатності. Зображення послідовно діляться на менші області для стиснення даних і для пірамідального представлення.

6. Стиснення зображення. Для передачі зображень на інші пристрої або через обмеження обчислювальної пам’яті зображення потрібно стиснути, і їх неможливо зберегти в початковому розмірі. Це також важливо для відображення зображень через Інтернет; наприклад, у Google невелика мініатюра зображення є сильно стисненою версією оригіналу. Тільки коли ви клацаєте зображення, воно відображається в оригінальній роздільній здатності. Цей процес економить пропускну здатність на серверах.

7. Морфологічна обробка. Компоненти зображення, які є корисними для представлення та опису форми, потрібно витягти для подальшої обробки або подальших завдань. Морфологічна обробка надає інструменти (які, по суті, є математичними операціями) для цього. Наприклад, операції розмивання та розширення використовуються для збільшення різкості та розмивання країв об’єктів на зображенні відповідно.

8. Сегментація зображення. Цей крок передбачає поділ зображення на різні ключові частини, щоб спростити та/або змінити подання зображення на щось більш значуще та легше для аналізу. Сегментація зображення дозволяє комп’ютерам зосереджувати увагу на більш важливих частинах зображення, відкидаючи решту, що дозволяє автоматизованим системам покращити продуктивність.

9. Представлення та опис. Процедури сегментації зображення, як правило, супроводжуються цим кроком, де завдання для представлення полягає в тому, щоб вирішити, чи потрібно сегментовану область зображувати як межу чи повну область. Опис має справу з виділенням атрибутів, які призводять до певної кількісної інформації, що представляє інтерес, або є основними для диференціації одного класу об’єктів від іншого.

10. Виявлення та розпізнавання об'єктів. Після того, як об’єкти сегментовані із зображення та завершено етапи представлення та опису, автоматизована система повинна призначити об’єкту мітку, щоб користувачі могли знати, який об’єкт було виявлено, наприклад, «дерево» або «людина».

11. База знань. Знання можуть бути такими ж простими, як координати обмежувальної рамки для цікавого об’єкта, знайденого на зображенні, разом із присвоєною йому міткою об’єкта. У базу знань можна закодувати все, що допоможе розв’язати задачу для конкретного завдання.

На зображенні 1.1 проілюстровано етапи обробки цифрових зображень.

Отримання зображення

Покращення зображення

Відновлення зображення

Обробка кольорових зображень

Вейвлети та обробка з різною роздільною здатністю

Стиснення зображення

Морфологічна обробка

Сегментація зображення

Представлення та опис

Виявлення та розпізнавання об'єктів

Рисунок 1.1 – Основні етапи роботи з цифровими зображеннями

Обробку зображень можна використовувати для покращення якості зображення, видалення небажаних об’єктів із зображення або навіть створення нових зображень з нуля. Наприклад, за допомогою обробки зображень можна видалити фон із зображення людини, залишивши лише об’єкт на передньому плані.

Обробка зображень – це величезна та складна галузь із багатьма різними алгоритмами та методами, які можна використовувати для досягнення різних результатів. У цьому розділі ми зосередимося на деяких із найпоширеніших завдань обробки зображень і способах їх виконання.

Покращення зображення. Одним із найпоширеніших завдань обробки зображень є покращення зображення або покращення якості зображення. Він має ключове застосування в задачах комп’ютерного зору, дистанційного зондування та спостереження. Одним із поширених підходів є налаштування контрастності та яскравості зображення.

Контраст – це різниця в яскравості між найсвітлішою та найтемнішою ділянками зображення. Збільшуючи контрастність, можна збільшити загальну яскравість зображення, що полегшує його перегляд. Яскравість – це загальна освітленість або темність зображення. Збільшуючи яскравість, зображення можна зробити світлішим, що полегшить його перегляд. Як контрастність, так і яскравість можна налаштувати автоматично за допомогою більшості програм для редагування зображень або їх можна налаштувати вручну.

Однак налаштування контрастності та яскравості зображення є елементарними операціями. Іноді зображення з ідеальним контрастом і яскравістю при підвищенні масштабу стає розмитим через меншу кількість пікселів на квадратний дюйм (щільність пікселів). Щоб вирішити цю проблему, використовується відносно нова та набагато вдосконаленіша концепція супер-роздільності зображення, згідно з якою зображення високої роздільної здатності отримують із аналогів із низькою роздільною здатністю. Для цього широко використовуються методи глибокого навчання. Приклад покращення роздільної здатності наведено на рисунку 1.2.

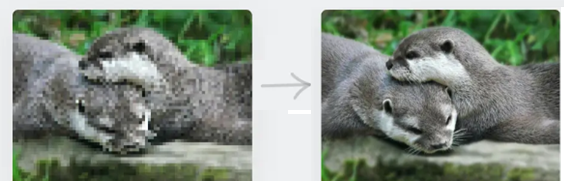


Рисунок 1.2 – Приклад підвищення роздільної здатності цифрового зображення

Відновлення зображення. Якість зображень може погіршитися з кількох причин, особливо фотографій з епохи, коли хмарне зберігання не було таким звичним явищем. Наприклад, на зображеннях, відсканованих із друкованих копій, зроблених за допомогою старих миттєвих камер, часто з’являються подряпини. Приклад реставрації цифрового зображення наведено на рисунку 1.3.

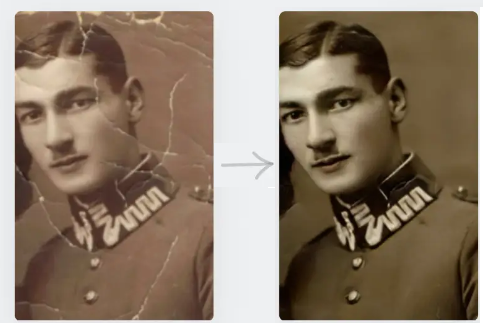


Рисунок 1.3 – Приклад реставрації цифрового зображення

Реставрація зображень особливо захоплююча, оскільки передові методи в цій галузі можуть потенційно відновити пошкоджені історичні документи. Потужні алгоритми відновлення зображень на основі глибокого навчання можуть виявити великі фрагменти відсутньої інформації з розірваних документів.

До цієї категорії, наприклад, відноситься малювання зображень, і це процес заповнення відсутніх пікселів у зображенні. Це можна зробити за допомогою алгоритму синтезу текстур, який синтезує нові текстури для заповнення відсутніх пікселів. Однак моделі на основі глибокого навчання є фактичним вибором через їхні можливості розпізнавання образів.

Сегментація зображення. Сегментація зображення – це процес поділу зображення на кілька сегментів або областей. Кожен сегмент представляє інший об’єкт на зображенні, і сегментація зображення часто використовується як етап попередньої обробки для виявлення об’єкта.

Існує багато різних алгоритмів, які можна використовувати для сегментації зображення, але одним із найпоширеніших підходів є використання порогових значень. Двійкове порогове визначення, наприклад, є процесом перетворення зображення у двійкове зображення, де кожен піксель є чорним або білим. Порогове значення вибирається таким чином, що всі пікселі з рівнем яскравості нижче порогового значення стають чорними, а всі пікселі з рівнем яскравості вище порогового значення стають білими. Це призводить до того, що об’єкти на зображенні сегментуються, оскільки тепер вони представлені чіткими чорними та білими областями. Приклад використання порогової сегментації цифрового зображення з звикористанням одного порогу наведено на рисунку 1.4.

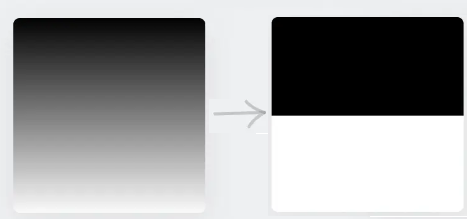


Рисунок 1.4 – Приклад порогової сегментації цифрового зображення представленого в градаціях сірого з порогом *Т*=127

У багаторівневому пороговому значенні, як випливає з назви, різні частини зображення перетворюються на різні відтінки сірого залежно від кількості рівнів. Наприклад, використовувалося багаторівневе визначення порогів для медичної візуалізації – зокрема, для сегментації МРТ головного мозку.

Сучасні методи використовують автоматизовані алгоритми сегментації зображень із використанням глибокого навчання як для бінарних, так і для задач сегментації з кількома мітками. Наприклад, PFNet або Positioning and Focus Network – це модель на основі CNN, яка вирішує проблему сегментації замаскованих об’єктів. Він складається з двох ключових модулів: модуля позиціонування (PM), призначеного для виявлення об’єктів (імітує хижаків, які намагаються визначити грубе положення здобичі); і модуль фокусування (FM), призначений для виконання процесу ідентифікації в хижацтві для уточнення початкових результатів сегментації шляхом фокусування на неоднозначних областях. Архітектура моделі PFNet показана нижче.

Виявлення об’єктів. Виявлення об’єктів – це завдання ідентифікації об’єктів на зображенні, яке часто використовується в таких програмах, як безпека та спостереження. Для виявлення об’єктів можна використовувати багато різних алгоритмів, але найпоширенішим підходом є використання моделей глибокого навчання, зокрема згорткових нейронних мереж (CNN). CNN – це різновид штучної нейронної мережі, яка була спеціально розроблена для завдань обробки зображень, оскільки операція згортання в їхньому ядрі допомагає комп’ютеру «бачити» фрагменти зображення відразу, замість того, щоб мати справу з одним пікселем за раз. CNN, навчені для виявлення об’єктів, виведуть обмежувальну рамку (як показано на ілюстрації вище), що зображує місце, де виявлено об’єкт на зображенні разом із його міткою класу.

Прикладом такої мережі є популярна модель Faster R -CNN ( Region-based C onvolutional N neural N network ), яка є наскрізною повністю згорточною мережею, яку можна навчати. Модель Faster R-CNN чергує точне налаштування для завдання пропозиції щодо регіону (передбачення областей на зображенні, де може бути присутній об’єкт) і потім точне налаштування для виявлення об’єкта (визначення наявного об’єкта), зберігаючи пропозиції фіксованими. Нижче наведено архітектуру та деякі приклади регіональних пропозицій.

Стиснення зображення. Стиснення зображення – це процес зменшення розміру файлу зображення, при цьому намагаючись зберегти якість зображення. Це робиться для економії місця для зберігання, особливо для запуску алгоритмів обробки зображень на мобільних і крайніх пристроях, або для зменшення пропускної здатності, необхідної для передачі зображення.

Традиційні підходи використовують алгоритми стиснення з втратами, які працюють, дещо знижуючи якість зображення, щоб досягти меншого розміру файлу. Формат файлу JPEG, наприклад, використовує дискретне косинусне перетворення для стиснення зображення.

Сучасні підходи до стиснення зображень передбачають використання глибокого навчання для кодування зображень у низьковимірний простір функцій і подальшого відновлення цього на стороні приймача за допомогою мережі декодування. Такі моделі називаються автокодерами , які складаються з гілки кодування, яка вивчає ефективну схему кодування, і гілки декодера, яка намагається відновити зображення без втрат із закодованих функцій.

Маніпулювання зображеннями. Маніпулювання зображенням – це процес зміни зображення для зміни його зовнішнього вигляду. Це може знадобитися з кількох причин, наприклад, видалення небажаного об’єкта із зображення або додавання об’єкта, якого немає на зображенні. Графічні дизайнери часто роблять це для створення плакатів, фільмів тощо.

Прикладом маніпулювання зображеннями є Neural Style Transfer, який є технікою, яка використовує моделі глибокого навчання для адаптації зображення до стилю іншого. Наприклад, звичайне зображення можна перенести в стилі «Зоряної ночі» Ван Гога. Neural Style Transfer також дозволяє штучному інтелекту створювати нові цифрові зображення. Приклад таких цифрових зображень наведено на рисуноку 1.5.



Рисунок 1.5 – Приклад маніпулювання зображеннями в стилі Neural Style Transfer

Генерація зображення. Синтез нових зображень – ще одне важливе завдання в обробці зображень, особливо в алгоритмах глибокого навчання, які потребують великої кількості позначених даних для навчання. Методи створення зображень зазвичай використовують генеративні змагальні мережі (GAN), які є ще однією унікальною архітектурою нейронних мереж .

GAN складаються з двох окремих моделей: генератора, який генерує синтетичні зображення, і дискримінатора, який намагається відрізнити синтетичні зображення від реальних. Генератор намагається синтезувати зображення, які виглядають реалістично, щоб обдурити дискримінатора, а дискримінатор тренується, щоб краще оцінювати, чи є зображення синтетичним чи справжнім. Ця суперницька гра дозволяє генератору створювати фотореалістичні зображення після кількох ітерацій, які потім можна використовувати для навчання інших моделей глибокого навчання.

Перетворення із зображення в зображення. Трансляція «зображення в зображення» – клас візуальних і графічних задач, мета яких полягає в тому, щоб навчитися відображати вхідне та вихідне зображення за допомогою навчального набору вирівняних пар зображень. Наприклад, ескіз від руки можна намалювати як вхідні дані, щоб отримати реалістичне зображення об’єкта, зображеного на ескізі, як результат, як показано нижче (рисунок 1.6).



Рисунок 1.6 – Приклад перетворення із зображення в зображення

В даний перелік зібрано основні завдання які вирішують з допомого методів та алгоритмів цифрової обробки зображень. Для вирішення більш складних завдань, дані задачі поєднуються формуючи під час своєї роботи ціли цикли та складні програмно-апаратні системи по обробці зображень. Узагальнений набір задач цирової обробки зоббражень наведено на рисунку 1.7.

Завдання цифрової обробки зображень

Покращення зображення

Відновлення зображення

Генерація зображення

Перетворення із зображення в зображення

Стиснення зображення

Маніпулювання зображеннями

Сегментація зображення

Детекція об’єктів

Рисунок 1.7 – Завдання цифрової обробки зображення

## Цифрові зображення, класифікація та основні характеристики

Щоб стати придатною для цифрової обробки, функція зображення *f(x,y)* повинна бути оцифрована як просторово, так і амплітудно. Як правило, для дискретизації та квантування аналогового відеосигналу використовується пристрій захоплення кадрів або дігітайзер. Отже, щоб створити цифрове зображення, потрібно перевести безперервні дані в цифрову форму. Це робиться в два етапи:

* відбір проб;
* квантування.

Частота дискретизації визначає просторову роздільну здатність оцифрованого зображення, а рівень квантування визначає кількість рівнів сірого в оцифрованому зображенні. Величина вибірки зображення виражається як цифрове значення під час обробки зображення. Перехід між неперервними значеннями функції зображення та її цифровим еквівалентом називається квантуванням.

Кількість рівнів квантування має бути достатньо високою для сприйняття людиною тонких деталей затінення зображення. Поява хибних контурів є основною проблемою для зображення, квантованого з недостатнім рівнем яскравості.

Цифрові зображення інтерпретуються комп’ютером як 2*D* або 3*D* матриці, де кожне значення або піксель у матриці представляє амплітуду, відому як «інтенсивність» пікселя (рисунок 1.8). Як правило, користувачі звикли мати справу з 8‑бітними зображеннями, де значення амплітуди коливається від 0 до 255.

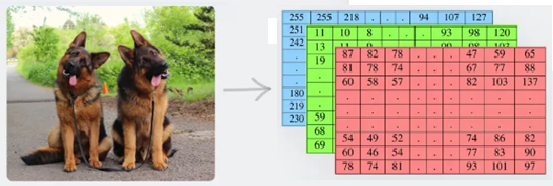


Рисунок 1.8 – Представлення цифрового зображення в вигляді матриць

Таким чином, комп’ютер «бачить» цифрові зображення як функцію: *I(x, y)* або *I(x, y, z)*, де « *I* » – інтенсивність пікселів, а *(x, y)* або *(x, y, z)* представляють координати (для двійкових зображень/зображень у градаціях сірого або RGB відповідно) пікселя на зображенні (рисунок 1.9).



Рисунок 1.9 – Представлення цифрового зображення в вигляді точок

Комп’ютери мають справу з різними «типами» зображень на основі їх представлення функцій. Давайте розглянемо їх далі.

1. Бінарне зображення. Зображення, які мають лише два унікальних значення інтенсивності пікселів - 0 (що позначає чорний колір) і 1 (позначає білий колір), називаються бінарними зображеннями. Такі зображення, як правило, використовуються для виділення дискримінаційної частини кольорового зображення. Наприклад, він зазвичай використовується для сегментації зображення, як показано на рисунку 1.10.

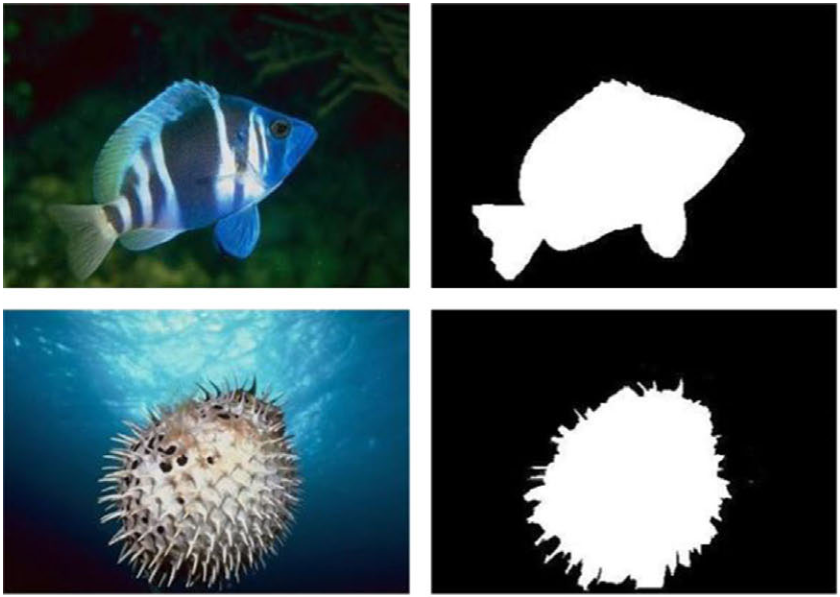


Рисунок 1.10 – Приклад бінарного зображення

2. Зображення в градаціях сірого. Відтінки сірого або 8-бітні зображення складаються з 256 унікальних кольорів, де інтенсивність пікселя 0 означає чорний колір, а інтенсивність пікселя 255 – білий колір. Усі інші 254 значення між ними є різними відтінками сірого.

Нижче наведено приклад зображення RGB, перетвореного на його версію у відтінках сірого (рисунок 1.11). Закважимо, що форма гістограми залишається незмінною для зображень RGB і зображень у градаціях сірого.



Рисунок 1.11 – Приклад зображення в градація сірого

3. Кольорове зображення RGB. Зображення, до яких ми звикли в сучасному світі, це RGB або кольорові зображення, які є 16-розрядними матрицями для комп’ютерів. Тобто для кожного пікселя можливо 65 536 різних кольорів. «RGB» представляє червоний, зелений і синій «канали» зображення.

Досі зображення використовувати лише з одним каналом. Тобто дві координати могли б визначити розташування будь-якого значення матриці. Тепер три матриці однакового розміру (так звані канали), кожна зі значеннями в діапазоні від 0 до 255, накладаються одна на одну, і тому нам потрібні три унікальні координати для визначення значення елемента матриці.

Таким чином, піксель у зображенні RGB буде чорного кольору, якщо значення пікселя дорівнює (0, 0, 0), і білого кольору, якщо воно дорівнює (255, 255, 255). Будь-яка комбінація чисел між ними породжує всі різноманітні кольори, існуючі в природі. Наприклад, (255, 0, 0) – це червоний колір (оскільки для цього пікселя активовано лише червоний канал). Подібним чином визначається (0, 255, 0) – зелений, а (0, 0, 255) – синій канали.

Нижче наведено приклад зображення RGB, розділеного на компоненти каналу (рисунок 1.12). При цьому видно, що форми гістограм для кожного з каналів відрізняються.

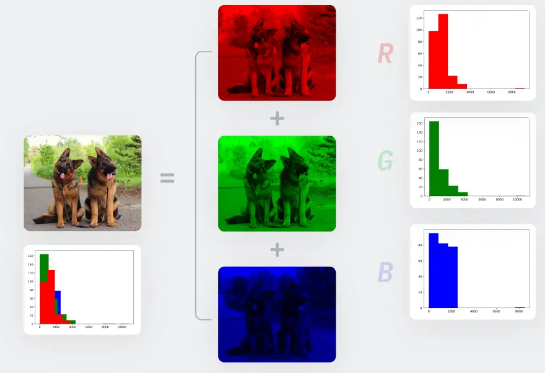


Рисунок 1.12 – Приклад кольорового зображення в форматі RGB

4. Зображення RGBA. Зображення RGBA – це кольорові зображення RGB із додатковим каналом, відомим як «альфа», який відображає непрозорість зображення RGB. Непрозорість коливається від 0% до 100% і, по суті, є «прозорою» властивістю.

Непрозорість у фізиці відображає кількість світла, яке проходить через об'єкт. Наприклад, целофановий папір прозорий (100% непрозорість), матове скло – напівпрозоре, а дерево – непрозоре. Альфа-канал у зображеннях RGBA намагається імітувати цю властивість фізичних тіл. Приклад цього проілюстровано на рисунку 1.13.

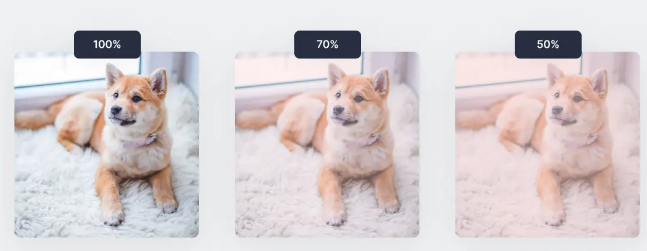


Рисунок 1.13 – Приклад зміни параметра «альфа» в зображеннях RGBA

Серед основних кількісних параметрі цифрового зображення слід виділити наступний прелік:

* Яскравість зображення.
* Контраст зображення.
* Гістограма зображення.
* Рівень зашумленості зображення.

Яскравість зображення іншими словами розподіл градації сірого, це зміна світла та відтінку кольору, зазвичай вираженого від чорного до білого від 0% до 100%.

Контраст зображення – співвідношення чорного і білого зображення, тобто рівень градієнта від чорного до білого. Чим більше співвідношення, тим більше рівнів градієнта від чорного до білого, і тим багатшими будуть відображення кольорів.

Гістограма зображення. Вказує кількість пікселів із кожним рівнем сірого на зображенні, що відображає частоту кожного рівня сірого на зображенні. Форма зберігання зображення в комп'ютері схожа на матрицю з багатьма точками. Ці точки акуратно розташовані в рядках і стовпцях. Значення в кожній точці – це значення сірого зображення, а гістограма – це значення сірого для кожного відтінку в цій точці. Кількість входжень у матрицю.

Рівень зашумленості зображення. Як і слух, коли навколо багато шуму, це вплине на нашу здатність почути вміст. Подібним чином, для зображень ми можемо чітко бачити зображення, але іноді на зображенні є деякі шаблони, які нам не потрібні, тому ми не можемо чітко бачити зображення, це шум зображення.

## Програмні засоби для написання програмних кодів

Програмне забезпечення для розпізнавання зображень допомагає ідентифікувати об’єкти, людей, місця, записи та дії на зображеннях або відео. Такі програми використовують технологію нейронної мережі, яка обробляє всі пікселі, які є частиною зображення.

Інформацію, отриману програмним забезпеченням для розпізнавання фотографій, можна використовувати для багатьох цілей. Це допомагає краще зрозуміти клієнтів та їхні інтереси, а також розробити цільову рекламу для певної групи людей. Це дозволяє роздрібним торговцям створювати оголошення, які відповідають потребам і захопленням їх аудиторії.

* Google Image Recognition – розпізнавання об'єктів у 2 кліки.
* Amazon Rekognition – розпізнавайте знаменитостей.
* Clarifai – допомагає федеральним і комерційним організаціям.
* Google Vision AI – виявляє емоційні сигнали на обличчях.
* LogoGrab – розпізнавання логотипів і знаків.
* Imagga – класифікує зображення.
* IBM Image Detection – використовує першокласну технологію.
* Filestack Processing API – і тегрується зі службами обміну файлами.
* GumGum – чудово підходить для цифрових маркетологів і графічних дизайнерів.

Усе програмне забезпечення для виявлення зображень у цьому списку розпізнає обличчя за допомогою штучного інтелекту (AI). Завдяки алгоритму виявлення зображень, який також називають класифікатором зображень, програма аналізує вхідне зображення та відображає його вміст. Щоб алгоритм розпізнав вміст зображення, йому потрібно визначити, що робити якщо класи будуть різними.

Google Image Recognition. Google відомий тим, що створив найкраще пошукове рішення. Щоб ще більше вдосконалити його, у 2014 році компанія запустила пошукову службу розпізнавання зображень. Вона переглядає Інтернет у пошуках зображень, які здаються ідентичними тим, які ви завантажили. Ви можете надсилати фотографії у форматі jpeg або png. Як випливає з назви, це програмне забезпечення для розпізнавання зображень дозволяє завантажувати та шукати зображення. Його найвизначнішою особливістю є те, що він такий же зручний, як пошуковий сервіс Google, і має подібні можливості. Він забезпечує чудове охоплення, але не має аналітики в реальному часі, щоб допомогти вам зрозуміти результати.

Amazon Rekognition – це програмне рішення для розпізнавання зображень. Це дозволяє розпізнавати предмети, візерунки та обличчя. Його також можна використовувати для пошуку чи порівняння облич. Він базується на тій самій передовій технології навчання, яка була розроблена дослідниками комп’ютерного зору для Amazon Prime Photos для щоденного аналізу мільярдів фотографій. Він використовує моделі глибокої нейронної мережі для виявлення та позначення об’єктів і візерунків на ваших зображеннях. Він може знайти будь-яке зображення, однак якість пошуку залежить від зображень, які ви надаєте. Це означає, що якщо користувач хочете знайти логотип FixThePhoto, вам спочатку потрібно надати йому тисячі зображень, пов’язаних із FixThePhoto.

Clarifai – це розширений безкоштовний API для розпізнавання зображень, який може додавати теги, упорядковувати та інтерпретувати зображення та відео за допомогою машинного навчання та штучного інтелекту. Завдяки використанню технології комп’ютерного зору це програмне забезпечення для обробки зображень ШІ допомагає федеральним і комерційним організаціям визначати вміст їхніх даних, що допомагає їм легше вирішувати проблеми.

Google Vision AI. За допомогою Google API Cloud Vision ви можете аналізувати зображення за допомогою кількох потужних інструментів, від розпізнавання явного вмісту до ідентифікації емоційних сигналів на обличчі. Завдяки такому широкому набору опцій, це дуже корисний інструмент, який можна налаштувати відповідно до ваших конкретних потреб. Доступно декілька варіантів, від обробки зображень із відкритим кодом до використання попередньо створеної моделі зображення, наданої Google. Користувачі можете вказати джерела ваших зображень, і Vision проаналізує їх і покаже вам деталі. Їх можна використовувати для всіх цілей, від перевірки якості до пошуку потрібних продуктів.

LogoGrab здатний ідентифікувати логотипи, символи та бренди. Це дозволяє вашій організації відстежувати та контролювати графічний медіа-вміст у різних типах середовищ, як-от соціальні мережі, телерадіомовлення та роздрібні веб-сайти. Інтеграція програми в різні платформи відмінна. Користувачі відмічають, що перехід на LogoGrab і високоякісну інформацію, яку він надає, буде легким і приємним.

Imagga API – це автоматизоване рішення для додавання тегів до зображень і керування категоріями, розроблене для роботи з великими обсягами зображень. Imagga вказано як API розпізнавання цифрових зображень. Він містить бібліотеку, яка дозволяє класифікувати активи та керувати метаданими. За допомогою інструментів пошуку/фільтра ви можете знаходити мультимедійні файли та керувати ними. Користувачі також можете створювати звіти та проводити аналіз. Це схоже на інші API для керування цифровими активами, наприклад Box, Airtable або Canto Digital Asset Management. Тим не менш, якщо ви порівняєте Imagga з іншими API управління цифровими активами, які дозволяють безкоштовно розпізнавати зображення онлайн, ви побачите, що він набагато доступніший.

IBM Image Detection. IBM є технологічним супергероєм, і вона випустила одне з найкращих програмних продуктів для розпізнавання зображень. Він може розпізнавати людські обличчя, приблизний вік, стать тощо. Однією з найважливіших особливостей IBM Image Detection є його здатність до навчання. Це означає, що IBM надає вам структуру з високим ступенем регулювання, яку можна налаштувати відповідно до будь-яких потреб.

Filestack Processing API. Filestack Processing API зберігає, стискає та перетворює файли. Крім того, він може автоматично підключатися до служб обміну файлами, таких як Google Drive, Dropbox і Facebook. Він також виконує кілька інших завдань, наприклад, виявляє невідповідний вміст і розпізнає символи. Filestack Processing має кілька інших особливостей. Користувачі можете використовувати його для позначення тегами відео та пошуку фотографій, захищених авторським правом. З його допомогою також можна змінювати розмір, обрізати та повертати зображення.

GumGum.Розробники комп’ютерного зору, які стоять за створенням GumGum, навчилися розпізнавати логотипи, проблеми з емаллю зубів і пошкодження автомобіля. Це програмне забезпечення також може обробляти зображення, що робить його одним із найкращих фоторедакторів AI.

Сьогодні GumGum використовує інструменти штучного інтелекту для розпізнавання зображень, які допомагають менеджерам цифрової реклами порівнювати свою кампанію з зусиллями конкурентів і знаходити найбільш відповідні та корисні способи розміщення рекламних банерів. Це програмне забезпечення не повністю інтегроване з соціальними мережами, але воно ідеально підходить для графічних дизайнерів, які створюють рекламу.

Даний перелік програмних засобів не є остаточним, оскільки задача аналізу та детекції об’єктів на зображенні є широковикористовуваною, а розробкою програмних систем даного типу займаються безліч розробників програмного забезпечення

## 1.4 Постановка задач дослідження

Під час проведених досліджень в даному розділі було наведено результати аналізу задач цифрової обробки зображень, приклади та сфери їх використання. Досліджено структури та принципи кодування даних у вигляді цифрових зображень, а також проведено їх класифікацію. Досліджено програмні засоби для детекції об’єктів на цифрових зображеннях, виділено їх основні функції та структури даних для зберігання, обробки, предачі та аналізу даних представлених у форматі цифрового зображення.

Для досягнення поставленої мети необхідно розв’язати наступні задачі.

* провести класифікацію задач обробки цифрових зображень;
* проаналізувати існуючі формати зберігання цифрових зображень;
* провести аналітичний огляд наявних програмних засобів детекції об’єктів на цифрових зображеннях;
* проаналізувати існуючі методи та алгоритми аналізу та обробки цифрових зображень;
* розробити алгоритм детекції облич на основі виділення ключових точок;
* реалізувати програмний додаток детекції облич на цифрових зображеннях, провести її тестування та порівняти з програмами-аналогами.

## 1.5 Висновки до розділу

Проведено аналіз та класифікацію завдань та основних етапів цифрової обробки зображень, що дозволило виділити основні задачі аналізу зображень при проектуванні системи моніторингу з елементами детекції об’єктів.

Проаналізовано формати зберігання цифрових зображень, що дозволило обрати структури даних для швидкого аналізу та детекції об’єктів на зображеннях.

Проведено аналіз наявних програмних засобів для отримання, редагування та аналізу цифрових зображень, що дозволило виділити основні структурні модулі та встановити інтерфейси обміну даними між окремими частинами програмних систем.

# 2 Методи та алгоритми виділення об’єктів на цифрових зображеннях

## 2.1 Методи та алгоритми виділення об’єктів на цифрових

Виявлення об’єктів – це область комп’ютерного зору, яка займається локалізацією та класифікацією об’єктів, що містяться на зображенні чи відео.

Або, іншими словами, виявлення об’єктів зводиться до малювання обмежувальних рамок навколо виявлених об’єктів, що дозволяє нам знайти їх у певній сцені (або як вони рухаються крізь неї).

Обмежувальна рамка – це уявний прямокутник, накреслений навколо об’єкта, який описує його розташування в просторі. Це найпростіший інструмент, який використовується для завдань виявлення та локалізації об’єктів.

Анотації обмежувальної рамки містять координати з інформацією про те, де знаходиться об’єкт на зображенні чи відео. Вони підходять для об’єктів однакової форми, проектів із низьким об’ємом структур та об’єктів, які не перекриваються.

Класи – це імена об’єктів у наборі даних. Якщо створюється служба для виявлення вм’ятин і подряпин, то відповідно спроектувати так, що ці два записи можна повторно використовувати в нових проектах або ієрархічно розгалужувати у міру зростання ваших даних.

Ось кілька обов’язкових обмежень:

* Класи обмежувальних рамок можна використовувати в різних проектах.
* Створення класу має досить простий інтерфейс, щоб ним могли користуватися нетехнічні користувачі.
* Користувач можете додати принаймні 1 мініатюру для представлення об’єкта.
* Користувач можете додати опис або інструкції для цього класу.

Атрибути – це просто анотаційні теги, які можуть визначати специфічні особливості даного об’єкта.

Багато проектів виявлення об’єктів вимагають від розмічувачів додавати атрибути міток поверх анотації обмежувальної рамки – це допомагає більш детально описати даний об’єкт. Наприклад, зазвичай додають такі атрибути‑мітки, як закритий, усічений та переповнений, що вказує на те, що анотовані об’єкти тісно пов’язані з іншими об’єктами на зображенні.

Різниця між виявленням об'єктів та класифікацією зображень полягає в маштабі отримуваних результатів. Класифікація зображень надсилає ціле зображення через класифікатор (наприклад, глибоку нейронну мережу), щоб він виділив тег. Класифікатори враховують все зображення, але не повідомляють вам, де на зображенні з’являється тег.

Виявлення об’єктів є трохи розширенішим, оскільки воно створює обмежувальну рамку навколо класифікованого об’єкта. Приклади різниці в механізмах роботи та отриманих результатанх наведено на рисунку 2.1.

|  |  |
| --- | --- |
|  |  |
| а) Зображення типу «кіт» | b) Об’єкти типу «кіт», «собака», «собака» |

Рисунок 2.1 – Приклад результатів роботи методів класифікації (а) та виділення(b)

Класифікація має свої переваги – це кращий варіант для тегів, які фактично не мають фізичних меж, наприклад «розмиті» або «сонячні». Однак системи виявлення об’єктів майже завжди перевершують класифікаційні мережі у виявленні об’єктів, які мають матеріальну присутність, таких як автомобіль.

Різниця між виявленням об'єктів та сегментацією зображень полягає в точності отримуваних результатів Сегментація зображення – це процес визначення того, які пікселі класу об’єктів містяться на зображенні.

Семантична сегментація зображення позначатиме всі пікселі, що належать цьому тегу, але не визначатиме межі кожного об’єкта. Натомість виявлення об’єктів не сегментує об’єкт, а чітко визначає розташування кожного окремого екземпляра об’єкта за допомогою рамки.

Поєднання семантичної сегментації з виявленням об’єктів призводить до сегментації екземплярів, яка спочатку виявляє екземпляри об’єктів, а потім сегментує кожен у межах виявлених блоків (відомих у цьому випадку як регіони інтересу).

|  |  |  |
| --- | --- | --- |
|  |  |  |
| а) виділення блоків типу «кінь» | b) виділення сегменту типу «кінь» | с) виділення об’єктів типу «кінь» |

Рисунок 2.2 – Приклад результатів роботи методів виділення (а), сегментації (b) та поєднання сегментації та виділення (с)

Виявлення об’єктів дуже якісно та з високим рівнем точності спрацьовує у випадках опрацювання наступних об’єктів:

* Виявлення об’єктів, які займають від 2% до 60% площі зображення.
* Виявлення об'єктів з чіткими межами.
* Виявлення кластерів об'єктів як 1 елемент.
* Локалізація об'єктів на високій швидкості (>15fps).

Для більш точних та якісних результатів необхідно підбирати методи та алгоритми виділення на основі оцінки обєктів на вхідному цифровому зображенні.

Витягнуті об’єкти – використовуйте сегментацію екземплярів.

Довгі та тонкі предмети, такі як олівець, при виявленні займатимуть менше 10% площі коробки. Це зміщує модель до фонових пікселів, а не до самого об’єкта.

Об’єкти, які фізично відсутні – використовуйте класифікацію

Такі речі на зображенні, як тег «сонячний», «яскравий» або «скошений», найкраще ідентифікувати за допомогою методів класифікації зображень, дозволяючи мережі взяти зображення та визначити, яка функція корелює з цими тегами.

Об’єкти, які не мають чітких меж під різними кутами – використовуйте семантичну сегментацію

Небо, земля чи рослинність на аерофотознімках насправді не мають чітких меж. Семантична сегментація більш ефективна при «малюванні» пікселів, які належать до цих класів. Виявлення об’єктів усе одно сприйматиме «небо» як об’єкт, але з такими об’єктами йому буде набагато складніше.

Об’єкти, які часто закриті – за можливості використовуйте сегментацію екземплярів

Оклюзія обробляється набагато краще в двоетапних мережах виявлення, ніж одноразові підходи. У цій галузі детекторів моделі сегментації екземплярів краще розумітимуть і сегментують закриті об’єкти, ніж прості детектори обмежувальної рамки.

До початку глибокого навчання в 2013 році майже все виявлення об’єктів здійснювалося за допомогою класичних методів машинного навчання. Загальні включали техніку виявлення об’єктів Віола-Джонса, масштабно-інваріантні перетворення ознак (SIFT) і гістограму орієнтованих градієнтів.

Вони могли б виявити низку загальних ознак на зображенні та класифікувати їх кластери за допомогою логістичної регресії, кольорових гістограм або випадкових лісів. Сучасні методи глибокого навчання значно перевершують ці.

Підходи на основі глибокого навчання використовують такі архітектури нейронних мереж, як RetinaNet, YOLO (You Only Look Once), CenterNet, SSD (Single Shot Multibox Detector), регіональні пропозиції (R-CNN, Fast-RCNN, Faster RCNN, Cascade R-CNN) для виявлення ознак об'єкта, а потім ідентифікація в мітках.

Виявлення об’єктів зазвичай поділяють на 2 етапи:

* Одноступеневі детектори предметів.
* Двоступеневі детектори предметів.

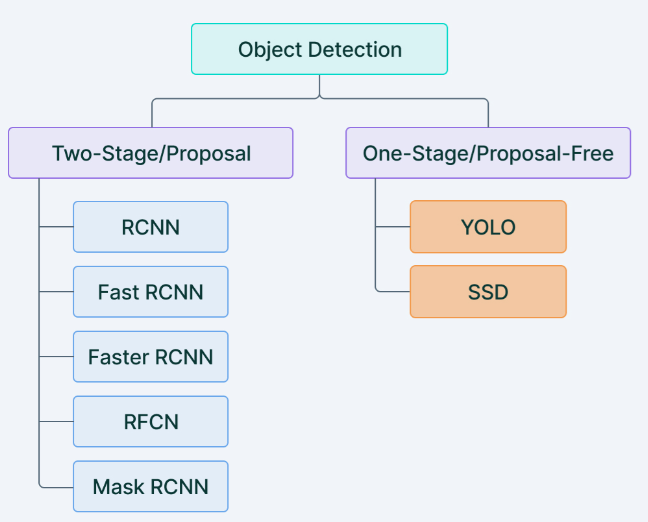


Рисунок 2.3 – Класифікація алгоритмів виділення об’єктів

Сучасні архітектури виявлення об’єктів складаються з 2-етапних архітектур, багато з яких попередньо навчені на наборі даних COCO. COCO – це набір даних зображень, що складається з 90 різних класів об’єктів (автомобілі, люди, спортивні м’ячі, велосипеди, собаки, коти, коні тощо).

Набір даних було зібрано для вирішення типових проблем виявлення об’єктів. Нині воно застаріло, оскільки його зображення були зроблені переважно на початку 2000-х років, що робить їх набагато меншими, зернистішими та з іншими об’єктами, ніж сьогоднішні зображення. Новіші набори даних, такі як OpenImages, займають своє місце де-факто як набір даних для попереднього навчання.

Одноступеневі детектори предметів. Одноступінчастий детектор усуває процес вилучення RоI та безпосередньо класифікує та регресує потенційні опорні блоки. Приклади: сімейство YOLO (YOLOv2, YOLOv3, YOLOv4 і YOLOv5), CornerNet, CenterNet та інші. Наприклад, давайте подивимося, як працює YOLO.

YOLO – це архітектура виявлення об’єктів, яка просто називається YOU ONLY LOOK ONCE. Це передбачає використання єдиної нейронної мережі, навченої наскрізно, для отримання фотографії як вхідних даних і прогнозування обмежувальних рамок і міток класів безпосередньо для кожної обмежувальної рамки. YOLO – типовий однокаскадний сповіщувач.

Двоступінчасті детектори поділяють задачу виявлення об’єктів на два етапи: виділяють RoI (регіон інтересу), потім класифікують і регресують RoI. Приклади архітектур виявлення об’єктів, орієнтованих на 2 етапи, включають R-CNN, Fast-RCNN, Faster-RCNN, Mask-RCNN та інші.

Маска R-CNN є типовою технікою сегментації екземплярів об’єктів для виявлення об’єктів. Ця архітектура є розширенням Faster R-CNN шляхом додавання гілки для прогнозування масок сегментації для кожної RoI паралельно з існуючою гілкою для класифікації та регресії обмежувальної рамки. Гілка маски – це невеликий FCN, застосований до кожної RoI, що передбачає попіксельну маску сегментації. Нижче наведено демонстрацію архітектури Mask R-CNN.

З іншого боку, Faster R-CNN – це модель виявлення об’єктів, яка вдосконалює Fast R-CNN, використовуючи мережу пропозицій регіону ( RPN ) із згенерованими картами об’єктів зі згорткового рівня для оцінки класифікації об’єктів на основі регіону (ROI). об'єднання).

Крім того, Fast R-CNN є вдосконаленою версією R-CNN, яка об’єднує функції CNN незалежно від регіону інтересу (ROI) в одному прямому проході по зображенню. Загалом R-CNN (вибір регіону з функціями CNN) працює повільно, оскільки виконує передачу ConvNet для кожної пропозиції об’єкта без спільного використання обчислень. Тому Fast R-CNN було розроблено для вирішення проблеми повільних обчислень.

Для більшого ознайомлення з принципами роботи різних груп алгоритмів, які використовуються для виявлення об’єктів проведемо їх порівняльний аналіз.

Сімейство моделей R-CNN включає наступне:

1. R-CNN. Тут використовується метод вибіркового пошуку для визначення місцезнаходження RoI у вхідних зображеннях і використовується регіональний класифікатор на основі DCN (Deep Convolutional Neural Network) для незалежної класифікації RoI.
2. SPPNet і Fast R-CNN – це вдосконалена версія R-CNN, яка займається вилученням ROI із карт функцій. Виявилося, що це набагато швидше, ніж звичайна архітектура R-CNN.
3. Faster R-CNN – це вдосконалена версія Fast R-CNN, яка була навчена наскрізно за допомогою RPN (мережі регіональних пропозицій). RPN – це мережа, яка використовується для генерації ROI шляхом регресії опорних блоків. Таким чином, блоки прив’язки потім використовуються в задачі виявлення об’єктів.
4. Mask R-CNN додає гілку передбачення маски на Faster R-CNN, яка може одночасно виявляти об’єкти та передбачати їхні маски.
5. R-FCN замінює повністю з’єднані шари картами балів, чутливими до положення, для кращого виявлення об’єктів.
6. Cascade R-CNN вирішує проблему переобладнання під час навчання та невідповідності якості під час висновку шляхом навчання послідовності детекторів зі збільшенням порогових значень IoU.

Сімейна модель YOLO включає:

1. YOLO використовує менше опорних блоків (розділіть вхідне зображення на сітку S×S) для виконання регресії та класифікації. Це було створено за допомогою нейронних мереж даркнету.
2. YOLOv2 покращує продуктивність завдяки використанню більшої кількості прив’язок і нового методу регресії обмежувальної рамки.
3. YOLOv3 – це вдосконалена версія варіанту v2 із глибшою мережею детектора функцій і незначними змінами в представленні. YOLOv3 має відносно швидкий час висновку, займаючи приблизно 30 мс на висновок.
4. YOLOv4 (оновлення YOLOv3) працює, розбиваючи завдання виявлення об’єкта на дві частини, регресію для визначення позиціонування об’єкта за допомогою обмежувальних рамок і класифікацію для визначення класу об’єкта. YOLO V4 і його наступники технічно є продуктом іншої групи дослідників, ніж версії 1-3.
5. YOLOv5 – це вдосконалена версія YOLOv4 з технікою мозаїчного розширення для підвищення загальної продуктивності YOLOv4.

Сімейна модель CenterNet включає наступне:

1. SSD розміщує блоки прив’язки щільно над вхідним зображенням і використовує функції з різних згорткових шарів для регресії та класифікації блоків прив’язки.
2. DSSD представляє модуль деконволюції в SSD для поєднання низькорівневих і високорівневих функцій. У той час як R-SSD використовує операції об’єднання та деконволюції на різних функціональних рівнях для поєднання функцій низького та високого рівнів.
3. RON пропонує зворотний зв’язок і об’єктність перед ефективним вилученням багатомасштабних ознак.
4. RefineDet двічі уточнює розташування та розміри анкерних коробок, що успадковує переваги як одноетапного, так і двоетапного підходів.
5. CornerNet – ще один підхід на основі ключових точок, який безпосередньо виявляє об’єкт за допомогою пари кутів. Незважаючи на те, що CornerNet досягає високої продуктивності, у нього є ще багато можливостей для вдосконалення.
6. CenterNet досліджує візуальні шаблони всередині кожної обмежувальної рамки. Для виявлення об’єкта використовується трійка, а не пара ключових точок. CenterNet оцінює об’єкти як окремі точки, передбачаючи координати x і y центру об’єкта та його зону покриття (ширину та висоту). Це унікальна техніка, яка довела, що перевершує такі варіанти, як сімейство SSD і R-CNN.

Розпізнавання обличчя та людини. Більшість систем розпізнавання облич працює на основі виявлення об’єктів. Його можна використовувати для виявлення облич, класифікації емоцій або виразів і передачі отриманого поля в систему пошуку зображень, щоб ідентифікувати конкретну особу з групи. Розпізнавання обличчя є одним із найпопулярніших випадків використання виявлення об’єктів, мабуть, уже використовуєте його, коли розблоковуєте телефон обличчям. Виявлення особи також зазвичай використовується для підрахунку кількості людей у роздрібних магазинах або забезпечення показників соціального дистанціювання.

Інтелектуальна відеоаналітика. Виявлення об’єктів використовується в інтелектуальній відеоаналітиці (IVA) скрізь, де в торгових точках є камери відеоспостереження, щоб зрозуміти, як покупці взаємодіють з продуктами. Ці відеопотоки проходять через конвеєр анонімізації, щоб розмити обличчя людей і деідентифікувати людей. Деякі випадки використання IVA зберігають конфіденційність, дивлячись лише на взуття людей, розміщуючи камери нижче рівня колін і гарантуючи, що система фіксує присутність людини, не дивлячись безпосередньо на її ідентифікаційні ознаки. IVA часто використовується на заводах, в аеропортах і транспортних вузлах для відстеження довжини черг і доступу до зон обмеженого доступу.

Автономні транспортні засоби. Безпілотні автомобілі використовують функцію виявлення об’єктів, щоб помічати пішоходів, інші автомобілі та перешкоди на дорозі, щоб безпечно пересуватися. Автономні транспортні засоби, оснащені LIDAR, іноді використовують 3D-детектор об’єктів, який застосовує паралелепіпед навколо об’єктів.

Інтелектуальна відеохірургія. Хірургічне відео – це дуже шумні дані , які знімаються з ендоскопів під час найважливіших операцій. Виявлення об’єктів можна використовувати для виявлення важкопомітних об’єктів, таких як поліпи або ураження, які потребують негайної допомоги хірурга. Він також використовується для інформування персоналу лікарні про стан операції.

Перевірка дефектів. Компанії-виробники можуть використовувати функцію виявлення об’єктів для виявлення дефектів у виробничій лінії. Нейронні мережі можна навчити виявляти дрібні дефекти, від складок на тканині до вм’ятин або спалахів у литій пластмасі. На відміну від традиційних підходів до машинного навчання, виявлення об’єктів на основі глибокого навчання також може виявляти дефекти в дуже різноманітних об’єктах, таких як їжа.

Виявлення пішоходів. Це одне з найважливіших завдань комп’ютерного зору, яке застосовується в робототехніці, відеоспостереженні та автомобільній безпеці. Виявлення пішоходів відіграє ключову роль у дослідженні виявлення об’єктів, оскільки воно надає фундаментальну інформацію для семантичного розуміння відеозаписів. Проте, незважаючи на відносно високу продуктивність, ця технологія все ще стикається з проблемами, такими як різні стилі одягу або наявність оклюзійних аксесуарів, які знижують точність існуючих детекторів.

ШІ-навігація безпілотників. У наш час дрони оснащені неймовірними камерами та можуть використовувати моделі, розміщені в хмарі, щоб оцінити будь-який об’єкт, з яким вони стикаються. Наприклад, їх можна використовувати для перевірки важкодоступних місць у мостах на наявність тріщин та інших структурних пошкоджень або для перевірки ліній електропередач, замінюючи небезпечні звичайні вертолітні операції.

## 2.2 Алгоритми детекції людських облич на цифрових зображеннях

Алгоритм розпізнавання обличчя є основним компонентом будь-якої системи або програмного забезпечення для виявлення та розпізнавання обличчя. Фахівці поділяють ці алгоритми на два центральних підходи. Геометричний підхід зосереджується на відмінних рисах. Для отримання значень із зображення використовуються фотометричні статистичні методи. Потім ці значення порівнюються з шаблонами, щоб усунути розбіжності. Алгоритми також можна розділити на дві більш загальні категорії – моделі на основі функцій і цілісні моделі. Перший зосереджується на орієнтирах обличчя та аналізує їх просторові параметри та кореляцію з іншими ознаками, тоді як цілісні методи розглядають людське обличчя як єдине ціле.

Штучні нейронні мережі є найпопулярнішим і успішним методом розпізнавання зображень. Алгоритми розпізнавання обличчя засновані на математичних розрахунках, а нейронні мережі виконують велику кількість математичних операцій одночасно.

Алгоритми виконують три основні завдання:

* виявляють обличчя на зображенні, відео чи потоці в реальному часі;
* розрахувати математичну модель обличчя;
* порівнювати моделі з навчальними наборами або базами даних, щоб ідентифікувати або перевірити особу.

На сьогодні існує багато методів та алгоритмів для детекції людських облич, тому для проектування алгоритму детекції облич на основі виділення зарактерних точок необхідно провести їхній аналіз.

Згорткова нейронна мережа. Згорткова нейронна мережа (CNN) є одним із проривів у розвитку штучних нейронних мереж (ANN) та ШІ. Це один із найпопулярніших алгоритмів глибокого навчання, типу машинного навчання, у якому модель вчиться виконувати завдання класифікації безпосередньо на зображенні, відео, тексті чи звукі. Модель демонструє вражаючі результати в кількох сферах: комп’ютерний зір, обробка природної мови (NLP) і найбільший набір даних класифікації зображень (Image Net). CNN – це звичайна нейронна мережа з новими шарами – згортковим і пуловим. CNN може мати десятки й сотні цих шарів, і кожен із них вчиться виявляти різні особливості зображення.

Eigenfaces – це метод виявлення та розпізнавання облич, який визначає дисперсію облич у наборах даних зображень. Він використовує ці відхилення для кодування та декодування облич за допомогою машинного навчання. Набір власних облич – це набір «стандартизованих інгредієнтів облич», визначених статистичним аналізом великої кількості зображень облич. Рисам обличчя присвоюються математичні значення, оскільки цей метод використовує не цифрові зображення, а статистичні бази даних. Будь-яке людське обличчя є комбінацією цих значень з різними відсотками.

Fisherfaces – один із найпопулярніших алгоритмів розпізнавання облич; він вважається кращим за багато своїх альтернатив. Як вдосконалення алгоритму Eeigenfaces, його часто порівнюють з Eigenfaces і вважають більш успішним у розрізненні класів у процесі навчання. Ключовою перевагою цього алгоритму є його здатність інтерполювати та екстраполювати зміну освітлення та виразу обличчя. Є повідомлення про 93% точності алгоритму Fisherfaces у поєднанні з методом PCA на етапі попередньої обробки.

Методи ядра: PCA та SVM. Аналіз головних компонент (PCA) є універсальним статистичним методом з багатьма практичними застосуваннями. При використанні в процесі розпізнавання обличчя PCA має на меті зменшити розмір вихідних даних, зберігаючи при цьому найбільш відповідну інформацію. Він генерує набір зважених власних векторів, які, у свою чергу, створюють власні грані – великі набори різних зображень людських облич. Лінійна комбінація власних граней представляє кожне зображення в навчальному наборі. PCA використовується для отримання цих власних векторів із коваріаційної матриці набору навчальних зображень. Для кожного зображення розраховуються його основні компоненти (від 5 до 200). Інші компоненти кодують незначні відмінності між обличчями та шумом. Процес розпізнавання включає порівняння головного компонента невідомого зображення з компонентами всіх інших зображень.

Машина опорних векторів (SVM) – це алгоритм машинного навчання, який використовує принцип класифікації двох груп для розрізнення облич від «не-облич». Для кожної категорії модель SVM отримує позначений набір навчальних даних для класифікації нових тестових даних. Дослідники застосовують лінійні та нелінійні моделі навчання SVM для розпізнавання обличчя. Останні результати показують, що нелінійна навчальна машина має більший запас і кращі результати розпізнавання та класифікації.

Каскад Хаара – це метод виявлення об’єктів, який використовується для визначення місцезнаходження об’єктів на зображеннях. Алгоритм вивчає велику кількість позитивних і негативних зразків – перші містить об’єкт інтересу, а другий містить будь-що, крім того об’єкта, який шукається. Після навчання класифікатор може знаходити об'єкт інтересу на нових зображеннях. Метод використовувався для ідентифікації злочинців у поєднанні з алгоритмом локального бінарного шаблону для розпізнавання облич. Каскадний класифікатор Хаара використовує 200 (із 6000) ознак, що забезпечує 85-95% розпізнавання навіть із різними виразами.

Тривимірне розпізнавання. В основі технології 3D-розпізнавання облич лежить унікальна структура людського черепа. Будова черепа кожної людини унікальна і може бути описана кількома десятками параметрів. Цей метод розпізнавання обличчя заснований на порівнянні 3D-сканування обличчя з шаблонами бази даних. Він має суттєву перевагу – макіяж, волосся на обличчі, окуляри та подібні фактори не впливають на процес виявлення та розпізнавання. В останньому дослідженні використовувалася технологія відображення 3D-геометричної інформації на звичайній 2D-сітці. Це дозволяє поєднувати описовість 3D-даних із обчислювальною ефективністю 2D-даних і демонструє найвищу продуктивність, зареєстровану у FRGC v2 (Face Recognition Grand Challenge 3D facial database).

Аналіз текстури шкіри. Технологія розпізнавання шкіри має багато застосувань – алгоритми виявлення обличчя, фільтрація небажаних зображень, аналіз жестів рук тощо. Зазвичай вона використовує зображення з високою роздільною здатністю. Окремі випадки аналізу текстури шкіри використовують різні унікальні параметри, такі як родимки, колір шкіри, відтінки шкіри та багато інших. Останні дослідження, засновані на поєднанні особливостей текстури та кольору шкіри, показали цікаві результати. Дослідники використовували нейронну мережу для розробки та тестування системи розпізнавання шкіри. Нейронні мережі прямого зв’язку, використані в проекті, класифікували вхідні текстурні зображення як «шкіра» та «не-шкіра» та показали вражаючу продуктивність.

Теплові камери. Тепловізійна камера - це пристрій, який використовується для спостереження за розподілом температури досліджуваної поверхні. Розподіл температури відображається у вигляді кольорового зображення з різними кольорами, що відповідають температурам. Технологія вже має кілька практичних застосувань, які адаптуються до глобальних змін – сертифікати імунітету на основі смартфонів, дистанційне виявлення лихоманки та теплове розпізнавання обличчя. Моделі теплового розпізнавання обличчя засновані на унікальних температурних моделях людського обличчя. Постійні температурні «підписи» людини вимірюються за допомогою теплового інфрачервоного (ІЧ) уявного. Використання теплового методу в розпізнаванні обличчя має незаперечну перевагу – макіяж, волосся на обличчі, головні убори та окуляри не впливають на його точність. Крім того, він розрізняє братів-близнюків.

Адаптивна нейронечітка система перешкод (ANFIS) – це тип штучної нейронної мережі. Цей метод інтегрує принципи нейронних мереж із принципами нечіткої логіки та поєднує їхні переваги в одній структурі. ANFIS використовується для класифікації характеристик зображення, отриманих із наборів даних на етапі попередньої обробки. Науковці даних поєднують цей метод із різноманітними алгоритмами вилучення ознак. Таким чином, деякі дослідження повідомляють про неймовірну точність класифікації ANFIS у 97,1% після виділення ознак за допомогою 2D аналізу основних компонентів.

Локальні бінарні гістограми (LBPH). Цей метод використовує локальні двійкові шаблони (LBP), простий ефективний оператор текстури в комп’ютерному зорі, який позначає пікселі на зображенні, встановлюючи порогове значення сусідства кожного пікселя та розглядаючи результат як двійкове число. На етапі навчання алгоритм LBPH створює гістограми для кожного зображення, яке позначено та класифіковано. Кожна гістограма представляє кожне зображення з навчального набору. Таким чином, фактичний процес розпізнавання передбачає порівняння гістограм будь-яких двох зображень.

Facenet. Система розпізнавання облич FaceNet, розроблена дослідниками Google у 2015 році, заснована на тестових наборах даних розпізнавання облич. Доступні попередньо навчені моделі та різні сторонні реалізації з відкритим кодом роблять цю систему досить поширеною. FaceNet показує відмінні результати в дослідницьких опитуваннях, тестуванні продуктивності та точності порівняно з іншими алгоритмами, розробленими раніше. FaceNet точно витягує вбудовані обличчя, високоякісні функції, які використовуються для навчання систем ідентифікації обличчя на наступних етапах.

NEC. Рішення, розроблене японською технологічною компанією NEC, дозволяє високоточно ідентифікувати людей, розпізнаючи вікові зміни. Рішення використовує Adaptive Region Mixed Matching, модель, яка зосереджується на дуже подібних сегментах для відображення. Технологія NEC розділяє вхідні та зареєстровані зображення на невеликі сегменти та зосереджується лише на сегментах із більшою схожістю. Це дозволяє системі показувати більш високу точність ідентифікації, навіть якщо обличчя в масці або окулярах. Як базовий алгоритм рішення NEC використовує узагальнене векторне квантування з навчанням (GLVQ).

Алгоритм Megvii. Китайська технологічна компанія Megvii стала відомою в усьому світі після того, як представила платформу додатків для розпізнавання облич. Це багатоцільове програмне забезпечення. Алгоритм Megvii заснований на технології виявлення графів і пошуку нечітких зображень. Технічне рішення використовує власну структуру глибокого навчання компанії MegEngine, побудовану на великих даних. Технологія компанії успішно виконує вилучення інформації про обличчя та включає кілька ключових функцій: виявлення та відстеження обличчя та людини, розпізнавання та кластеризацію облич, визначення ключових точок, оцінку атрибутів обличчя та систему пошуку облич.

Кожна технологія розпізнавання обличчя має свої ефективні функції. Однак останні дослідження довели, що найкращі результати досягаються за допомогою комбінацій різних алгоритмів і методів. Ці комбінації спрямовані на вирішення багатьох рутинних проблем процесу розпізнавання обличчя – відмінностей у виразах обличчя, позуванні, умовах освітлення, шумах зображення тощо. Останні експерименти поєднують алгоритм LBP із передовими методами обробки зображень: двосторонній фільтр, вирівнювання гістограми, налаштування контрастності, і змішування зображень. Така техніка демонструє вражаючі покращення коду LBP і виглядає дуже перспективною для подальших досліджень.

Існує безліч алгоритмів і методів розпізнавання облич. Хоча всі вони мають одну переважну мету, вони можуть бути пов’язані з завданням і проблемою. Залежно від цілей використання та умов реалізації, вони варіюються від нейронних мереж і математичних моделей до технічних рішень приватних компаній.

Було розглянуто найпоширеніші алгоритми та методи. Проте все більше досліджень і наукових експериментів показують незаперечні переваги поєднання різних алгоритмів для отримання кращих результатів у процесі розпізнавання обличчя. Це призводить до появи нових технік і методів використання.

На рисунку 2.4 наведено класифікацію технологій виділення людських облич.

Методи детекції людських облич

Згорткова нейронна мережа

Метод Eigenfaces

Аналіз текстури шкіри

Локальні бінарні гістограми

Каскад Хаара

Тривимірне розпізнавання

Метод Fisherfaces

Методи ядра: PCA та SVM

Рисунок 2.4 – Класифікація методів детекції людських облич

## 2.3 Алгоритми детекції облич на основі виділення ключових точок

При розробці алгоритму детекції людських облич за основу було обрано припущення, що всі людські обличчя є подібними, мають подібну форму та розташування структурних елементів (рисунок 2.6). Звісно, при отриманні ушкоджень обличчя може отримати деякі фізичні зміни, а сама модель не зможе зі 100% ймовірністю описувати конкретний випадок. Проте якщо розглядати звичайну ситуацію, то можна зробити наступні висновки. Скруктурні елементи обиччя (очі, ніс, рот) мають чітку послідовність в розташуванні на людському обличчі. Форма голови людини є овал з параметром колоподібності від 0,7. Тому для пошуку та детекції людських облич необхідно здійснювати на зображенні пошук овальних областей після чого за допомогою додаткового аналізу превіряти наявність сруктурних елементі обличчя.



Рисунок 2.5 – Модель розташування складових елементів людського обличчя

На основі даної моделі розроблено алгоритм детекції облич на основі виділення ключових точок. Блок-схема алгоритму наведена на рисунку 2.6.

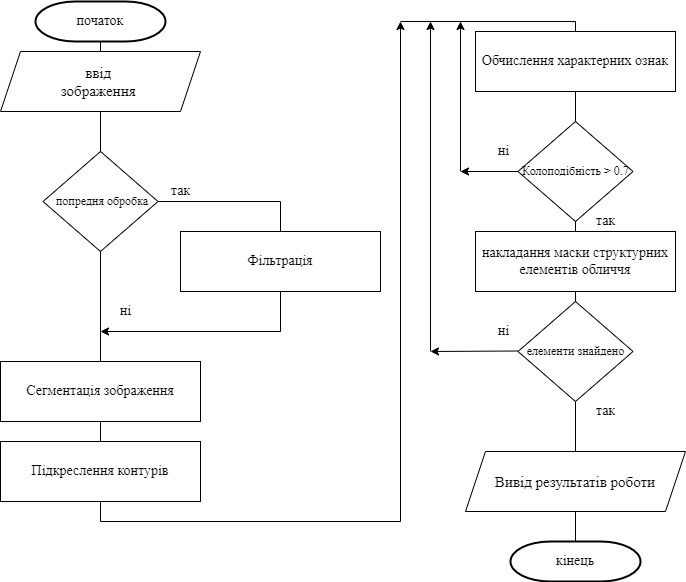


Рисунок 2.6 – Блок-схема алгоритму детекції людських облич на основі виділення ключових точок

Розроблений алгоритм детекції людських облич на основі виділення ключових точок базується на розробленій моделі людського обличчя та набору характерних точок містить наступну послідовність кроків:

1. Встановлення параметрів роботи алгоритму
2. Завантаження зображення.
3. Попередня обробка зображення на основі статистичної фільтрації.
4. Сегментація зображення за допомогою алгоритму водоподілу.
5. Підкреслення границь однорідних областей.
6. Визначення характеристичних ознак кожної однорідної області (колоподібність, параметри описаного прямокутника).
7. Аналіз характерних ознак одноріжних областей. Якщо всі області проаналізовані, то переходимо на крок .
8. Якщо значення параметра колоподібності менше ніж 0,7 то дана область відкидається з подальшого аналізу та здійснюється перехід на крок 7.
9. За допомогою накладання масок проводиться пошук структурних елементів людського обличчя: очі, ніс, рот.
10. Якщо структруних елементів за допомогою маски не було знайдено то дана область відкидається з подальшого аналізу та здійснюється перехід на крок 7.
11. На основі виділених структурних елементів проводимо опис отриманого людського обличчя та додаєм даний опис у відповідний масив.
12. Виводимо результати роботи алгоритму у зручному для користувача форматі.

Пошук людських облич за допомогою аналізу однорідних областей з подальшим аналізом області інтересу дозволяє зменшити часові та ресурсні затрати на опрацювання вхідного зображення, оскільки однорідні області відсікають на кожному з етапів аналізу чим зменшують кількість необхідних операцій на наступних кроках. Окрім того запропонована модель та підібрані шаблони розташування структурних елементів людського обличчя значно спрощують процес детекції.

До основних переваг розробленого алгоритму відносяться:

* висока швидкодія за рахунок видалення малоінформативних областей після кожного етапу аналізу;
* можливість роботи на комп’ютерах з низькими апаратними характеристиками;
* можливість реалізації на лбій мові програмування;
* корекція роботи алгоритму може здійсьнюватись в процесі його роботи.

Недоліки:

* кіцевий результат роботи алгоритму залежить від точності результатів роботи алгоритмів попередньої обробки (фільтрації, сегментації);
* результати роботи залежать від якості вхідного зображення та відсотка відкритості обличчя.

В результаті моделювання роботи запропонованого алгоритму він показав хороші результати по швидкості роботи та ефективність під час детекції людських облич на цифровому зображенні.

## 2.4 Висновки до розділу

Проведно аналітичний огляд алгоритмів виділення об’єктів на цифрових зображеннях на основі статистичних, математичних операцій та використанні механізму штучного інтелекту, що дало можливість визначити переваги та недоліки сучасних технологій виділення людських облич.

Розроблено алгоритм детекції людських облич на основі виділення ключових точок, що дало можливість розробити структуру та реалізувати програмних додаток детекції людських облич в системі аналізу цифрових зображень.

# 3 Програмний додаток детекції людських облич на основі виділення ключових точок

## 3.1 Структура програмного додатку детекції людських облич

Виявлення та розпізнавання об’єктів є двома важливими завданнями комп’ютерного зору. Виявлення об’єкта визначає присутність об’єкта та/або його обсяг і розташування на зображенні. Розпізнавання об’єктів визначає клас об’єктів у навчальній базі даних, до якої належить об’єкт. Виявлення об'єкта зазвичай передує розпізнаванню об'єкта. Його можна розглядати як двокласове розпізнавання об’єктів, де один клас представляє клас об’єкта, а інший клас – необ’єктний клас. Виявлення об’єктів можна далі розділити на м’яке виявлення, яке виявляє лише присутність об’єкта, та жорстке виявлення, яке виявляє як присутність, так і розташування об’єкта. Поле виявлення об’єктів зазвичай виконується шляхом пошуку кожної частини зображення, щоб локалізувати частини, фотометричний або геометричний параметр яких 2 властивості відповідають властивостям цільового об’єкта в навчальній базі даних. Це можна зробити шляхом сканування шаблону об’єкта на зображенні в різних місцях, масштабах і поворотах, і виявлення оголошується, якщо подібність між шаблоном і зображенням є достатньо високою. Подібність між шаблоном і областю зображення можна виміряти їхньою кореляцією (SSD). За останні кілька років було показано, що детектори об’єктів на основі зображень чутливі до навчальних даних. Алгоритм Віоли Джонс є базовим для реалізації в програмному додатку. Обраний алгоритм є ефективним методом для виявлення обличчя. Загалом, цей алгоритм не лише обмежений виявленням обличчя, але також може використовуватися для багатьох завдань виявлення жорстко структурованих об’єктів. Алгоритм складається з трьох основних концепцій, які дозволяють розробити детектор обличчя в режимі реального часу: функції, подібні до Хаара, інтеграл зображення, навчання та каскадний класифікатор. Застосовуючи ці ознаки, система може визначити наявність чи відсутність обличчя людини. Характеристики, подібні до Хаара. Характеристики, подібні до Хаара, використовуються каскадним класифікатором Хаара для виявлення обличчя людини. Існує три утворення хаар-подібних: перший формат – це грань, другий тип – лінія, а останній тип – чотири прямокутники. Використовуючи цілісне зображення, принцип Хаара забезпечить швидке обчислення. Це називається Хаар-подібні особливості. Алгоритм шукає специфічну рису обличчя. Це виявлення бере зображення та перетворює його у вікно 24x24 і розмазує кожну функцію Хаара на цьому вікні піксель за пікселем. Спочатку алгоритм вимагає багато позитивних зображень (зображення облич) і негативних зображень (зображення без облич) для навчання класифікатора.

Функції прямокутника інтегрального зображення можна швидко визначити за допомогою проміжного представлення зображення, яке називається інтегральним зображенням. Цілісне зображення складається з невеликих одиниць зображення даного зображення.

Після того, як обличчя виявлено, наступним кроком є виділення ознак, що виконується за допомогою алгоритму лінійного двійкового шаблону. Початковий крок цього алгоритму полягає в перетворенні тестового зображення в градації сірого. Це зображення розміром *L*x*M* у пікселях буде розділено на регіони. Для регіонів використовується однаковий розмір пікселя, утворюючи *n*x*n* регіонів. Кожна область проходитиме через оператор лінійного двійкового шаблону. Виконання цього процесу призведе до 8 двійкових значень. Пов’язуючи двійкові значення, це призводить до двійкового числа. Значення LBP отримується шляхом перекладу 8двійкового числа в десяткове число, воно буде в діапазоні 0-255.

На основі виділених етапів роботи алгоритму детектування людських облич на цифровому зображенні була розроблена структура програмного додатку для аналізу та обробки цифрових зображень з елементами детекції облич. Спрощену структуру програмного додатку наведено на рисунку 3.1.

Центральний модуль програмного додатку

Модуль імпорту вхідних зображень

Модуль попередньої обробки

Модуль виділення однорідних областей

Модуль обчислення кількісних параметрів областей

Модуль оцінки результатів роботи

Модуль встановлення атрибутів

Модуль виводу даних

Рисунок 3.1 – Узагальнена структура програмного додатку для детекції людських облич

Запропонована структура дозволяє розбити процес реалізації програмного додатку на окремі етапи та проводити розробку окремих модулів незалежно один від одного. Запропоноване архітектурне рішення дозволить ефективно використовувати наявні апаратні ресурси та виконувати поставлені задачі з необхідною точністю та гранично допустимий час. Окрім того використання програмних блоків дозволяє при розробці розпаралелити розробку окремих блоків, що значно спрощує та прискорює процес створення програмного забезпечення. Серед головних структурних частин слід відмітити такі:

Центральний модуль програмного додатку – в загальній архітектурі було виділено ряд методів та структур для організації елемента керування та взаємодії між окремими структурними підсистемами. Зокрема даний модуль забезпечує можливість перекодування структур даних при пердачі між різними частинами програмної розробки. Додатково даний модуль може підключати або відключати окремі програмні одиниці в структурі, що дозволить проводити більш гнучно процедуру тестування та налаштування програмної системи в реальних умовах. Додатково в даний модуль було віднесені функції перевірки цілісності та робочого стану окремих блоків програми з метою організації системи повідомлень користувача при винекненні критичних або помилкових ситуацій. Вхідними даними даного модуля є вектор параметрів роботи програми. Вихідними – активні параметри роботи активного модуля, а також запити до користувача у випадку винекнення помилкової ситуації або відсутності/ не справності окремо взятого модуля або зовнішнього анаратного засобу.

Модуль виділення однорідних областей – даний модуль ж головним в структурі програмної системи і виконує більше 60% задач які ставляться перед системою в цілому. По-перше, в ньому реалізовані алгоритми сегментації. В даній реалізації було реалізовано алгоритми порогової сегментації з одним або декількома порогами та алгоритм водоподілу. Проте при необхідності даний перелік алгоритмів сегментації може бути розширений за рахунок підключення сторонніх цифрових бібліотек або реалізації додаткових алгоритмів. Проте як показали тести реалізованих алгоритмів цілком достатньо для вирішення поставлених завдань. Інша група алгоритмів – це алгоритми проходження контуром. В дану групу ввійшли алгоритми жука та послідовного проходження з можливістю зворотнього ходу. Дані алгоритми забезпечують можливість отримання опису контуру однорідної області з метою подальшої аналізу та отримання характеристичних ознак. Дані алгоритми дозволяють отримати як чотирьох так і восьмизв’язний контур. Вибір типу опису виконується користувачем в залежності від заданих обмежень та поставлених завдань. Окрім того в даному модулі реалізовано механізми перевірки достовірностіотриманих даних шляхом аналізу отриманих контурних ліній. Серед умов які перевіряються має бути те, що контур обовязково повинен бути замкнутий, одноріжна област не повинна містити велику кількість вкраплень інших об’єктів, не більше 4-5. Ця кількість відповідає кількості структурних елементів на обличчі: двоє очей, ніс та рот. І ще одна одноріжна область допускаєтсья на випадок якщо будут присутні природні утворення на шкірі, наприклад пігментне п’ятно. Якщо кількість внутрішніх областей більша за заданий поріг то така область викидається з подальшого аналізу.

Модуль виділення однорідних областей

Функції сегментації

Функції контурного аналізу

Функції встановлення та корекції параметрів роботи

Функції користувацького інтерфейсу

Функції контролю отриманих результатів

Рисунок 3.2 – Узагальнена структура модуля виділення одноріжних областей

Модуль попередьої обробки. Вхідні зображення можуть бути отримані як з бази дани і вже попереддньо пройти додаткову цифрову обробку з метою усунення дефектів або підкреслення важливих елементів на цифровому зображенні. Проте, зображення можна отримувати і з відеореєструючої апаратури, що в свою чергу може призвести появу додаткових шумів та інших артефактів які можуть призвести до зниження ефективності роботи програми та збільшення кількості помилкових виділень. Тому в структуру програмного додатку було додано набір функцій для покращення якості цифрового зображення. Зокрема проводиться процедура видалення малоінформативних областей за допомогою механізмів фільтрації на основі статистичних фільтрів. Додатково після отримання опису контурної функції можна проіести фільтрації на основі морфологічних операці нарощування та стоншення. На вхід модуля попредньої обробки подається цифрове зображення у вигляді масиву точок які відповідають елементам дискретизованого вхідного зображення. В результаті робти даного модуля буде отримано зображення з покращеними характеристиками.

Модуль встановлення атрибутів – допоміжний модуль який реалізує механізми встановлення та корекцію параметрів робти програмної системи. Даний модуль буде корисним для досвідчених користувачів, оскільки дозволяє більш точно налаштувати роботу програмної системи в залежності від поставлених завдань. Проте окрім системних параметрів за допомогою даного модуля можна встановити параметри графічного інтерфейсу, що дозволить зробити програму більш зручною у роботі та дозволить індивідоально налаштувати більшусть параметрів. Серед значень які можна налаштувати є параметри відображення розміру та типу шрифтів, вибір алгоритмів сегментації та контурного аналізу, параметри статистичних фільтрів, необхідність/ відсутність додаткових підказок, формат та опції звіту тощо.

Модуль отримання характеристичних ознак – Методи даного модуля проводять додатковий аналіз однорідних областей з метою отримання кількістних та якісних параметрів, що описують дану область. Серед кількісних парметрів слід відмітити:

* довжина центральної лінії (найбільша відстань між двома точками які належать контуру області);
* колоподібність (відношення центральної лінії до відрізка перпендикулярно проведеного через середину центральної лінії, а кінці якого знаходяться на контурі області);
* площі (сумі всіх точок які належать до одноріжної області);
* периметру (довжині контурної лінії);
* однорідність (кількість одноріжних облстей які розміщені в середині досліджуваної однорідної області);
* зовнішній прямокутник (координати вершин прямокутника який описує дану однорідну область);
* колір (усереднене значення кольорів пікселів які входля в одноріжну одласть).

Окрім вище перелічених параметрів в програмі реалізована можливість їх поєднання для отримання авих характеристичних ознак, наприклад поєднання площі та кольору, дозволить вирахувати який відсоток обличчя займають структурні елементи (очі, ніс, рот) тощо.

Модуль оцінки результатів роботи – після провдення усіх операцій над зображенням програма формує звіт про виконану роботу, а саме візуалізує виділені обличчя, та повертає масив з обчисленими кількісними зарактеристиками. Дані результати роботи програми можна перевірити візуально або активувати систему автоматизованої перевірки результатів. В системі закладено механізми оцінки кількісних характеристик і можливість їх поєднання. Наприклад якщо відсоток які займають структурні елементи на обличчі більші за 20%, то отрмані дані є малоймовірними, а при їх обчисленні скоріш за все сталась помилка або людина має нестандартну структуру обличчя. Іншим варіантом помилкового обчислення параметрів обличчя є наявність менше чим 4 внутрішніх областей на однорідній області. Оскільки малоймовірним є той факт, що в людини відсутні якісь структурін елементи. Звісно можливий варіант, що колір може бути близький до кольору шкіри обличчя, проте за природніх умов та без використання додаткових штучних засобів (наприклад косметики чи гриму) малоймовірно. Тому цей модуль дозволяє провести автоматичну оцінку та відкинути наперед хибні виділення з подальшої обробки. Якщо методи однозначно не можуть прийняти рішення по якійсь одноріжній області, то в такому випадку система надає можливість користувачеві самостійно оцінити виділений фрагмент для остаточного прийняття рішення.

При проектуванні та подільшому моделюванні архітектури програмної системи був використаний модульний підхід. Перевагами даного способу організації програмної системи є можливість паралельної роботи над різними модулями програми та механізм встановлення тестових вхідних даних. При використанні тестових даних вони замінюють результат роботи модуля на наперед відомий набір даних, що в свою чергу дозволяє оцінити роботу системи без підключення окремих модулів які перебувають у процесі відлагодження або розробки. Окрім того при такіій побудові систему просто модифікувати та додавати/вилучати нові модулі.

Наступним етапом створення програмного додатку є процес реалізації запропонованих алгоритмів, проте перед цим доцільно провести процес моделювання можливості коректної взаємодії між різними модулями програмного додатку та функціональними модивостями запропонованих алгоритмів виконувати поставлені перед системою завдання. Моделюваня проводилось за допомогою механізмів універсальної мови моделювання (UML), а семе ввикористовувалисьь діаграми прециденів та діаграма послідовності Серед недоліків застосування універсальної мови моделювання слід відмітити:

* додатковий час на розробку. Оскільки процес моделювання вимагає розробку моделей Для проведення грунтовного моделювання необхідно затратити час для того, щоб діаграма була синхронізована з фактичним кодом та розумною. Діаграми UML не запускаються, але вимагають багато часу. Тому вони корисні тільки в тому випадку, якщо розробник може ними керувати.
* можливі втрати інформації. Оскільки моделювання це процес яки вимагає створення та аналізу моделей, то це не можливо зробити за короткий час. Тому для проведення якісного моделювання та оцінки отриманих результатів необхідно потратити час. Проте проблеми та недоліки які будуть виявлені перед процесом написання програмного коду зможуть бути виправлені, що значно зменшити часові та фінансові затрати.
* затрати фінансових ресурсів системи. Для проведення якісного модулювання необхідне відповідне програмне забезпечення, а отже потрібні кошти на його закупівлю та підтримку у робочому стані.

Моделювання на основі діаграми предецентів дозволить оцінити рівень взаємодії окремих функції та вирахувати відсотки їх важливості в загальній структурі проекту. Додатково розглядається функціональні можливості які отримає користувач при роботі з системою. Приклад діаграми прецеденів наведено на рисунку 3.3.

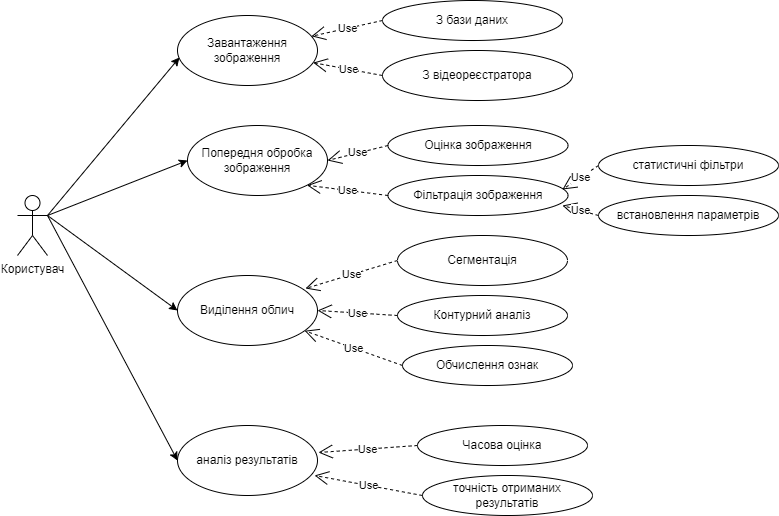


Рисунок 3.3 – Діаграма прецедентів програмної системи

Як показують результати моделювання, що наведені на діаграмі прецедентів запропонована архітектура програмного додатку дозволяє виконувати усі закладені на етапі функції. Аналіз отриманої модулі показав, що користувачі отримують повний доступ до функцій системи для можливості виконання поставлених завдань. Проте не для використання частини з них необхідно пройти певний шлях. Це зроблено з метою запобігання введення некоректної інформації або активації функцій в непраивльній послідовності. В системі передбачено механізи додаткового контролю отриманих даних, проте все ж таки деякі функції були приховані від користувачів з метою підвищення стійкості та роботоздатності програмної системи. Дане моделювання показало правильність обраної структури та набору реалізованих функцій. Процес опанування програмою є достатньо короткий та не ввимагає від нових кристувачів грибоких попередніх знань комп’ютерної техніки.

Наступний етап моделювання – це аналіз запропонованих кроків та їх коректність які необхідно зробити користувачеві для досягнення результатів. Аналізувалась взаємодія процесів та об’єктів, що будуть відбуватись під час роботи програми. Результат моделювання наведено на рисуноку 3.4.

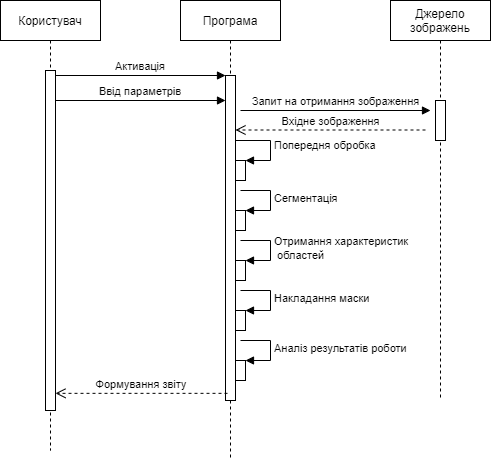


Рисунок 3.4 – Діаграма послідовності програмної системи

Навдена діаграма у повній мірі дозволяє проілюстурвати можливості запропонованої архітектури програмного додатку. При роботі з програмою було реалізовано лінійних процес обробки даних та виконання поставлених задач. З однієї сторони розпаралелення внутрішніх процесів могло б прискорити роботу програмного додату, проте вхідна інформація подається у вигляді окремих зображень, та не вимагає мттєвого результату, тому було прийнято рішення саме по такому принципу організації роботи. Даний підхід достатньо не вимогливий до апаратних засобів та забезпечує достатню точніс при аналізі вхідних цифрових зображень.

Ще одним важливим еементом дослідження внутрішньої структури програмного додатку є аналізу взаємодії між окремими класами в середині самої програми. Для проведення оцінки взаємодії окремих програмних класів та повноти їх снутрішньої рахітектури використовується діаграма класів яка наведена на рисунку 3.5

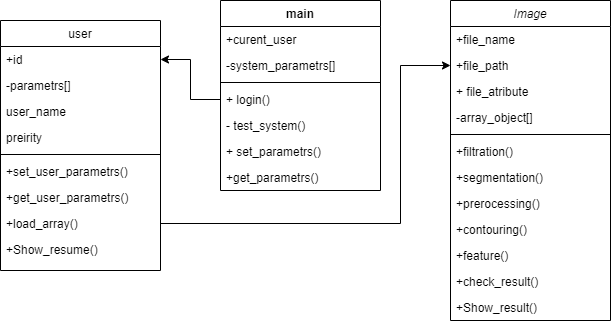


Рисунок 3.5 – Діаграма класів програмної системи

Аналіз діаграми класів показав, що при розробці дерева класів жодний з них не містить повторюваної інформації та має передбачені механізми взаємодії між окремими його елементами. Створення класів для опису центрального блоку, користувача та самого цифрового зображення дозволив зосередити основні даніта функції які їх оброблятимуть в одному місці. Це дало можливість створювати велику кількіст об’єктів класу, при цьому зміни які вносяться для якого одгого з ним будуть доступні і іншим.

Етап моделювання дозволив оцінити основні фактиро функціонування програмного забезпечення, а також внести корективи в місця де могли виникати потенційні помилкові ситуації. Даний етап моделювання успішно завершився. В результаті було доведено правильність спроектованої структури програмного засобу, повнота та цілісність внутрішніх класів та коректність взаємодії окремих модулів при обробці та аналізу інформації.

Розроблений програмний додаток має допоміжну функцію при роботі з середовищами розробки програмних засобів, а тому при проектуванні графічного інтерфейсу було принято рішення використати мінімальстичний стиль. Оскільки користувач всеодно не буде безпосередньо постійно взаємодіяти з програмою, тому на головному вікні передбачено використання невеликої кількості перемикачів які забезпечать неохідний мініму для коректної роботи програми. Гафічний інтерфейс користувача розробленої програмної системи наведено на рисунку 3.6.

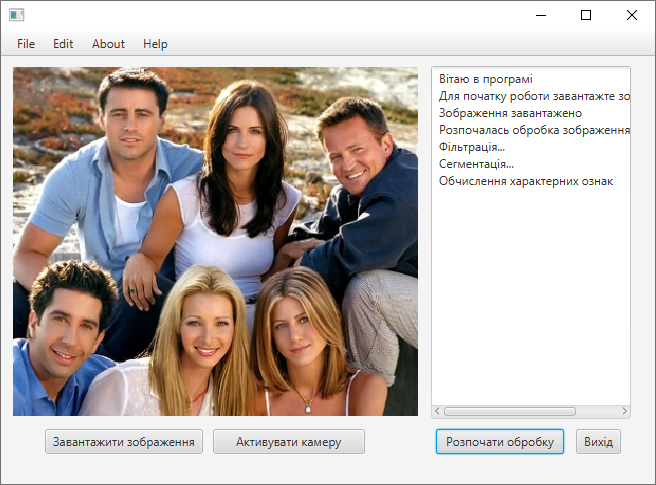


Рисунок 3.6 – Графічний інтерфейс користувача

Графічний інтерфейс користувач було розроблено з врахуванням сучасних вимог до програмних інтерфейсів, а саме використано принцип мінімальзму. Принцип мінімальзму полягає в тому, щоб зменшити кількість активних елементів на основних вікнах програмних додатків. Дане дизайнерське рішення дозволяє в подальшому мінімізувати кількість операцій для отримання результатів роботи програмного додатку, що в свою чергу зменшує кількість часу необхідного для роботи з системою. В випадку потреби отримати результат за короткий час дана превага є відчутною. Окрім того, зменшення кількості функціональних елементів дозволяє зменшити час на опанування даним програмним додатком користувачам з малим досвідом використання комп’ютерної техніки

Для роботи з програмою користувачеві необхідно виконати просту послідовність кроків. Для даної послідовності кроків в системі впроваджено елементи допомоги у вигляді підказок які можуть з’являтись під час запуску програмного додатку, а також може бути проілюстрований в процесі роботи програми у вигляді додаткових підказок. Послідовність необхідних дій відображено на рисунку 3.7:

Активація системи

Встановлення параметрів роботи

Завантаження зображення

Попередня обробка

Детекція облич

Оцінка отриманих результатів

Рисунок 3.7 – Послідовність дій користувача при роботі з інтелектуального вводу тексту програмного коду

Як показано зі схеми дій користувача, при роботі користувач повинен виконати три основі дії з програмною системою:

* запуск та корекція параматрів роботи;
* завантаження цифрового зображення;
* оцінка отриманого масиву областей з людськими обличчями та їх опис.

Окрім того користувач може виконати ще дві додаткові дії, які необхідні у більшій мірі для підвищення рівня коректності роботи програми.

## 3.2 Програмні модулі додатку детекції облич

Сучасний світ переповнений даними, і зображення становлять значну частину цих даних. Однак перш ніж їх можна буде використовувати, ці цифрові зображення мають бути оброблені – проаналізовані та оброблені, щоб покращити їхню якість або витягти деяку інформацію, яку можна використати.

Загальні завдання обробки зображень включають дисплеї; базові маніпуляції, такі як кадрування, перевертання, обертання тощо; сегментація зображення, класифікація та виділення ознак; відновлення зображення; і розпізнавання зображень. Python є чудовим вибором для таких типів завдань обробки зображень через його зростаючу популярність як наукової мови програмування та безкоштовну доступність багатьох найсучасніших інструментів обробки зображень у його екосистемі.

Бінаризація зображення перетворює зображення у градаціях сірого на чорно-біле зображення (тобто 0 і 255 пікселів відповідно). Цього можна досягти за допомогою процесу, який називається пороговим визначенням. Отже, за допомогою порогового визначення ми можемо отримати бінарні зображення. Тепер порогове значення можна визначити як процес, у якому кожен піксель перетворюється на 0 або 255 залежно від того, більше чи менше його значення порогового значення. Якщо значення пікселя перевищує порогове значення, воно перетворюється на 255, інакше воно перетворюється на 1. Ось як працює порогове значення. Зараз існує 3 різні типи порогових методів:

Просте порогове визначення. У простому методі порогового значення ми повинні передати параметр порогового значення функції. Таким чином, порогове значення вручну визначає програміст після багатьох звернень і спроб, чи підходить конкретне значення для обробки зображення. Це порогове значення залишається незмінним для кожного пікселя та перетворюється на 0, якщо його значення менше за порогове значення, і на 255 в іншому випадку. Це можна реалізувати за допомогою бібліотеки cv2 наступним чином:

import cv2 as cv

img = cv.imread('sample.png',)

ret.thresh1 = cv.threshold(img,127, 255, cv.THRESH\_BINARY)

Наведений вище код реалізує функцію cv. Threshold(). Він приймає перший аргумент як зображення, яке має бути сірим. Другим аргументом має бути порогове значення, яке використовується для перетворення пікселя. Третє значення є максимальним значенням 255, яке використовується у наведеному вище прикладі. Четвертий параметр – тип використовуваного порогового значення.

Адаптивне порогове визначення. Цей метод іноді більш корисний для бінаризації, коли зображення має різні умови освітлення. У простому пороговому значенні ми використовували глобальне граничне значення для всіх пікселів. У разі адаптивного порогового визначення алгоритм ділить зображення на менші області та автоматично визначає порогове значення для кожної меншої області. Це дає набагато кращі результати, ніж глобальне порогове значення. Його можна реалізувати за допомогою бібліотеки cv2 наступним чином:

import cv2 as cv

img = cv.imread('sample.png',)

th2 =cv.adaptiveThreshold(img, 255, cv.ADAPTIVE\_THRESH\_MEAN\_C, cv.THRESH\_BINARY, 11, 2)

У методі адаптивного порогового визначення першим параметром є зображення. Другий параметр – максимальне значення пікселя. У третьому аргументі може бути 2 параметри, які можна передати, тобто cv.ADAPTIVE\_THRESH\_MEAN\_C і cv.ADAPTIVE\_THRESH\_GAUSSIAN\_C. П'ятий аргумент – це розмір блоку зображення, тобто розмір меншої області зображення. Останнім аргументом є константа C.

Пороги Оцу. Ця техніка обрбки Оцу схожа на техніку простого оббивання порога. Єдина головна відмінність полягає в тому, що тут нам не потрібно вибирати глобальне значення порогу, на відміну від простого порогового значення, яке явно вимагає значення порогу. Цей алгоритм автоматично визначає порогове значення та застосовує його для отримання найкращого результату. Це також реалізовано аналогічно, просто cv.THRESH\_OTSU передається як додатковий параметр.

ret2,th2 = cv.threshold(img,0,255,cv.THRESH\_BINARY+ cv.THRESH\_OTSU)

Окрім того в проекту було використаро ряд зовнішніх цифрових бібліотек які дозволяють реалізувати окремі допоміжні алгоритми та мемханіки. Це дозволить в значній мірі скоротити час на реалізації проекутв цілому, зосередившись в основному на головних елементах програми. Одним з таких додаткових модулів є Scikit–image. Scikit–image – це пакет Python з відкритим кодом, який працює зNumPy масивамиВін реалізує алгоритми та утиліти для використання в дослідженнях, освіті та галузевих програмах. Це досить проста і зрозуміла бібліотека навіть для тих, хто вперше знайомиться з екосистемою Python. Код високої якості, рецензований і написаний активною спільнотою волонтерів.

import matplotlib.pyplot as plt

%matplotlib inline

from skimage import data,filters

image = data.coins()

edges = filters.sobel(image)

plt.imshow(edges, cmap='gray')

NumPy є однією з основних бібліотек у програмуванні Python і забезпечує підтримку масивів. Зображення – це, по суті, стандартний масив NumPy, що містить пікселі точок даних. Таким чином, використовуючи базові операції NumPy, такі як нарізка, маскування та модне індексування, ви можете змінювати значення пікселів зображення. Зображення можна завантажити за допомогою skimage та відобразити за допомогою Matplotlib. Дана бібліотека дозволяє реалізовувати такі алгоритми як перетворення з одного колірного простору в інший, отримувати зображення з камер, обчислювати деякі кількісні ознаки. Приклад коду переконвертуваня зображення з кольорового в монохромне наведено нижче.

import numpy as np

from skimage import data

import matplotlib.pyplot as plt

image = data.camera()

type(image)

numpy.ndarray #Image is a NumPy array:

mask = image < 87

image[mask]=255

plt.imshow(image, cmap='gray')

На сьогоднішній день мова python є багатофункціональним засобом для реалізації програмних систем різної складності з великою кількістю цифрових бібліотек. Вона дозволяє спростити написання складних програмних систем за рахунок підключення уже готових рішень, що в результаті дозволяє зосередити основну увагу на вирішенні та кодуванні основних алгоритмів та структур даних.

## 3.3 Тестування та аналіз реалізованого програмного додатку

Для проведеня тестових випробувань запропонованого програмного додатку було обрано апаратне забезпечення виходячи з наступних особливостей проведення тестування:

* програма повинна виконувати обчислення на робочих станціях з середніми технічними параметрами;
* детекція може проводитись як з статичних зображень так і з кадрів відеофайлів отриманих з камер спостереження;
* програма повинна повинна повертати результати опрацювання зображення за заданий проміжок часу, можливий варіант роботи в режимі реального часу.

В результаті для проведення тестування було обрано робочу станцію з «офісними» параметрами, що є достатніми для повсякденного використання та дозволяє проектувати на відлагоджувати програмні системи для різних сфер використання. Може використовуватися як робоче місце дизайтера, секретаря, науковця тощо. Робоча станція має такі технічні характеристики:

* Процесор: Intel Core i3-7100 1 шт.
* Чіпсет: Intel® B250.
* Оперативна пам’ять: 8GB (2x4GB) DIMM DDR4 (частота шини 2400MHz) Crucial.
* Носій інформації: 500GB (1x500GB) SATA 6Gb/s 3.5" Seagate.
* Відеокарта: ASUS, серія: GeForce GT 730, об'єм відео-пам'яті 2GB, GT730, встановлено 1 шт.
* Мережа: на мат. платі: 1Gb/s 1шт.
* Блок живлення: 1шт. 450W.
* Фірма виробник: Gigabyte.

Для зручності робти з програмною системою та для об’єктивної оцінки отримуваних результатів було обрано монітор фірми dell. Дані монітори відповідають вимогам технічного завдання та можуть використовуватись в проекті. Технічні характеристики пристрою відображення зображень (DELL P2018H 19.45″ ) можна переглянути на рисуноку 3.8:

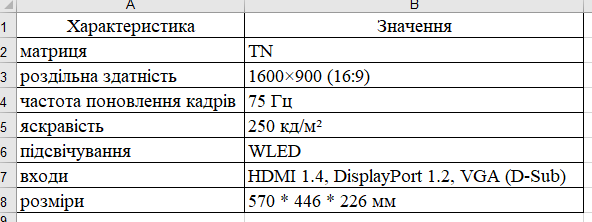


Рисунок 3.8 – Технічні характеристики пристрою відображення зображень

Обрані технічні параметри робочої станції для тестування в повній мірі відповідають обмеженням технічного завдання та вимогам до програмної системи. Тому, дану робчу станції можна використовувати для обробки тестових вибірок та отримання достовірних результатів обробки вхідних зображень.

Для провдення тестових досліджень запропонованої програмної системи необхідно сформувати набори тестових вибірок. Достовірність та об’єктивність отриманих результатів напряму залежить від того наскільки коректно та повноцінно буде сформована вибірка. Для об’єктивного проведення тестування було запропоновано проведення ряду тестів. Під час формування тестових наборів враховувались різні фктори які могли вплинути на кінцевий результат. Серед таких параметрів слід зазначити:

* кількість людей на зображенні;
* рівень відкритисті обличчя;
* рівень зашумленості зображення;
* рівень освітленості під час фотофіксації;
* відстань до обличчя людей.

Окрім того, під час проведення тестування було виставлено ряд додаткових критеріїв, на основі яких було проведено поділ тестових вхідних даних на такі на окремі групи:

* прості зображення – в даний тестовий набір додавались зображення які містили тільки одну людину, обличчя не має ніяких природніх спотворень чи дефектів, обличчя займає більше 50% площі зображення, відсутні додаткові чинники закритості обличчя (одяг, маски, окуляри тощо), на зображенні відсутні шуми та інші сторонні артефакти, зображення було отримано при високому рівні освітлення. Дана група зображень відображає найякісніше зроблені фотографії та вимагає мінімальних додаткових операцій по її обробці;
* звичайні зображення – на зображеннях даної групи можуть перебувати від 1 до 5 людей, усі люди повернуті обличчям до об’єктиву камери, кути нахилу або повороту голови є мінімальними, відсутні або мінімально видимі додаткові чинники закритості обличчя (одяг, маски, окуляри тощо), на зображенні присутні невеликі шуми, зображення було отримано при використанні природнього освітлення в різні пори дня. Дана група зображень відображає випадок отримання зображень в природніх умовах при фотографуванні простими фотоапаратами. В загальному випадку зображення даного їгупи вимагають додатквої попередньої обробки;
* складні зображення – зображеннях даної групи можуть містити більше 5 людей, при цьому люди можуть бути розташовані як обличчям до обєктиву так в спиною чи боком, кут нахилу або повороту голови теж є довільним, площа облич окремих людей може значно відрізнятись одна від одної (люди знаходяться на різній відстані від фотографа), на обличчях можуть бути присутні елементи одягу або інші предмети які закривають значніу частину обличчя, на зображенні можуть бути присутні шуми, рівнь освітлення не однорідний та може суттяво змінюватись взалежності від області фотографії. Дана група зображень відображає випадоки створення не професійних фотографій або випадки коли люди птрапляють в кадр випадково, що створює додаткові проблеми при роботі з цифровими зображеннями. В загальному випадку зображення даногої групи вимагають додаткової попередньої обробки.

В результаті аналізу отриманих результатів тестування можна зробити наступні висновки. При аналізу зображень які належться до першої групи резуьтати як і очікувалось були найкращі. Похибок під час детекції облич не було виявлено, всі описуючі прамокутники максимально точно описували людські обличчя, контури овалів однорідних областей не викликають двозначного трактування, кількісні показники чітко відображеють параметри області яка відповідає обличчу людини. При аналізу результатів які надала програма після оброки зображень другої групи слід відмітити появу хибно виділених об’єктів, проте була помічена закономірність, що чим чіткіше можна відрізнити облччя на зображенні тим точнішим був результат, при цьому сторонні фактори не мали такого серйозного впливу. Наприклад рівень освітлення або наявність шумів мало впливар на кінцевий результат, а от наявність одягу або інших предметів могло заначно спотрюювати кінцевий результат. При обробці зображень третьої групи було виявлено значні похибки кінцевого результату. Серед основних чинників які впливали на кінцевий результат були: наявність одягу та значний поворот або нахил голови. В першому випадку проблему вирішити складно, бо фактично незрозуміло чи ця область дійсно відповідає обличчю чи навпаки це є елемент зображення, що не відносить до класу «обличчя». Частково вдалось вирішити дану проблему за рахунок використання значення параметру кольора одноріжної області, адже можна припустити, що колір людської шкіри має деякий обмежений набір. Тому одноріжні області наприклад червоного або зеленого кольорів одразу можна відкидати, а рожевого чи чорного відправляти на додатковий аналіз оператором. Другу пробелму можна вирішити якщо над однорідною областю провести додаткове афінне перетворення – здійснити поворот на кут, який би забезпечив розташування одноріжної області таким чином, що центральна віть буда перпендикулярною до осі абсцис. Проте інші фактори на зображенні також вносять свої похибки в кінцевий результат роботи програми, тому зображення даного типу при аналізу показували найгірші результати.

Отримані результати підтвердили припущення, які були зроблені на етапі моделювання, що розроблений програмний код аналізу цифрових зображень з елементами детекцї облич має високу швидкодію, та пороводить опрацювання завантажених зображень з ріхними характеристиками, а отримані результати відповідають очікуваним. Окрім того за результатами тестування в програму було додано ряд функцій для додаткового опрацювання зображень, що дозволило в кінцевому рузультаті підвищити рівень точності детекції людських облич та зменшити вплив зовнішніх факторів на результати роботи програми.

В результаті проведених експерементів та на основі аналізу отриманих даних була сформована підсумкова таблиця де відображаються отримані результати у згрупованому вигляді. Результати оцінювались якісно та носять невеликий суб’єктивний характер, проте в загальному можна вважали їх достатньо достовірними. Результати наведено в таблиці 3.1

Таблиця 3.1 – Узагальнена таблиця результатів тестування

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Групи зображень/  Хар-ки зображення | Освітлення | Наявність шумів | Відкритість обличчя | Поворот/ нахил голови | Природні утворення |
| Прості зображення | Мінімальний вплив | Мінімальний вплив | Мінімальний вплив | Мінімальний вплив | Відчутний вплив |
| Звичайні зображення | Мінімальний вплив | Відчутний вплив | Відчутний вплив | Відчутний вплив | Відчутний вплив |
| Складні зображення | Відчутний вплив | Відчутний вплив | Максимальний вплив | Максимальний вплив | Відчутний вплив |

Отримані результати в процесі тестування показали правильність обраного шляху розробки програмного додатку. Усі поставлені завдання були виконані, програмна система працювала без помітних проблем, зображення були опрацьовані в режимі реального часу та з достатньою точністю отриманих результатів.

## 3.4 Висновки до розділу

Розроблено та проведено теоретичне моделювання програмного додатку детекції людських облич, що надало можливість програмно реалізувати програмних додаток обробки цифрових зображень з функцією детекції.

Здійснено тестування розробленого додатоку детекції людських облич, на основі виділення ключових точок, що підтвердило правильність у виборі механізму реалізації програми та розробленого алгоритму.

# Висновки

На основі аналізу сучасних програмних систем аналызу та опису цифрових зображень, а також алгоритмыв цифровоъ обробки зображень можна зробити такі висновки:

1. Проведено аналіз та класифікацію завдань та основних етапів цифрової обробки зображень, що дозволило виділити основні задачі аналізу зображень при проектуванні системи моніторингу з елементами детекції об’єктів.
2. Проаналізовано формати зберігання цифрових зображень, що дозволило обрати структури даних для швидкого аналізу та детекції об’єктів на зображеннях.
3. Проведено аналіз наявних програмних засобів для отримання, редагування та аналізу цифрових зображень, що дозволило виділити основні структурні модулі та встановити інтерфейси обміну даними між окремими частинами програмних систем.
4. Проведно аналітичний огляд алгоритмів виділення об’єктів на цифрових зображеннях на основі статистичних, математичних операцій та використанні механізму штучного інтелекту, що дало можливість визначити переваги та недоліки сучасних технологій виділення людських облич.
5. Розроблено алгоритм детекції людських облич на основі виділення ключових точок, що дало можливість розробити структуру та реалізувати програмних додаток детекції людських облич в системі аналізу цифрових зображень.
6. Здійснено тестування розробленого додатоку детекції людських облич, на основі виділення ключових точок, що підтвердило правильність у виборі механізму реалізації програми та розробленого алгоритму.

# Список використаної літератури

1. Сергійчук В.В., Жаборинська І.В. Алгоритм детекції людських облич. Збірник тез VІ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі»., Тернопіль, 10 листопада 2022 р. с. 28.
2. Сергійчук В.В., Жаборинська І.В. Алгоритм розпізнавання та підрахунку людей у відеопотоці. Збірник тез VІ Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі»., Тернопіль, 10 листопада 2022 р. с. 28.
3. Bradski G. Learning OpenCV - Computer Vision with the OpenCV Library, 2018. 580 с.
4. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB, М.: Техносфера, 2016. 616 с.
5. Сафонов В. О. Параметризованные типы данных. История, теория, реализация и применение. М.: Издательство Санкт-Петербургского университета, 2019. 116 c.
6. Секунов Н.Ю. Самоучитель Visual C++ 6.0. М.: СПб: BHV, 2018. 960 c.
7. Страуструп Б. Язык программирования C++. М.: Радио и связь,2019. 350 c.
8. Страуструп Бьерн Дизайн и эволюция С++ М.: ДМК Пресс, 2016. 446c.
9. Сойфер В.А. Методы компьютерной обработки програмних кодов М.: Физматлит, 2020. 784 с.
10. Оппенгейм А. Цифровая обработка сигналов. М.: Техносфера,2016. 856с.
11. Прэтт У. Цифровая обработка изображений. В 2-х книгах: пер. с англ. М.: Мир,2019. 792 с.
12. Грузман И.С. Цифровая обработка изображений в информационных системах: учеб, пособие. 2002. 352 с.
13. Яне Б. Цифровая обработка сигналов. М.: Техносфера, 2017. 584 с.
14. Селянкин В.В. Компьютерное зрение. Анализ и обработка сигналов: учебное пособие. 2019. 152с.
15. Форсайт Д. Компьютерное зрение. Современный подход. 2014. 928 с.
16. Поляков А. Методы и алгоритмы компьютерного анализа. 2019. 545 с.
17. Безрядин С. Н. Преобразование яркости в программном обеспечении. 2016. 211с.
18. Баженова И. Ю. Язык программирования С++ . 2017. 366 с.
19. Бартлетт Н. Программирование на Delphi Путеводитель. 2016. 116с.
20. Вебер Дж. Технология программирования в подлиннике. 2017. 256с.
21. Волш А. И. Основы программирования на С++для World Wide Web. 2016. 458с.
22. Абрамов С. А. Задачи по программированию. 2018. 256 с.
23. Березин Б.И. Начальній курс программирования на языках высокого уровня. 2016. 331с.
24. Бондарев В.М. Основы программирования. 2017. 446с.
25. Вирт Н. Алгоритмы и структуры данных. 2019. 345с.
26. Гладков В. П. Задачи по информатике на вступительном экзамене в вуз и их решения: Учебное пособие. 2021. 516с.
27. Грогоно П. Программирование на языке Java. 2022. 216с.
28. Дагене В.А. 100 задач по программированию. 2019. 106с.
29. Джамса К. Библиотека программиста Java. 2019. 656с.
30. I.Merino, J. Azpiazu 2D Features Detector And Descriptor Selection Expert System. 2019. – С. 51–61.
31. Заварикин В.М. Основі информатики и вычислительной техники. 2019. 556с.
32. Кен А. Язык программирования С++. 2017. 378с.
33. Керниган Б. Язык программирования Java. 2022. 391с.
34. Ляхович В.Ф. Руководство к решению задач по основам информатики и вычислительной техники. 2014. 127с.
35. Мейнджер Дж. С++ Осноы программирования. 2017. 346с.
36. Миков А. И. Информатика. Введение в компьютерные науки. 2018. 442с.
37. Могилев А. В. Информатика: Учеб. пособие для студ. 2019. 629с.
38. Нотон П. JAVA:Справ.руководство. 2016. 447с.
39. Нотон П. Полный справочник по Java. 2017. 556с.
40. Ренеган Э.Дж. 1001 адрес WEB для программистов:Новейший путеводитель программиста по ресурсам World Wide Web. 2021. 512с.
41. Томас М. Секреты программирования для Internet на Java. 2017. 396с.
42. Семакина И. Г. Информатика. Задачник-практикум. 2019. 476с.
43. Сокольский М.В. Все об Intranet и Internet. 2018. 254с.
44. Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. 2021. 56с.
45. Тюрин Ю.Н. Анализ данных на компьютере. 2019. 384с.
46. Флэнэген Д. Java in a Nutshell. 2018. 473с.
47. Чен М.С. Программирование на С++:1001 совет:Наиболее полное руководство по С++ и Visual С++. 2017. 640с.
48. Эферган М. С++: справочник. QUE Corporation.2018. 256с.
49. Sinha P. Perceiving and Recognizing threedimensional forms. 2016. 278р.
50. Pablo F. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. 2013. 187с.
51. Березький О.М., Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп’ютерна інженерія. Магістерська програма - Комп’ютерна інженерія". Тернопіль: ЗУНУ, 2022. 32 с.
52. Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» Тернопіль: ТНЕУ, 2019. 33 с.