НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри СПСКС

_____
В.П.Тарасенко_
(підпис) (ініціали, прізвище)

"___"_____2018р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія (Комп'ютерні системи та компоненти)
на тему МОДИФІКОВАНИЙ МЕТОД ВИЯВЛЕННЯ ЧАСТИН ТІЛА ЛЮДИНИ НА ЗОБРАЖЕННЯХ

Виконав: студент ІІ курсу, групи КВ-71мп
(шифр групи)

Поліщук Михайло Олегович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Петрашенко А.В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра комп'ютерні системи та компоненти

Рівень вищої освіти – другий (магістерський)
Спеціальність 123 Комп'ютерна інженерія (Комп'ютерні системи та компоненти)

ЗАТВЕРДЖУЮ
Завідувач кафедри СПСКС
_____ _В.П.Тарасенко__
(підпис) (ініціали, прізвище)

«___»_____2018р.

ЗАВДАННЯ
на магістерську дисертацію студента
Поліщук Михайла Олеговича

1. Тема дисертації: Модифікований метод виявлення частин тіла людини на зображенні, науковий керівник дисертації: д.т.н., доцент Петрашенко,
затверджені наказом по університету від « 30» жовтня 2018 р. № 4030-с.

2. Термін подання студентом дисертації: 7 грудня 2018 р.

3. Об'єкт дослідження: процес розпізнавання тіла людини та його частин на зображенні.

4. Предмет дослідження: методи підвищення ефективності розпізнавання частин тіла людини.

5. Перелік завдань, які потрібно розробити:
- розглянути основні методи та інструменти розпізнавання образів на зображеннях;
- провести порівняльний аналіз методів та дослідити їх переваги та недоліки;
- дослідити програмно реалізовані методи на основі згорткових мереж;
- проаналізувати недоліки та способи модифікації методі розпізнавання;
- написати модифікацію для методу розпізнавання;
- провести тести для методу до і після модифікації на зображеннях поганої якості;
- проаналізувати отримані результати.

6. Перелік ілюстративного матеріалу:
- Структурна схема архітектури програмної системи;

7. Перелік публікацій: «Модифікований метод виявлення частин тіла людини на зображеннях», XI наукова конференція молодих вчених «Прикладна математика та

комп'ютинг» ПМК-2018-2 (Київ, 14-16 листопада 2018 р.); «Способи модифікації обробки зображень для згорткової нейронної мережі», V Міжнародна науково-технічна конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (Київ, 22-23 листопада 2018 р.).

8. Дата видачі завдання 5 вересня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з предметною галуззю	17.12.2017	
2	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	19.02.2018	
3	Аналіз існуючих джерел і початок роботи над першим розділом	19.03.2018	
4	Аналіз існуючих програмних засобів	29.06.2018	
5	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження та другим розділом	18.09.2018	
6	Робота над програмним засобом; робота над третім розділом магістерської дисертації; підготовка матеріалів доповіді на конференції «Прикладна математика та комп'ютинг» ПМК-2018-2	9.10.2018	
7	Завершення роботи над магістерської дисертації; підготовка ілюстративного матеріалу	17.10.2018	
8	Оформлення текстової і графічної частини магістерської дисертації	21.11.2018	
9	Попередній розгляд магістерської дисертації на кафедрі	26.11.2018	

Студент _____ Поліщук М.О.

Науковий керівник дисертації _____ Петрашенко А.В.

РЕФЕРАТ

Актуальність теми. Комп'ютерний зір – це сучасний напрямок технологій який має широкий потенціал для використання у великій кількості сфер діяльності. Перспективним напрямом є саме розпізнання людського тіла та його частин. Такі технології могли б широко використовуватися у медицині, системах відеоспостереження або у кіноіндустрії, щоб не використовувати дорогі прилади захоплення руху.

Об'єкт дослідження – методи розпізнавання об'єктів.

Предмет дослідження – пошук шляхів для модифікації розпізнавання частин людського тіла у згортковій неронній мережі, огляд та аналіз всіх кроків згорткової нейронної мережі.

Мета роботи: модифікація методу розпізнавання частин тіла людини для підвищення ефективності обробки темних зображень з низькою деталізацією.

Наукова новизна полягає у створенні методу який дозволяє ефективніше розпізнавати образи частин тіла людини на зображеннях з поганою деталізацією у порівнянні з існуючими аналогами розпізнавання.

Практична цінність одержаних в роботі результатів можуть бути використані для вибору шляхів модифікації існуючих методів розпізнавання частин тіла людини.

Апробація роботи. Основні результати роботи були представлені та обговорювались на XI науковій конференції молодих вчених «Прикладна математика та комп'ютинг» ПМК-2018-2 (Київ, 14-16 листопада 2018 р.), а також на V Міжнародній науково-технічній конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами» (Київ, 22-23 листопада 2018 р.).

Структура та обсяг роботи. Магістерська дисертація складається з вступу, трьох розділів та висновків.

У *вступі* подано загальну характеристику роботи, описано сучасні методи розпізнавання і перспектив розвитку комп'ютерного зору.

У *першому розділі* наведено загальний огляд методів розпізнавання за допомогою нейронних мереж, їх основні характеристики, архітектури та особливості, а також обгрунтовано чому був обраний метод на основі згорткових мереж.

У *другому розділі* наведено основні недоліки розпізнавання частин людського тіла,

можливі модифікації методу розпізнавання.

У *третьому розділі* проаналізовано результати тестів до і після модифікацій.

У *висновках* представлені результати проведеної роботи.

Робота представлена на 86 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: РОЗПІЗНАВАННЯ ЧАСТИН ТІЛА, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА.

Актуальность темы. Компьютерное зрение - это современное направление технологий который имеет широкий потенциал для использования в большом количестве сфер деятельности. Перспективным направлением является именно распознавание человеческого тела и его частей. Такие технологии могли бы широко использоваться в медицине, системах видеонаблюдения или в киноиндустрии, чтобы не использовать дорогие приборы захвата движения.

Объект исследования - методы распознавания объектов.

Предмет исследования - поиск путей для модификации распознавания частей человеческого тела в згорткових неронний сети, обзор и анализ всех шагов згорткових нейронной сети.

Цель работы: модификация метода распознавания частей тела человека для повышения эффективности обработки темных изображений с низкой детализацией.

Научная новизна заключается в создании метода который позволяет эффективно распознавать образы частей тела человека на изображениях с плохой детализацией по сравнению с существующими аналогами распознавания.

Практическая ценность полученных в работе результатов могут быть использованы для выбора путей модификации существующих методов распознавания частей тела человека.

Апробация работы. Основные результаты работы были представлены и обсуждались на XI научной конференции молодых ученых «Прикладная математика и компьютинг» ПМК-2018-2 (Киев, 14-16 ноября 2018), а также на V Международной научно-технической конференции «Современные методы, информационное, программное и техническое обеспечение систем управления организационно-техническими и технологическими комплексами» (Киев, 22-23 ноября 2018).

Структура и объем работы. Магистерская диссертация состоит из введения, трех глав и выводов.

Во введении представлена общая характеристика работы, описаны современные методы распознавания и перспектив развития компьютерного зрения.

В первой главе приведен общий обзор методов распознавания с помощью нейронных сетей, их основные характеристики, архитектуры и особенности, а также обоснованно почему был выбран метод на основе сверточный сетей.

Во втором разделе приведены основные недостатки распознавания частей

человеческого тела, возможные модификации метода распознавания.

В третьем разделе проанализированы результаты тестов до и после модификаций.

В выводах представлены результаты проведенной работы.

Работа представлена на 86 листах, содержит ссылки на список использованных литературных источников.

Ключевые слова: РАСПОЗНАВАНИЕ ЧАСТЕЙ ТЕЛА, СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ.

Actuality of theme. Computer vision is a modern technology trend that has wide potential for use in a wide range of fields of activity. A promising direction is the recognition of the human body and its parts. Such technologies could be widely used in medicine, video surveillance systems or in the cinema industry, in order not to use expensive motion capture devices.

Object of research - methods of object recognition.

The subject of the study is the search for ways to modify the recognition of parts of the human body in the convolutional neural network, review and analysis of all steps of the convolutional neural network.

Purpose: to modify the method of recognizing human body parts to improve the processing efficiency of dark images with low detail.

The scientific novelty consists in the creation of a method that allows more efficient recognition of images of parts of the human body in images with poor detail compared with existing analogues of recognition.

The practical value of the results obtained in the work can be used to select ways to modify existing methods for recognizing human body parts.

Test work. The main results of the work were presented and discussed at the XI Scientific Conference of Young Scientists "Applied Mathematics and Computer", PMK-2018-2 (Kyiv, November 14-16, 2018), as well as at the V International Scientific and Technical Conference "Modern Methods , information, software and technical support of control systems for organizational, technical and technological complexes "(Kyiv, November 22-23, 2018).

Structure and scope of work. The master's dissertation consists of an introduction, three sections and conclusions.

The introduction gives a general description of the work, describes the modern methods of recognition and the prospects for the development of computer vision.

The first section provides a general overview of methods for recognizing with neural networks, their main characteristics, architecture and features, and also why the method was chosen.

The second section presents the main disadvantages of recognizing parts of the human body, possible modifications to the method of recognition.

The third section analyzes the test results before and after the modifications.

The conclusions are the results of the work.

The work is presented on 86 pages, contains a reference to the list of used literary sources.

Keywords: HUMAN BODY PARTYS DETECTING, CONVOLUTIONAL NEURAL NETWORK.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ, ОДИНИЦЬ І ТЕРМІНІВ

ШНМ – штучна нейронна мережа

Net – network (мережа)

ReLU – Rectified Linear Unit

CONV – convolution (згортка)

FCL – fully connected layer (шар повного з’єднання)

CNN – convolutional neural network (згорткова нейронна мережа)

ВСТУП

Сучасні методи розпізнавання об'єктів на відео або зображеннях мають широкий спектр використання починаючи від захисту приватних територій до створення спецефектів у кінолентах. Щодня великі компанії та невеликі команди ентузіастів створюють нові підходи та алгоритми обробки кадрів аби розвивати технології комп'ютерного зору.

Комп'ютерний зір, як напрям, тільки починає свій шлях розвитку. Перспективи його розвитку мають великий потенціал. Так великі приватні компанії вже використовують технології розпізнавання і аналізу даних у багатьох свої продуктах і успішно займаються продажем своїх продуктів та технологій. У більшості випадках можливості технологій розпізнавання мають великий ефект на покращенні та збільшенні функціоналу продукції для кінцевого користувача.

Більшість методів розпізнання мають велику кількість недоліків. Основними недоліками є те, що для обробки даних потрібні великі обчислювальні ресурси, та навіть за великих обчислювальних ресурсах часто результати розпізнавання бувають некоректними. Однією з найбільших проблем комп'ютерного зору є саме робота над розпізнанням людського тіла та його частин. Наше тіло може бути у найрізноманітніших положеннях на зображенні, що підвищує скаладність роботи із розпізнанням потрібних нам об'єктів. Але це стає поштовхом для пошуку все нових і нових шляхів для розвитку комп'ютерного зору і методів розпізнання об'єктів.

Саме пошук більш альтернативних методів розпізнання частин тіл людини та пошуків модифікації таких методів і стало ціллю данного дослідження.

1. ТЕОРЕТИЧНИЙ ОГЛЯД ОБЛАСТІ КОМП’ЮТЕРНОГО ЗОРУ

1.1 Поняття комп’ютерного зору

Як людина, ми усвідомлюємо тривимірну структуру навколишнього світу легко. Ви можете визначити форму та прозорість кожного об’єкту, тонкі закономірності світла і затінення, які грають по всій її поверхні і легко відрізняються. Дивлячись на кадровий портрет групи,

ви можете легко підрахувати (і назвати) всіх людей у зображенні та навіть здогадатися їх емоції за зовнішнім виглядом. Вчені проводять десятиліття, намагаючись зрозуміти, як функціонує візуальна система, і хоча вони можуть припускати про оптичні ілюзії, щоб зрозуміти деякі його принципи, остаточне рішення цієї загадки залишається незрозумілим.

Дослідники комп'ютерного бачення розвивали паралельно математичні методи для створення тривимірної форми та появи предметів на зображенні. Тепер існують надійні методи для точного обчислення часткової 3D-моделі середовища з сотнями тисяч точок. Зараз ми можемо відслідковувати людину, що рухається напроти певного фону. Ми можемо навіть з помірним успіхом спробувати знайти і назвати всіх людей на фотографії, використовуючи форми облччя, одягу та волосся. Однак, незважаючи на всі ці досягнення, мрія про те, що комп'ютер інтерпретує зображення на тому ж рівні, що і людина чи тварина залишається недосягненою. Частково розпізнавання нам дається важко тому, що бачення - це зворотна проблема, в якій ми прагнемо відновити деякі невідомі дані, що даються з недостатньою кількістю інформації для повної генерації рішення.

Моделі розпізнавання неоднозначно розглядають потенційні рішення без потрібної кількості початковиї параметрів. Проте моделювання візуального світу у всій своїй різноманітності набагато складніше, ніж, скажімо, моделювання голосового тракту, що генерує певні звуки.

Моделі, які ми використовуємо при комп'ютерному зорі, зазвичай розробляються в фізиці (оптика, сенсорний дизайн тощо) та комп'ютерній

графіці. Обидва ці напрями мають задачу моделювати як об'єкти рухаються та контактують, як приклад світло, що відбивається від їх поверхонь, розсіюється атмосферою, переломлюється через лінзи камери (або людські очі), і нарешті проеціюється на квартиру (або вигнутий) площину зображення.

У комп'ютерному баченні ми намагаємося описати світ, який ми бачимо в одному або декількох зображеннях і описати його властивості, такі як форма, освітленість та колір. Алгоритми зору настільки схильні до помилок, що часто невелика зміна у пікселях зображення призводить до кардинально інакших результатів.

Таким чином комп’ютерне бачення – це набір методів та алгоритмів для взаємодії з візуальним кавколишнім середовищем, ціллю яких є розпізнавання та аналіз графічних об’єктів.

1.2 Задача розпізнавання

Людина звичайно може легко розрізняти звук образи, звуки, аромати. Проте для комп'ютера важко вирішити такі проблеми сприйняття. Ці проблеми важкі, тому що кожен шаблон зазвичай містить велику кількість інформації, і проблеми розпізнавання зазвичай мають неясну, високовимірну структуру.

Розпізнавання образів - це наука робити висновки з даних, що були отримані шляхом сприйняття, використовуючи інструменти зі статистики, ймовірності, обчислювальної геометрії, машинного навчання, обробки сигналів та алгоритмів. Таким чином, це має найважливіше значення для штучного інтелекту та комп'ютерного бачення, і має далекосяжні застосування в галузі машинобудування, науки, медицини та бізнесу. Зокрема, досягнення, досягнуті протягом останнього півстоліття, дозволяють комп'ютерам ефективніше взаємодіяти з людьми та природним світом (наприклад, програмне забезпечення для розпізнавання мовлення). Проте найважливіші проблеми розпізнавання образів ще не вирішені.

Цілком природно, що ми повинні прагнути спроектувати і побудувати машини, здатні розпізнавати шаблони. Від автоматичного розпізнавання мови, ідентифікації відбитків пальців, оптичного розпізнавання символів, ідентифікації послідовності ДНК та багато іншого, зрозуміло, що надійне, точне розпізнавання образів за допомогою машини буде надзвичайно корисним. Більше того, при вирішенні невизначеного числа проблем, необхідних для побудови таких систем, ми отримуємо глибше розуміння та оцінку систем розпізнавання образів. Для деяких завдань, таких як мова та візуальне розпізнавання, на наші проектні зусилля насправді можуть впливати знання про те, як вони вирішуються в природі, як в алгоритмах, які ми використовуємо, так і в дизайні спеціального устаткування.

1.3 Основні поняття задачі розпізнавання

Ознаки (features) можна визначити як будь-який відмітний аспект, якість або характеристику, які можуть бути символічними (наприклад, колір) чи чисельними (наприклад, висота). Комбінація функцій d представляється як вектор d-розмірного стовпця, який називається вектором властивостей. D-мірний простір, визначений функцією вектора, називається *простором ознак (feature space)*. Об'єкти представлені у вигляді точок у просторі ознак. Це подання називається графіком розсіювання (scatter plot) [4].

Шаблон (pattern) визначається як комбінація ознак, характерних для особи. У класифікації шаблон - це пара змінних *{x, w}*, де *x* - це сукупність спостережень або ознак (вектор ознак), а *w* - концепція, що лежить в основі спостереження (label). Якість вектора властивостей пов'язана з його здатністю відрізняти приклади з різних класів (рис. 1.1). Приклади з того ж класу повинні мати подібні значення ознак, а приклади з різних класів мають різні значення ознак.

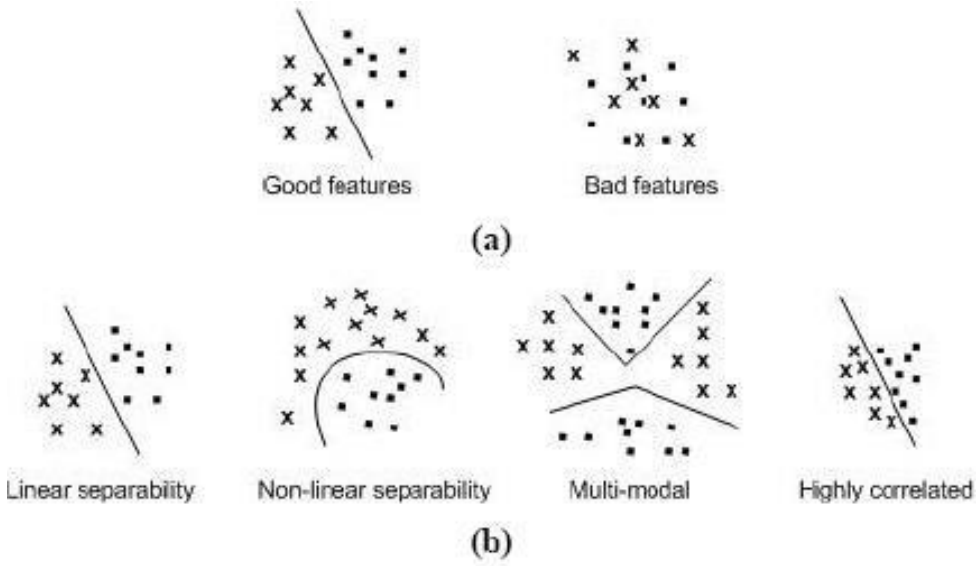

Рисунок 1.1 - Ознаки (характеристики): a - відмінність між хорошими (good) та поганими (bad) ознаками; b - властивості ознак за способом розділення: лінійні, нелінійні, мультимодальні, висококорельовані. [5]

Метою класифікатора є розділення простору ознак на позначені класом області прийняття рішень. Межі між регіонами прийняття рішень називаються межами рішень (рис. 1.2).

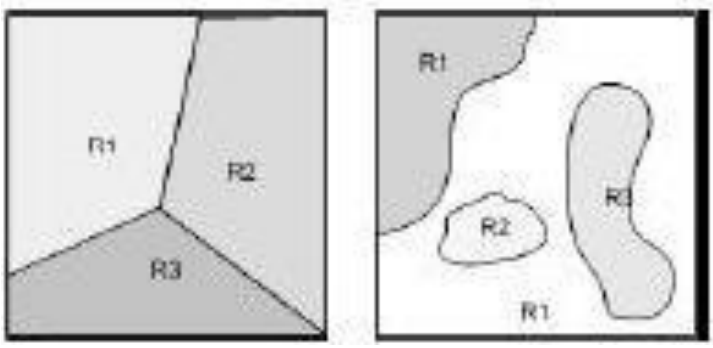

Рисунок 1.2 - Межі класифікатора та рішення. [5]

Якщо характеристики або атрибути класу відомі, окремі об'єкти можуть бути ідентифіковані як такі, що належать або не належать до цього класу. Об'єкти відносяться до класів, дотримуючись шаблонів відмінних характеристик і порівнюючи їх з типовими членами кожного класу. Розпізнавання образів передбачає виділення образів з даних, їх аналізу та,

нарешті, ідентифікації категорії (класу), до якої належить кожний образ. Типова система розпізнавання образів містить датчик, механізм попередньої обробки (сегментація), механізм виділення ознак (ручний або автоматичний), алгоритм класифікації або опису та набір прикладів (тренувальний набір), які вже класифіковані або описані (після обробки) (рис. 1.3).

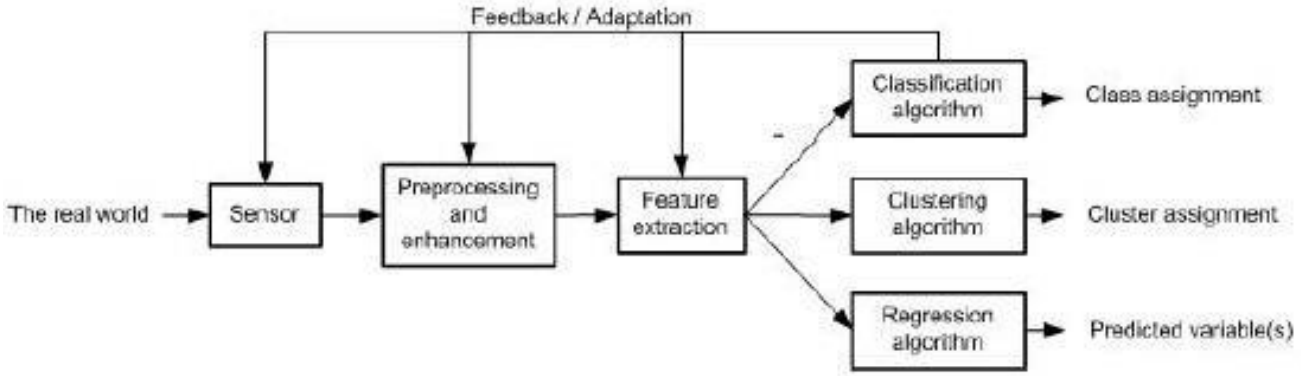

Рисунок 1.3 – Система розпізнавання образів [5]

1.4 Підходи до розпізнавання образів

В області розпізнавання образів існує два основних підходи, перший це статистичний, а другий структурний. Звичайно часто використовують також і гібридні методи розпізнавання, які мають властивості обох підходів. Гібрибні підходи мають також назву уніфіковані підходи, Кожен з підходів має свої алгоритми опису структур і класифікацій [6].

Статистичне моделювання базується на основі теорії статистичних рішень, аби виділяти різні групи на основі певних характеристик даних. Є великий спектр можливих статистичних методів, що використовують для пошуку характерних ознак даних. Статистичні методи можуть бути простою функцією опису або складним математичним перетворенням. Так наприклад статистичні методи можуть включати в себе перетворення Фур'є, вейвлет-перетворення, перетворення Кархунена-Лоуєва або трансформації

Хога. Кількісні характеристики, виділені з кожного об'єкта для статистичного розпізнавання образів, організовані у вектор-функцію з фіксованою довжиною, коли значення, пов'язане з кожною функцією, визначається його положенням у векторі (тобто перша ознака описує певну характеристику даних, друга ознака описує іншу характеристику тощо). Колекція векторів функцій, породжених завданням опису, передається до завдання класифікації. Статистичні методи, що використовуються в якості класифікаторів у межах задачі класифікації включають ті, які базуються на подібності (наприклад, відповідність шаблону, k-найближчого сусіда), ймовірність (наприклад, правило Байєса), межі (наприклад, дерева рішень, нейронні мережі) та кластеризація (наприклад, k-середні, ієрархічний).

Кількісний характер статистичного розпізнавання образів ускладнює виділення різниці між групами на основі морфологічних (тобто форми або структурних) підшаблонів та їх взаємозв'язків, вбудованих в дані. Це обмеження дало імпульс розвитку структурного підходу до розпізнавання образів, який підтверджується психологічними свідченнями, що стосуються функціонування людського сприйняття та пізнання. Розпізнавання об'єктів в людях було продемонстровано для залучення психічних уявлень до явних, структурно-орієнтованих характеристик об'єктів, і було зроблено висновок, що рішення про класифікацію людини складаються з урахуванням ступеня подібності між витягнутими ознаками та прототипом, розробленим для кожної групи. Наприклад, визнання за теорією компонентів пояснює процес розпізнавання образів у людей: (1) об'єкт сегментований у окремі області за границями, визначеними різними поверхневими характеристиками (наприклад, яскравістю, текстурою та кольором); (2) кожна сегментована область апроксимується простою геометричною формою, і (3) об'єкт визначається на підставі подібності у складі між геометричним представленням об'єкта та центральною тенденцією кожної групи. Це теоретичне функціонування людського сприйняття та пізнання служить основою для структурного підходу до розпізнавання образів.

Розпізнавання структурних образів, яке іноді називають розпізнаванням синтаксичних образів через його походження в формальній теорії мовлення, спирається на синтаксичні граматики, щоб розрізняти дані різних груп на основі морфологічних взаємозв'язків (або взаємозв'язків), що містяться в даних. Структурні ознаки, часто називаються примітивними, представляють собою підшаблони (або структурні блоки) та відносини між ними, які складають дані. Семантика, пов'язана з кожною функцією, визначається схемою кодування (тобто вибіркою морфології), що використовується для ідентифікації примітивів у даних. Вектори ознак, породжених структурними системами розпізнавання образів, містять змінюване число ознак (по одній для кожного примітиву, витягнутого з даних), щоб врахувати наявність надлишкових структур, які не впливають на класифікацію. Оскільки взаємовідносини між видобутими примітивами також повинні бути закодовані, векторний компонент повинен включати додаткові функції, що описують відносини між примітивами або приймають альтернативну форму, наприклад, реляційний граф, який може бути проаналізовано синтаксичною граматикою.

Акцент на відносинах між даними робить структурний підхід до розпізнавання образів найбільш розумним для даних, які містять наслідувальну ідентифіковану організацію, таку як дані зображень (які організовуються за місцем розташування всередині візуального рендеринга) та дані про часові ряди (організовані за часом); дані, що складаються з незалежних зразків кількісних вимірювань, не мають упорядкування і вимагають статистичного підходу. Методології, що використовуються для виділення структурних ознак з даних зображення, таких як технології обробки зображення, призводять до таких примітивів, як ребра, криві та регіони; Технологія вилучення функції для даних часової серії включає в себе ланцюгові коди, кусочно-лінійну регресію та фігуру кривої, які використовуються для генерації примітивів, які кодують послідовні, упорядковані за часом співвідношення. Завдання класифікації доходить до

ідентифікації, використовуючи синтаксичний аналіз: вилучені структурні ознаки ідентифікуються як представники певної групи, якщо вони можуть бути успішно проаналізовані синтаксичною граматикою. При розрізненні більш ніж двох груп синтаксична граматика необхідна для кожної групи.

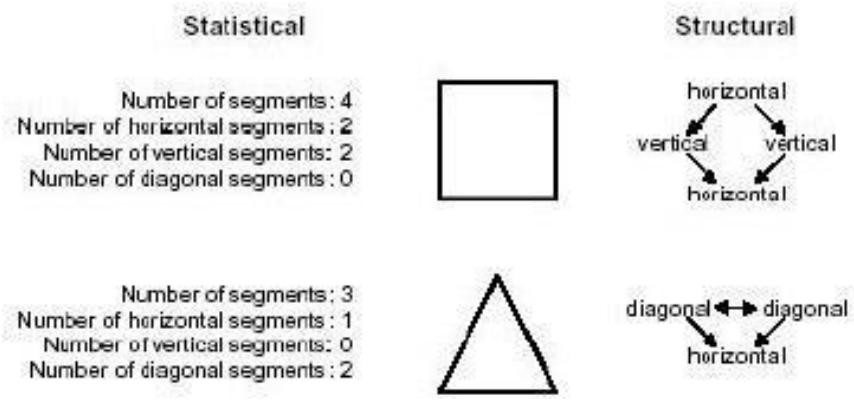

Рисунок 1.4 - Статистичні та структурні підходи до розпізнавання образів, що застосовуються до загальної проблеми ідентифікації.

На рисунку 1.4 зображено, як обидва підходи можуть бути застосовані до однієї проблеми ідентифікації. Мета полягає в тому, щоб диференціювати квадрат і трикутник. Статистичний підхід витягує кількісні характеристики, такі як кількість горизонтальних, вертикальних та діагональних сегментів, які потім передаються класифікатору, що вирішує, до якого класу віднести виділені ознаки. Структурний підхід виділяє морфологічні особливості та їх взаємозв'язки в межах кожної фігури. Використовуючи прямолінійний сегмент як елементарну морфологію, створюється реляційний граф і класифікується за допомогою визначення синтаксичної граматики, яка може успішно проаналізувати реляційний граф. У цьому прикладі як статистичні, так і структурні підходи зможуть точно розрізняти дві геометрії. Однак у більш складних даних дискримінаційність безпосередньо впливає на особливий підхід, використовуваний для розпізнавання образів, оскільки отримані ознаки представляють різні характеристики даних.

Суттєвими відмінностями між статистичними та структурними

підходами є: (1) опис, сформований статистичним підходом, є кількісним, тоді як структурний підхід створює опис, що складається з підшаблонів або будівельних блоків; і (2) статистичний підхід розрізняє, ґрунтуючись на численних відмінностях між ознаками різних груп, тоді як граматики використовуються структурним підходом для визначення мови, що охоплює прийнятні конфігурації примітивів для кожної групи. Гібридні системи можуть поєднувати два підходи як спосіб компенсувати недоліки кожного підходу, зберігаючи при цьому переваги кожного. Як система одного рівня, структурні ознаки можуть використовуватися як зі статистичними, так і з структурними класифікаторами. Статистичні ознаки не можуть бути використані з структурним класифікатором, оскільки вони не мають реляційної інформації, однак статистична інформація може бути пов'язана з структурними примітивами і використовується для вирішення невизначеностей під час класифікації (наприклад, як при розборі з приписаними граматиками) або безпосередньо вкладені безпосередньо в самий класифікатор (наприклад, як при розборі з стохастичними граматиками). Гібридні системи також можуть об'єднувати два підходи в багаторівневу систему, використовуючи паралельну або ієрархічну композицію.

1.5 Системи розпізнавання образів

У системі розпізнавання образів виділяють три різні операції: попередньої обробки, вилучення властивостей та класифікацію. Щоб зрозуміти проблему проектування такої системи, ми повинні розуміти проблеми, які необхідно вирішити кожному з цих компонентів [3].

Датчик. Вхід системи розпізнавання образів часто є свого роду перетворювачем, таким як камера або мікрофон. Складність проблеми може цілком залежати від характеристик і обмежень датчика - її пропускної здатності, роздільної здатності, чутливості, спотворення, співвідношення сигнал / шум, затримки тощо.

Сегментація та групування. На практиці предмети, які необхідно розпізнати, часто перетинаються, і система повинна буде визначити, де закінчується один предмет, а де наступний - індивідуальні зразки повинні бути сегментовані. Якщо ми вже виділили цей предмет, то було б простіше сегментувати його зображення. Як ми можемо сегментувати зображення, перш ніж вони будуть класифіковані, або класифікувати їх, перш ніж вони будуть сегментовані? Потрібен спосіб дізнатися, коли ми перейшли з однієї моделі в іншу, або знати, коли ми просто маємо фонову категорію або не маємо жодної категорії. Сегментація є однією з найглибших проблем розпізнавання образів. Трудно пов'язана з проблемою сегментації - це проблема розпізнавання або групування різних пасток композитного об'єкта.

Виділення ознак. Концептуальна межа між виділенням ознак та правильною класифікацією є дещо умовною: ідеальна функція виділення ознак дає представлення, яке робить роботу класифікатора тривіальною; навпаки, всемогутній класифікатор не потребує допомоги витонченої функції виділення ознак. Різниця виникає з практичних, а не з теоретичних причин.

Традиційна мета виділення ознак полягає в тому, щоб характеризувати об'єкт, який слід визнати вимірами, значення яких дуже схожі для об'єктів однієї категорії, і дуже різні для об'єктів у різних категоріях. Це призводить до ідеї пошуку відмінних функцій, інваріантних до несуттєвих перетворень входу. Загалом, функції, які описують такі властивості, як форма, колір та багато видів текстур, інваріантні для перекладу, обертання та масштабу.

Більш загальна інваріантність буде для обертання щодо довільної лінії в трьох вимірах. Образ навіть такого простого об'єкта, як чашка для кави, зазнає кардинальної зміни, оскільки чашка повертається до довільного кута. Ручка може бути прихована іншою частиною. Крім того, якщо відстань між чашкою та камерою може змінюватися, зображення піддається проекційному спотворенню. Як ми можемо забезпечити, щоб функції були

інваріантні для таких складних перетворень? З іншого боку, чи слід визначити різні підкатегорії для зображення чашки та досягти інваріантності обертання при більш високому рівні обробки?

Як і в процесі сегментації, в завданні з вилучення ознак набагато більше проблем - це залежить від домену, ніж є належним класифікацією, і, отже, вимагає знання домену. Хороший класифікатор для сортування риби, ймовірно, буде мати мало користі для ідентифікації відбитків пальців або класифікації мікрофотографії з клітин крові. Проте деякі принципи класифікації моделей можуть бути використані при проектуванні екстрактора властивостей.

1.6 Класифікатори

Завдання відповідної компоненти класифікатора повної системи полягає в тому, щоб використовувати векторний функціонал, який забезпечує екстрактор функцій, для присвоєння об'єкту категорії. Оскільки досконала класифікація часто неможлива, більш загальним завданням є визначення імовірності кожної з можливих категорій. Абстракція, що забезпечується функціонально-векторним представленням вхідних даних, дозволяє розробляти в основному незалежну від домену теорію класифікації.

Ступінь складності проблеми класифікації залежить від мінливості значень об'єктів для об'єктів у тій же категорії відносно різниці значень об'єктів для об'єктів у різних категоріях. Варіанти значень об'єктів для об'єктів у тій самій категоріїможуть бути пов'язані із складністю і можуть бути пов'язані із шумом. Ми визначаємо шум у дуже загальних термінах: будь-яка властивість чутливого шаблону, який не пов'язаний з істинною базовою моделлю, а навпаки, випадковості в світі або сенсорами. Усі нетривіальні рішення та проблеми розпізнавання образів пов'язані з шумом у певній формі.

Одна з проблем, яка виникає на практиці, полягає в тому, що не завжди

можливо визначити значення всіх ознак для певних вхідних даних. Як це повинен компенсувати категоризатор? Оскільки наша функція розпізнавання з двома функціями ніколи не мала значення критерію *x**, що змінювалось з однією мінливістю, визначене в очікуванні можливої відсутності функції, як він приймає найкраще рішення, використовуючи тільки наявну функцію? Найвищий спосіб просто припустити, що значення пропущеної функції дорівнює нулю, або середнє значення для вже відомих шаблонів є доказово неоптимальним.

1.7 Нейронні мережі

Людська зорова система є одним з чудес світу. Більшість людей легко визнають цифри, такі як 504192. Це легкість є оманливим. У кожному півкулі нашого мозку люди мають зорову кору, , що містить 140 мільйонів нейронів, з десятками мільярдів зв'язків між ними. І все-таки людське бачення передбачає не тільки ряд візуальних картинок, а й аналізує більш складну модель зображень. Але майже вся ця робота виконується несвідомо. І тому ми не оцінюємо наскільки важкою є проблема, яку вирішують наші візуальні системи.

Ідея нейронних мереж полягає в тому, щоб взяти велику кількість певних об’єктів, як приклади для навчання, а потім розробити систему, яка може навчатися на цих прикладах. Іншими словами, нейронна мережа використовує приклади для автоматичного визначення правил розпізнавання. Крім того, збільшуючи кількість навчальних прикладів, мережа зможе дізнатись більше про певні характеристики і риси об’єкта. Таким чином, коли я показав лише 100 зображень з певними об’єктами ми зможемо навчити розпізнавати лише базові речі, використовуючи тисячі або навіть мільйони або мільярди навчальних прикладів нейронна мережа досягне високих результатів.

В основному нейронні мережі базуються на пенцептронах. Перцептрон – це структура, що приймає кілька подвійних входів і видає єдиний

бінарний вихід. Спосіб полягає в тому, що це пристрій, який приймає рішення, зважуючи на свідчення. Наприклад, припустимо, що ми маємо певну подію, і ми хочемо її відвідати. Ми можемо прийняти рішення, зваживши певні фкісь фактори. Ми можемо представляти ці фактори за допомогою відповідних двійкових змінних. Побудувавши певну модель ми можемо створювати маніпулювати вагами цих значень і приймати певні рішення. Так наприклад змінюючи ваги певних умов перцептрон може приймати різні рішення. Змінюючи ваги та порогові значення, ми можемо отримати різні моделі прийняття рішень. Часто нейронні мережі мають системи перцептронів, таким чином перший елемент буде тим, що ми називаємо першим шаром персептронів - робить певні прості рішення, зважуючи вхідні докази. Кожен знаступних персептронів приймає рішення, зважуючи результати попереднього рівня прийняття рішень. Таким чином, персептрон у другому шарі може приймати рішення на більш складному та більш абстрактному рівні, ніж персептрони в першому шарі. І навіть більш складні рішення можуть бути зроблені персептроном у третьому шарі. Таким чином, багаторівнева мережа персептрон може приймати складні рішення.

Кожен нейрон отримує вхідні сигнали від своїх дендритів і виробляє вихідні сигнали вздовж його (єдиного) аксона. Аксон з часом розгалужується і з'єднується через синапси з дендритами інших нейронів. У розрахунковій моделі нейрона сигнали, які рухаються уздовж аксонів (наприклад, x_0), мультиплікативно взаємодіють (наприклад, w_0x_0) з дендритами іншого нейрону, виходячи з синаптичної сили на цьому синапсі (наприклад, w_0). Ідея полягає в тому, що синаптичні сили (ваги w) можуть навчатися і контролюють силу впливу (і його напрям: збудливий (позитивний вага) або інгібіторний (негативний вага)) одного нейрона на інший. У базовій моделі дендрити несуть сигнал до тіла комірки, де всі вони отримують підсумки. Якщо остаточна сума перевищує певний поріг, нейрон може спрацювати, посилаючи імпульс уздовж його аксона. У розрахунковій

моделі ми припускаємо, що точні таймінги імпульсів не мають значення, і що тільки частота спрацьовування передає інформацію. Виходячи з цієї частотної інтерпретації коду, ми моделюємо частоту спрацювання нейрона *як активаційну функцію f*, яка відображає частоту імпульсів вздовж аксона. Історично загальним вибором функції активації є сигмоподібна функція *σ*, оскільки вона приймає справжнє значення входу (сила сигналу після суми) і розподіляє її на відстань між 0 і 1.

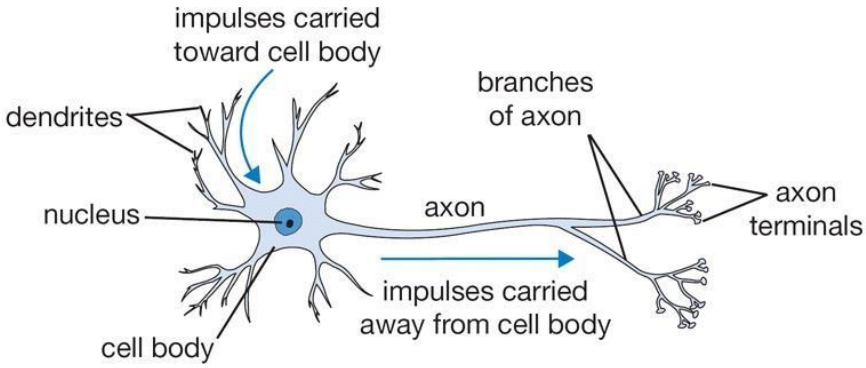

Рисунок 1.5 – Біологічна модель нейрона [7]

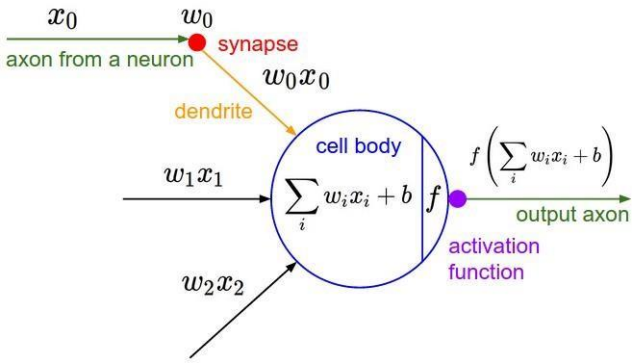

Рисунок 1.6 – Математична модель нейрона [7]

Пряме проходження нейрону можна описати так, як зображено на рисунку 1.7.

```
class Neuron(object):
  # ...
  def forward(self, inputs):
    """ assume inputs and weights are 1-D numpy arrays and
bias is a number """
    cell_body_sum = np.sum(inputs * self.weights) + self.bias
    firing_rate = 1.0 / (1.0 + math.exp(-cell_body_sum)) #
    sigmoid
activation function
    return firing_rate
```

Рисунок 1.7 – Код для прямого проходження нейрона

Іншими словами, кожен нейрон виконує скалярне множення його входу із його вагами, додає зміщення та застосовує нелінійність (або функцію активації), в цьому випадку сигмоїду.

1.8 Функції нейронних мереж

Кожна функція активації (або нелінійність) приймає одне число і виконує певну фіксовану математичну операцію на ньому. Існує кілька функцій активації, які ви можете зустріти на практиці:

Sigmoid (Сигмоїда). Сигмоїдна нелінійність представлена на графіку на рисунку 1.8:

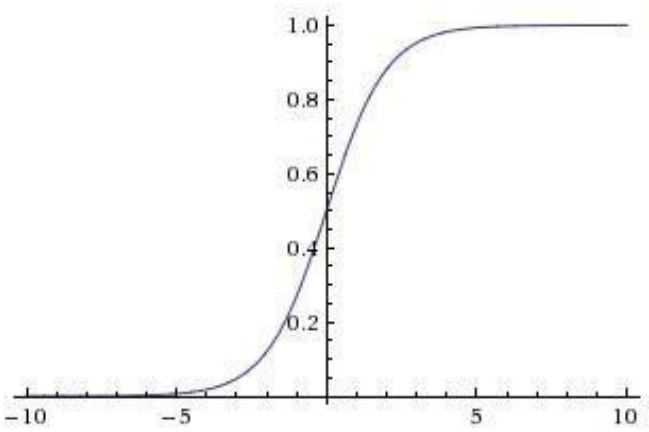

Рисунок 1.8 - Сигмоїдальна нелінійність, яка обмежує дійсні числа до діапазону [0,1]

Вона приймає дійсне число і "стискує" його в діапазон від 0 до 1.

Зокрема, великі негативні числа стають 0, а великі позитивні числа стають 1. Сигмоїдна функція часто зустрічається в історичному відношенні оскільки вона має приємну інтерпретацію як швидкість руху нейрона: від не спрацювання зовсім (0) до повністю насиченого спрацювання при прийнятній максимальної частоті (1). На практиці сигмоїдна нелінійність останнім часом випала з користі, і вона рідко використовується. Вона має два основних недоліки:

Сигмоїда насичується і вбиває градієнти. Дуже небажаним властивістю сигмоєвидного нейрону є те, що коли активація нейрона насичується в будь-якому хвості від 0 або 1, градієнт у цих областях майже дорівнює нулю. При зворотному розповсюдженні цей (локальний) градієнт буде помножений на градієнт виходу цього нейрону. Тому, якщо локальний градієнт дуже малий, він фактично «вб'є» градієнт, і практично не буде ніякого сигналу протікати через нейрон до його ваг та рекурсивно до його даних. Окрім того, потрібно приділяти особливу обережність при ініціалізації ваг сигмоєвидних нейронів, щоб запобігти насиченість. Наприклад, якщо початкові ваги занадто великі, більшість нейронів стануть насиченими, і мережа ледь навчиться. Виходи сигмоїди не нульові. Це небажано, оскільки нейрони у більш пізніх шарах обробки в нейронній мережі будуть отримувати дані, які не нульові. Це має наслідки для динаміки під час градієнтного спуску, тому що якщо дані, що надходять в нейрон, завжди позитивні (наприклад, $x> 0$ елементарно в $f = w^Tx + b$)), то градієнт на вагах w при зворотному поширенні стане будь-яким з усіх позитивний або усіх негативних (залежно від градієнта всього виразу f). Після того, як ці градієнти будуть додані до частини даних, остаточне оновлення для ваг може мати змінні ознаки, що трохи пом'якшує цю проблему. Тому це є незручно, але має менш серйозні наслідки в порівнянні з насиченою проблемою активації вище.

Tanh (Тангенсоїда, тангенційна нелінійність). Ця функція обмежує дійсне число до діапазону [-1, 1]. Подібно сигмоєвидному нейрону, його

активація насичується, але на відміну від сигмовидного нейрона його вихід є відцентрованим відносно нуля. Тому на практиці віддають перевагу тангенційній нелінійності, ніж сигмоподібній нелінійності. Тангенційний нейрон - це масштабований сигмоподібний нейрон, графік зображено на рисунку 1.9:

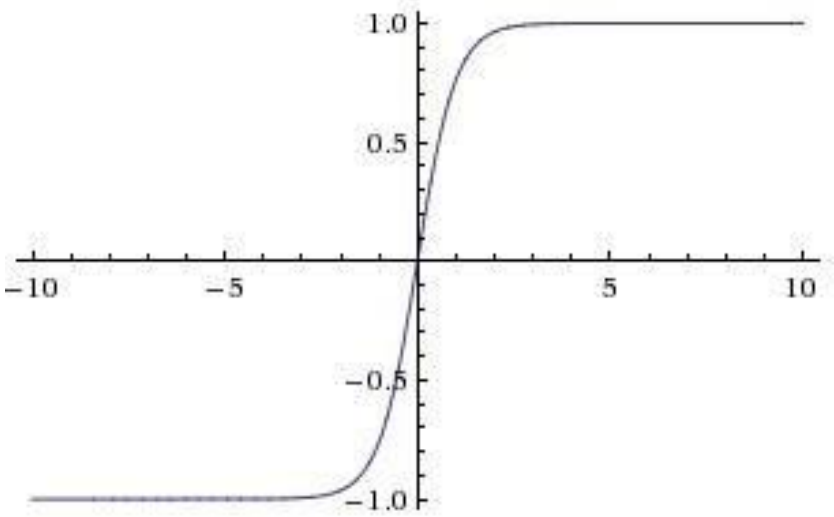

Рисунок 1.9 - Тангенційна нелінійність, яка обмежує дійсні числа до діапазону [-1,1] [7]

ReLU (Rectified Linear Unit). Активаційна функція ReLU стakf дуже популярним протягом останніх кількох років. Її математична формула виглядає так (формула 1.3), а її графік зображено на рисунку 1.10:

$$f(x) = max(0, x) \quad (1.3)$$

Іншими словами, активація – це просто перетин нуля. Є кілька плюсів і мінусів до використання ReLU:

(+) Знайдено значне прискорення збіжності стохастичного градієнтного походження порівняно з функціями Sigmoid / Tanh. Стверджується, що це пов'язано з його лінійною, не насичуючою формою.

(+) У порівнянні з Tanh / Sigmoid нейронами, що передбачають дорогі операції (експоненти тощо), *ReLU* може бути реалізований шляхом простого перетинання матриці активації нуля.

(-) На жаль, нейрони ReLU можуть бути нестабільними під час тренувань і можуть «вмирати». Наприклад, великий градієнт, що протікає через нейрон ReLU, може призвести до оновлення ваг таким чином, що нейрон більше ніколи не активується на будь-яких даних. Якщо це станеться, градієнт, що протікає через нейрон, назавжди стане нульовим з цієї точки. Тобто нейрони ReLU можуть незворотно вмирати під час тренувань. Наприклад, ви можете виявити, що до 40% вашої мережі може бути "мертвою" (тобто нейронами, які ніколи не активуються по всьому навчальному набору даних), якщо швидкість навчання встановлена занадто висока. При правильному налаштуванні швидкості навчання це рідше є проблемою.

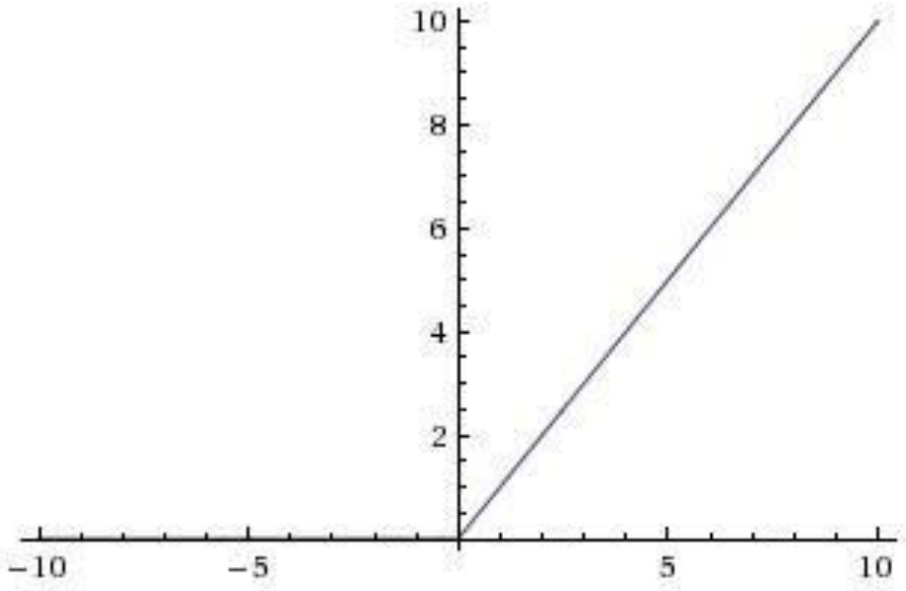

Рисунок 1.10 - Активаційна функція Left: Rectified Linear Unit (ReLU), яка приймає значення 0, коли x < 0 і є лінійно зростаючою, коли x > 0 [7]

1.9 Архітектура згорткових нейронних мереж

Згорткові нейронні мережі (СНС, CNN) дуже схожі на звичайні нейронні мережі: вони також побудовані на основі нейронів, які володіють постійно змінюваною вагою і зсувами. Кожен нейрон отримує деякі вхідні дані, виконує скалярний додуток інформації і в окремих ситуаціях супроводжує це нелінійностью. Як і у випадку зі звичайними нейронними мережами, вся CNN висловлює одну диференційовану функцію: з одного боку це необроблені пікселі зображення, з іншого - висновок класу або групи ймовірних класів, які характеризують картинку. Тут також присутня функція втрати на останньому (повністю підключеному) шарі, а всі елементи розроблені і модифіковані з простими нейронними мережами, залишаються справедливими при роботі з CNN.

Архітектура згорткових нейромереж робить явне припущення виду «вхідні дані є зображення», що дозволяє закодувати певні властивості під архітектуру. Завдяки цій особливості, попереднє оголошення можна реалізувати більш ефективно, зменшуючи при цьому кількість параметрів в мережі.

Як відомо, нейронні мережі отримують вхідні дані (один вектор), після чого трансформують інформацію, проводячи її через ряд прихованих шарів. Кожен прихований шар складається з безлічі нейронів, де всякий нейрон має стійкий зв'язок з усіма нейронами в попередньому шарі і де нейрони в функції одного шару повністю незалежні один від одного і не мають спільних з'єднань. Останній повнозв'язний шар називається вихідним шаром, і в класифікації він демонструє число класів.

Звичайні нейронні мережі погано масштабуються у випадку з зображеннями великих розмірів. Так, в системі комп'ютерного зору CIFAR-10, картинки становить [32 × 32 × 3] (32 - ширина, 32 - висота, 3 - колірні канали), тому один повністю підключений нейрон в першому прихованому шарі звичайної нейронної мережі має вагу 3 072 (32 * 32 * 3). Здається, що

це значення можна змінювати, але полносвязная структура не масштабується для великих зображень. Картинка великого розміру, наприклад, [200 × 200 × 3], призведе до того, що повністю підключений нейрон буде важити 120 000. Крім того, ми майже напевно хотіли б мати кілька таких нейронів, що призвело б до додавання параметрів. Повна зв'язність - це марнотратство, і величезна кількість параметрів може швидко привести до перенавчання.

Згорткові нейронні мережі користуються тим, що ввідні дані складаються з зображень, і вони обмежують побудову мережі більш розумним шляхом. На відміну від звичайної нейронної мережі, шари CNN складаються з нейронів, розташованих в 3-х вимірах: ширині, висоті і глибині. Тобто вимірах, які формують обсяг. Наприклад, зображення на вході CIFAR-10 є вхідними активаційними обсягами, а обсяг сформований вимірами 32 × 32 × 3. Нейрони будуть підключені тільки до невеликої області шару перед цією ділянкою. Крім того, результуючий вихідний шар для даної системи комп'ютерного зору складе 1x1x10, оскільки до кінця побудови CNN ми перетворимо повне зображення в єдиний вектор оцінок класу, розташованих по вимірюванню глибини.

Розташування нейронів у звичайній тришарової нейромережі. Нейрони свёрточной нейронної мережі розташовуються в 3-х вимірах, як це показано на одному з шарів. Кожен шар CNN перетворює вхідний 3D-об'єм у вихідний активаційний 3D-об'єм нейронів. В даному прикладі червоний вхідний шар містить зображення, тому його ширина і висота визначається розмірами картинки, а глибина буде дорівнює 3 (червоний, зелений, синій канали) (рис 1.11).

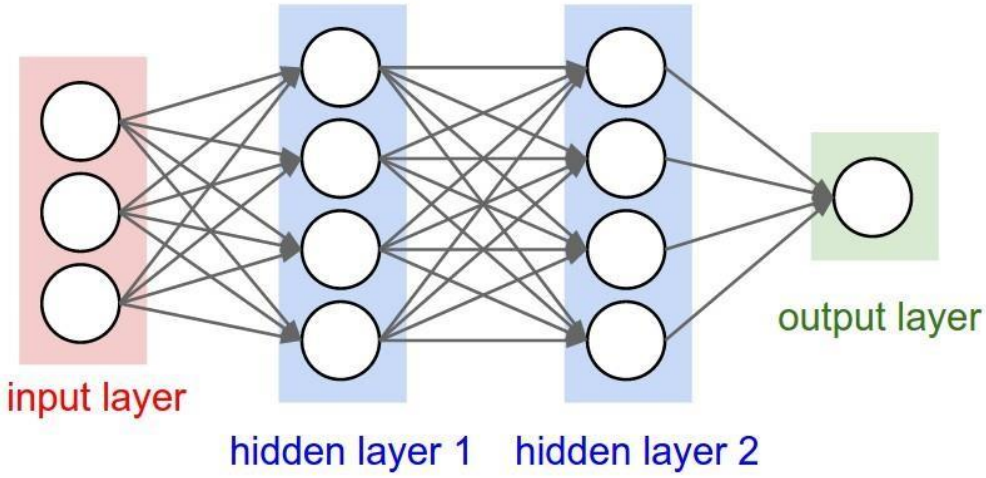

Рисунок 1.11 - Звичайна тришарова нейронна мережа

Основу згорткових нейромереж складають шари. Кожен шар характеризується простим API: він перетворює вхідні дані у вигляді 3D-об'єму у вихідний 3D-об'єм з деякою дифференцируемой функцією, яка може мати або не мати параметри (рис 1.12).

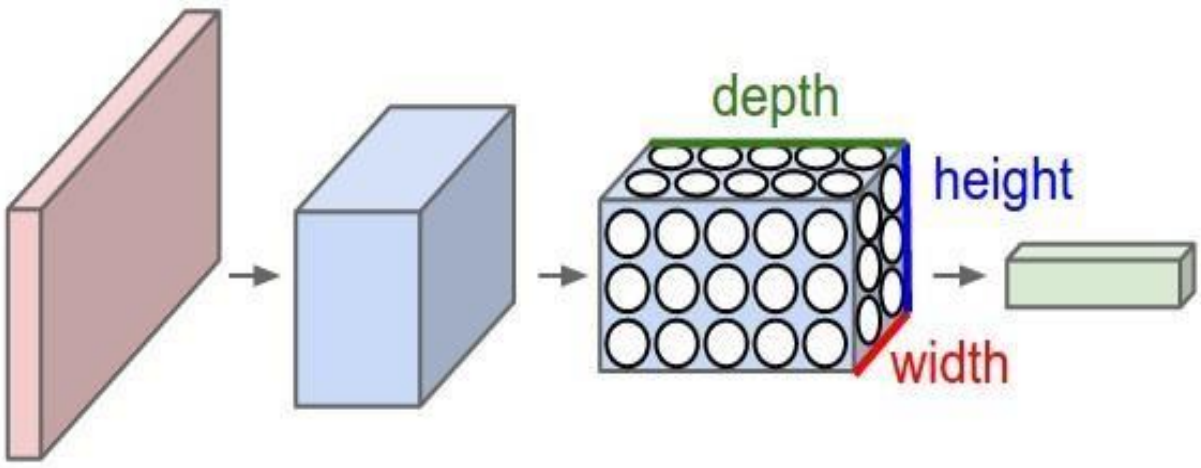

Рисунок 1.12 – Згорткова нейронна мережа

Для організації згорткової нейронної мережі застосовується 3 основних шару. Ці шари використовуються з метою побудови повної архітектури CNN:

- Згортки.
- Пулінгу (інакше підвибірки або Субдіскретізація).
- Повнозв'язний шар.

Першим кроком є «Input» (вхідні дані), що містить інформацію про зображення, наприклад зображення 128 × 128 × 3, де 128 - ширина, 128 - висота, 3 - канали кольору Red, Green, Blue.

Наступним, одним з найважливіших кроків нейронної мережі, є «Convolution» (шар згортки), де відбувається примноження значень фільтра на вихідні значення пікселів зображення (поелементне множення), після чого всі ці множення підсумовуються і записуються у нову структуру даних. Натренерована згорткова нейронна мережа має велику кількість фільтрів завдяки яким і відбувається пошук певних характеристик. Кожна унікальна позиція введеного зображення виробляє число, що записується у структуру даних. Якщо використовується 5 фільтрів, обсяг отриманих даних буде дорівнювати 128 * 128 * 5.

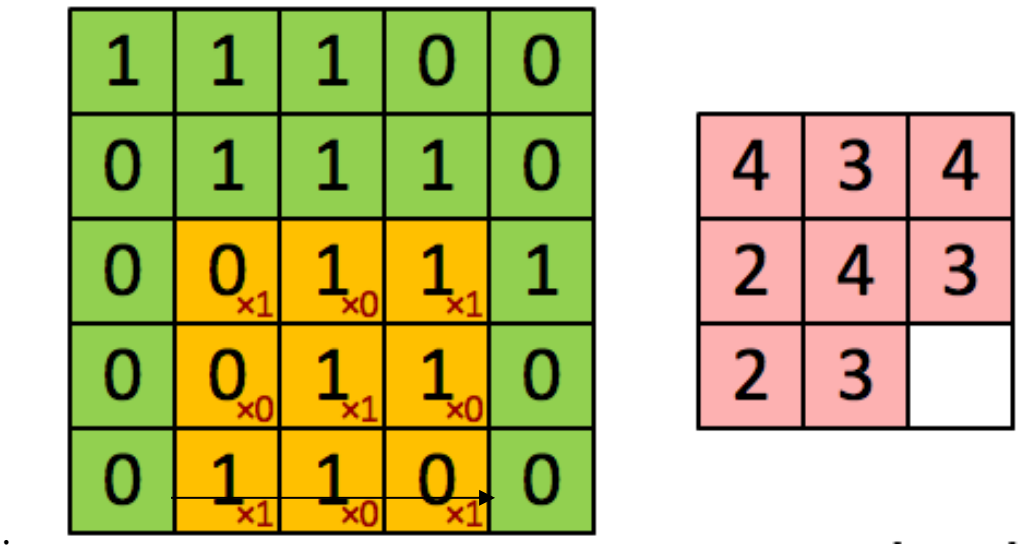

Рис 1.13 - Накладання фільтру на один з каналів зображення і отримання нове значення

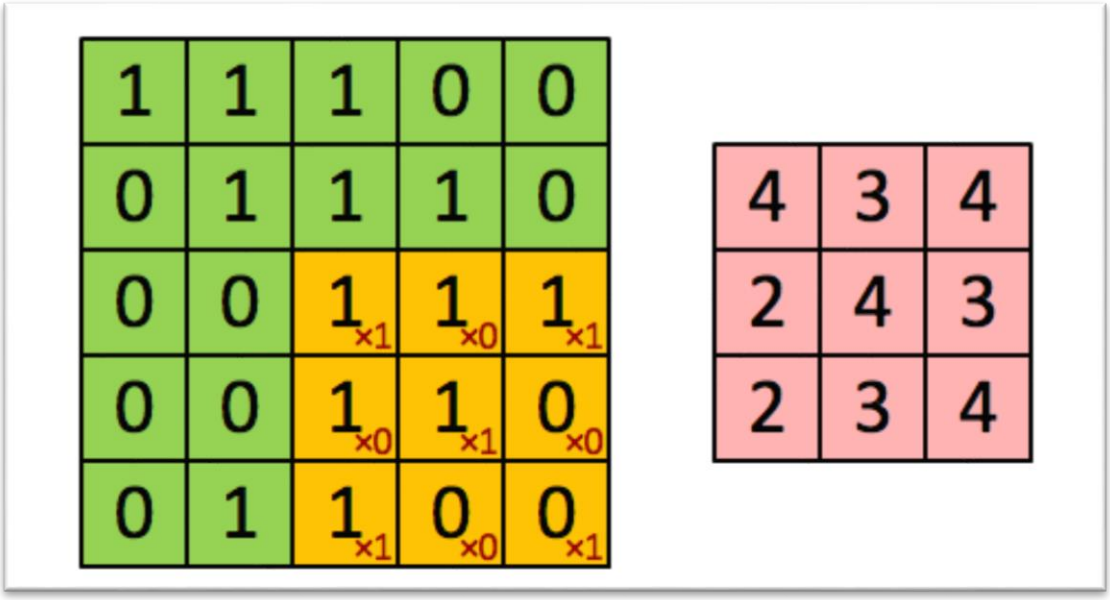

Рис 1.14 - Результат після накладання фільтру

Після згортки йде крок «Pool», що виконує операцію для зменшення розмірів зображення (широти і висоти), в результаті чого обсяг може скоротитися до [64 × 64 × 5]. Тобто на цьому етапі виконується нелінійне ущільнення карти ознак. Оскільки на попередній операції згортки вже були виявлені деякі ознаки, то для подальшої обробки настільки докладне зображення вже не потрібно. Для цього із частини пікселів ми обираємо тільки ті, що мають найбільші значення і ущільнюємо до менш докладної картинки.

Крок «Rectified linear units» (блок нормалізації) застосовує функцію f (x) = max (0, x) для кожного елементу даних, встановлюючи певний нульовий поріг. Іншими словами, якщо x > 0, то обсяг даних залишається колишнім (64 × 64 × 5), а якщо x < 0, то відкидаються непотрібні деталі в каналі шляхом заміни на 0. Простішими словами ми анулюємо ознаки які не грають важливаї ролі у розпізнаванні образу.

Крок «Fully connected layer» виводить N-вимірний вектор (N - число класів) для визначення потрібного класу з тих даних над якими ми провели всі потрібні операції. Таким чином беруться дані із попереднього кроку формуючи вектор властивостей після чого він зрівнюється з тими типами

об'єктів які нейронна мережа навчилася розпізнавати. Таким чином є можливість розпізнавати на зображенні певні об'єкти, їх позицію та маніпулювати ними [2].

Саме така структура дає змогу нейронній мережі розпізнавати різні об'єкти у різноманітних позіціях та показувати досить високий результат. Завдяки цьому є можливість отримати дані про положення основних частин тіла та число, що відображає оцінку, тобто на скільки нейронна мережа впевнена, що ця точка підходить під наш каскад. Із отриманих частини тіла (голова, руки, ноги тощо) формується певний каскад для розпізнавання.

У найпростішому випадку архітектура CNN - це набір шарів, які перетворять образ зображення в вихідний образ. Кожен шар відповідає за певний етап процесу обробки зображення. Кожен шар отримує на вході об'ємну 3D інформацію і трансформує зі збереженням 3D-об'єму за допомогою функції, що диференціюється. Шар як може мати додаткові параметри, так ці параметри можуть бути відсутні. Кожен активаційний об'єм в ході обробки зображення показаний у вигляді стовпчика. Коли візуалізувати 3D-об'єм стає важко, проводиться викладка обсягу кожного шару в ряд. Обсяг останнього шару містить оцінку вірогідності для кожного можливого класу, причому FC візуалізує класи в відсортованому порядку.

Шар згортки - це основоположний шар CNN, який виконує більшість важкої роботи. Припустимо, що шар згортки працює без «етапу логіки» або нейронного підгрунтя. Параметри шару згортки складаються з набору фільтрів. Кожен фільтр має малі просторові габарити (ширину і висоту), але проходить по всій глибині обсягу. Наприклад, стандартний фільтр першого шару згорткової нейронної мережі може бути розміру [5x5x3]. Під час проходу кожен фільтр йде по ширині і висоті вхідних даних і обчислює скалярний добуток між записами фільтра і входом в будь-яке положення. У міру проходження фільтра по ширині і висоті зображення, ми складаємо 2-мірну активаційну карту. Мережа навчає певні фільтри, які активуються при виявленні певної візуальної особливості. Це може бути певна грань,

плямистість конкретного кольору на першому шарі або кільцеподібні візерунки. Тепер ми будемо працювати з цілим набором фільтрів в кожному шарі згортки, і кожен з них буде формувати окрему 2-мірну активаційну карту. Ми будемо складати ці активаційні карти вздовж вимірювання «глибина» і формувати вихідний обсяг.

Кожен запис в вихідному обсязі можна інтерпретувати як вихідний нейрон, який дивиться тільки на невелику ділянку вхідного обсягу і просторово ділить параметри з усіма нейронами зліва і справа (оскільки вони є результатом застосування такого ж фільтра).

Коли мова йде про роботу з вхідною інформацією з високою розмірністю, установка зв'язку між нейронами і всіма нейронами з колишнім обсягом є недоцільною. Замість цього ми будемо підключати кожен нейрон тільки до локальної області вхідного обсягу. Просторова протяжність зв'язку з цим є гіперпараметром і називається рецептивним полем.

Припустимо, що картинка на вході має розмір [32 × 32 × 3] (наприклад, RGB-зображення CIFAR-10). Якщо розмір фільтра дорівнює 5 × 5, тоді кожен нейрон в шарі згортки матиме вагу в межах [5x5x3] вхідного обсягу, що в підсумку дасть 5 * 5 * 3 = 75 (+1 параметр зсуву). Зауважте, що просторова протяжність уздовж осі глибини повинна бути дорівнює 3: тоді є гарантія математичної вірності. Ще, як приклад, нехай тепер вхідне зображення має розмір [16 × 16 × 20]. Тоді, якщо ми використовуємо фільтр 3 × 3, кожен нейрон в шарі згортки матиме в сумі 180 (3 * 3 * 20) з'єднань з об'ємом на вході. Знову звертаємо вашу увагу на те, що зв'язність є локальною по ширині і висоті (тут - 3 × 3), але проходить через всю глибину введення (20).

В архітектурі CNN звичайною практикою є вставка шару «Pooling» (підвибірки) між послідовностями згорткових шарів. Його функція полягає в поступовому зменшенні просторові габарити зображення з метою зменшення кількості параметрів і обчислень в мережі, а також контролю

перенавчання. Шар «Pooling» працює незалежно від зрізу глибини вхідних даних і масштабує обсяг просторово, використовуючи функцію максимуму. Найчастіше використовується шар з фільтрами розміру 2 × 2 з кроком 2; подібний шар знижує дискретизацию кожного зрізу глибини входу в 2 рази як по ширині, так і по висоті, відкидаючи при цьому 75% даних. Кожна операція максимума в цьому випадку буде вибирати максимальне значення з 4 чисел. Розмір по глибині при цьому залишається незмінним (рисунок 1.15).

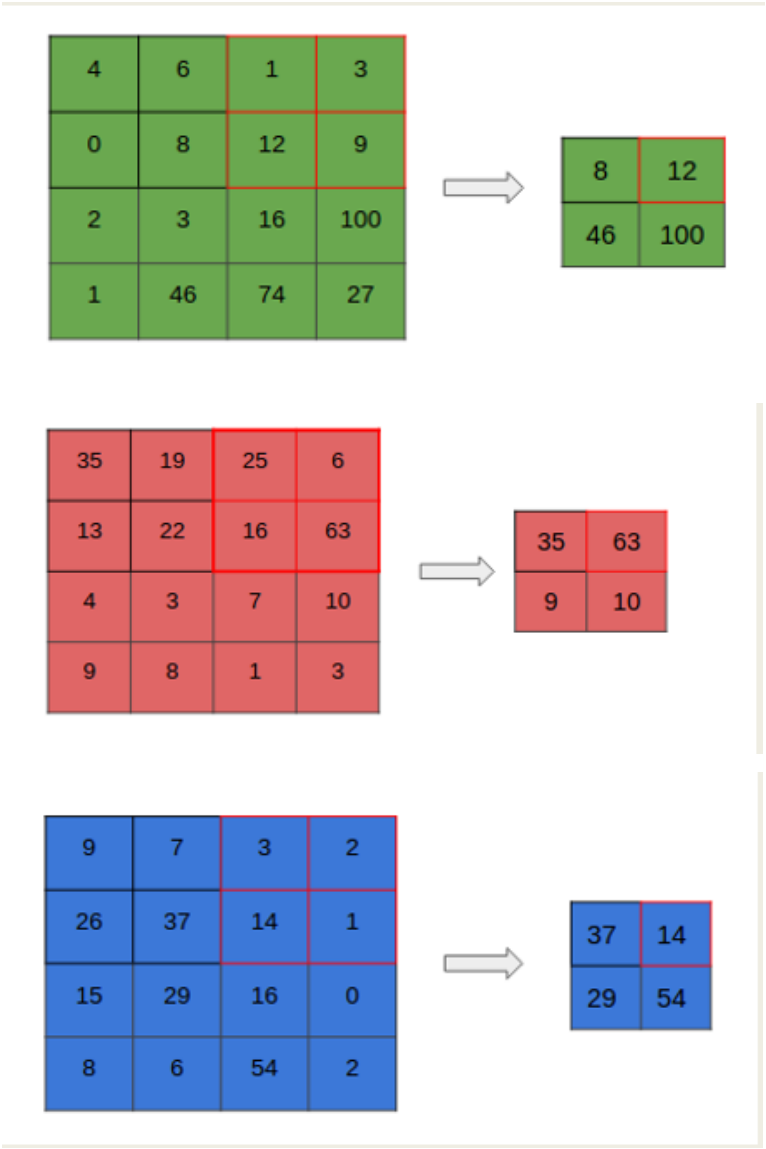

Рис 1.15 - Результат шару «Pooling» для трьох шарів

1.10 Висновки за розділом

Таким чином можна зробити висновок, що існує багато методів та способів роботи із зображенням. Обробка та розпізнавання зображень важкий і кропіткий процес, але перспектива таких технологій дуже висока. Інструменти обробки зображень мають в собі реалізації складних алгоритмів, що забезпечують швидку обробку пікселів зображень і виведення результатів. За допомогою різних іструментів є можливість реалізовувати методи розпізнавання образів рна зображеннях. Одним з таких популярних інструментів є OpenCV.Також було розглянуто популярніть та структуру згорткових нейронних мереж і їх можливості, а щк кожен крок обробки вхідних даних.

Тому можна сказати, що комп’ютерний зір як напрямок комп’ютерних технологій розвивається дуже швидко і відкриває все нові можливості.

2. АНАЛІЗ МЕТОДУ РОЗПІЗНАВАННЯ ОБРАЗІВ

2.1. Аналіз бібліотеки OpenCV

OpenCV (Open Source Computer Vision Library) випущений під ліцензією BSD і, отже, безкоштова як для академічного, так і для комерційного використання. Вона має інтерфейси для мов C ++, Python і Java і підтримує Windows, Linux, Mac OS, iOS та Android. Бібліотека OpenCV була розроблений для задач розпізнавання. Мова на якій вона написана це C/C++, сама ж бібліотека може скористатися багатоядерною обробкою. Бібліотека використовує OpenCL, вона може скористатися апаратним прискоренням базової неоднорідної обчислювальної платформи.

Прийнята по всьому світу, OpenCV налічує понад 47 тисяч користувачів спільноти та приблизну кількість завантажень понад 14 мільйонів. Використання цієї бібліотеки дуже широке від інтерактивного мистецтва до огляду фільмів та воєного обладнання.

OpenCV – це бібліотека з набором методів, алгоритмів та технологій для роботи із розпізнаванням зображень. Бібліотека має в собі готові методи для використання користувачем, тому написати просту програму для розпізнавання не завдає великих труднощів. Сама в собі бібліотека має готові каскади, що дають змогу змогу знаходити певні нам об’єкти. Так стандартна бібліотека дає нам можливість знаходити майже всі елементи лиця, певні частини тіла та все тіло взагалом. Таких можливостей часто достатньо для простих задач.

2.2. Переваги та недоліки бібліотеки OpenCV

Ціллю тестування OpenCV став аналіз результатів успішності бібліотеки з об’єктами пошуку на різних зображенях різної якості. Об’єктом для пошуку булао вибрано повноцінне людське тіло. Такий аналіз показав би успішність та придатність системи до скадних ситуацій. Таким чином уа

написана програма, що дає можливість знаходити тіо юдини на зораженні. Ми використаи мову програмування Python, тому що вона дає можливість швидко і якісно написати певне ПО у короткий час та з невеликим обсягом. Також для розпізнавання ми використали стандартний каскад для пошуку повного тіла людини, який ми можемо знайти у засобах OpenCV. Таким чином наша програма має представлена на наступних рисунках 2.2.1 і 2.2.2.

```
if __name__ == '__main__':
    import sys
    from glob import glob
    import itertools as it

    print(__doc__)

    hog = cv.HOGDescriptor()
    hog.setSVMDetector( cv.HOGDescriptor_getDefaultPeopleDetector() )

    default = ['../test.mp4 '] if len(sys.argv[1:]) == 0 else []

    for fn in it.chain(*map(glob, default + sys.argv[1:])):
        print(fn, ' - ',)
        try:
            img = cv.imread(fn)
            if img is None:
                print('Failed to load image file:', fn)
                continue
        except:
            print('loading error')
            continue

        found, w = hog.detectMultiScale(img, winStride=(8,8), padding=(32,32), scale=1.05)
        found_filtered = []
        for ri, r in enumerate(found):
            for qi, q in enumerate(found):
                if ri != qi and inside(r, q):
                    break
            else:
                found_filtered.append(r)
        draw_detections(img, found)
        draw_detections(img, found_filtered, 3)
        print('%d (%d) found' % (len(found_filtered), len(found)))
        cv.imshow('img', img)
        ch = cv.waitKey()
        if ch == 27:
            break
    cv.destroyAllWindows()
```

Рисунок 2.1 - Лістинг методів для пошуку та фільтрації об’єктів

Як ми можемо побачити тут відбувається спочатку отримання каскаду даних для форми людини, після чого йде обробка зображення таким чином,

щоб була можливість порівняти певні частини зображення. Далі відбувається пошук, якщо на певнній частині зображення був знайдений об’єкт схожий на тіло людини він додається до списка даних. Наприкінці ми отримуємо певну кількість знайдених об’єктів і відображаємо їх на вхідному зображенні. Для тестування була відібрана певна база зображень, для того аби для різних методів розпізнавання ми мали об’єктивний результат аналізу. Тому були відібрані зображення із різною якістю, яркістю та різними об’єктами на самому зображенні.

```
from __future__ import print_function

import numpy as np
import cv2 as cv


def inside(r, q):
    rx, ry, rw, rh = r
    qx, qy, qw, qh = q
    return rx > qx and ry > qy and rx + rw < qx + qw and ry + rh < qy + qh


def draw_detections(img, rects, thickness = 1):
    for x, y, w, h in rects:
        pad_w, pad_h = int(0.15*w), int(0.05*h)
        cv.rectangle(img, (x+pad_w, y+pad_h), (x+w-pad_w, y+h-pad_h), (0, 255, 0), thickne
```

Рисунок 2.2 - Лістинг методів для відображення об’єктів

Тестування показало, що на простих та чітких зображеннях розпізнавання давалось досить легко. Так, наприклад, на результатах можна побачити, що більшість людських тіл, які добре видно наша програма розпізнає на рис. 2.3 і рис 2.4.

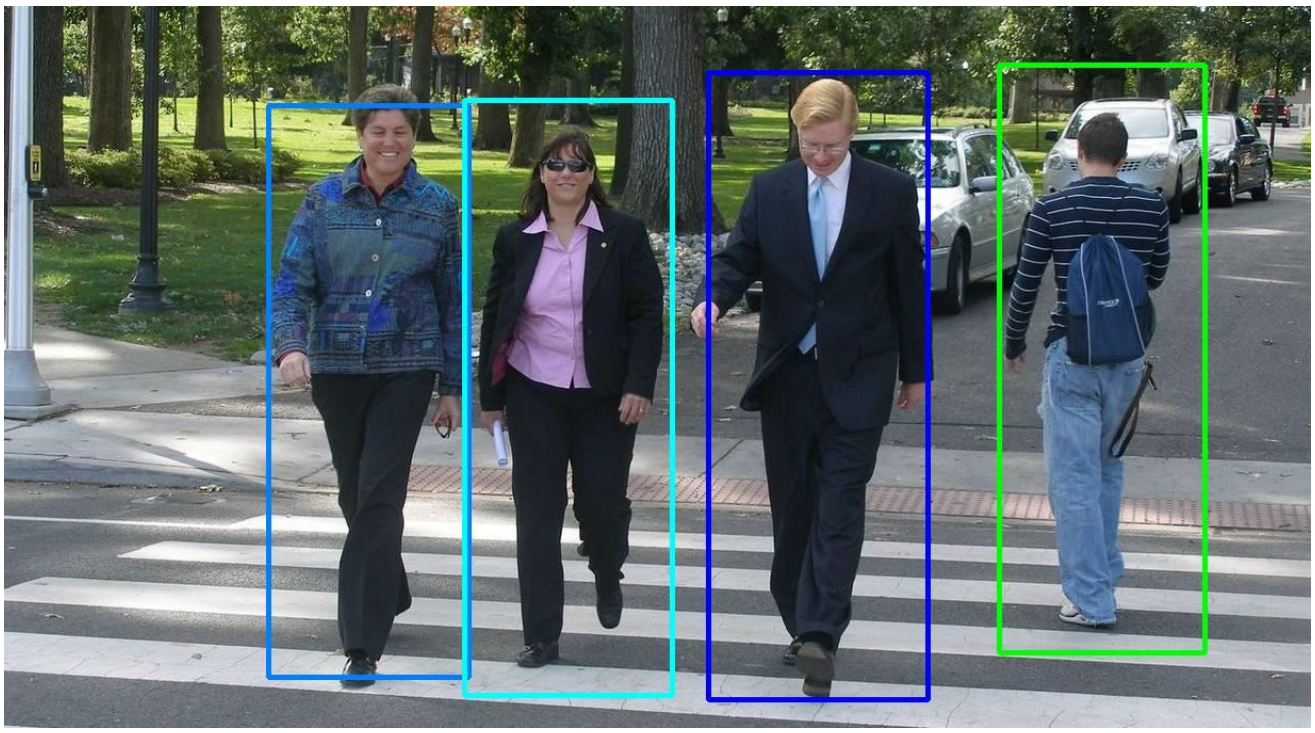

Рисунок. 2.3 - Пошук тіл людей за допомогою OpenCV

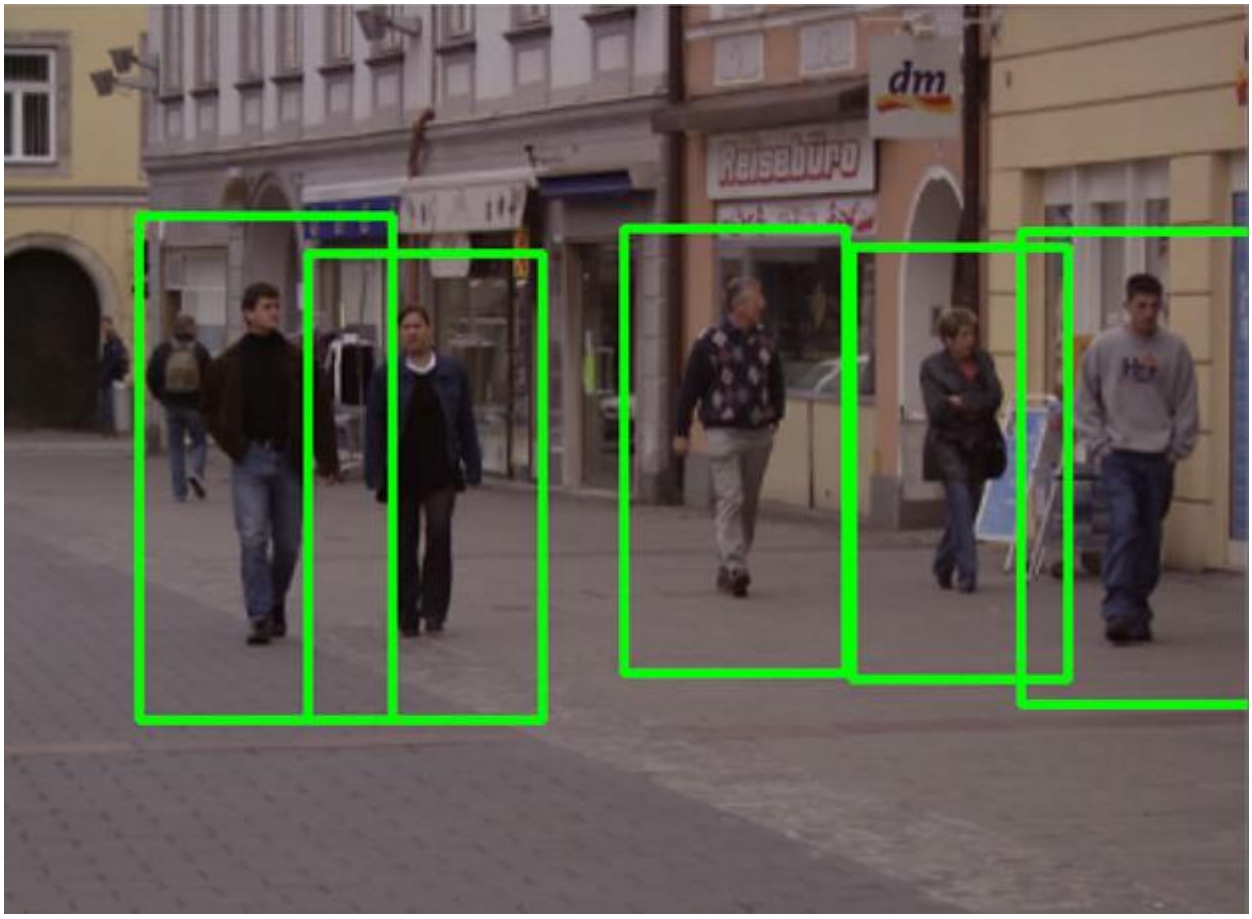

Рисунок 2.4 - Пошук великої кількості людей за допомогою OpenCV

Але наша система може знаходити тідки образи повного тіла, а не його частин. Також OpenCV погано справляється із великою кількістью людей та з образами, де об’єкти знаходяться в незвичних формах. Тобто якщо людина підіймає руки або приймає незвичне положення (наприклад танец або спорт) тоді спроба розпізнавання не дає жодних результатів. А тому пошук тіл людини не матиме хорошого результату і такий метод не є підходящим.

2.3. Метод розпізнавання за допомогою згорткової нейронної мережі

Як було розглянуто в першому розділі, згорткові нейронні мережі набувають широкої популярності у розпізнаванні об’єктів. Це завдяки своїм можливостям у навчанні та способі обробки зображень, оскільки маючи хороший набір даних для навчання нейронні мережі можуть навчитися розпізнавати будь-які об’єкті різної складності. Це дає змогу маніпулювати інструментами розпізнавання для різних задач. Для роботи із зображеннями аикористовують згорткові нейронні мережі. Завдяки своєму способу обробки зображень, вони легко знаходят образи на зображеннях, що навчились розпізнавати. Їх основною перевагою є гнучкість завдяки якій розпізнавання складних образів навіть у незвичних формах дає високі результати. Важливою частиною для згорткових нейронних мереж є саме велика сукупність зображень, що мають в собі образ для пошуку. Такі зображення називаються датасетом (DataSet – множина даних). Датасети бувають різними: датасети для навчання, датасети для тестування тощо. Основним для нейронних мереж є датасет для навчання, оскільки він має придатні для розпізнавання зображення, на яких точно присутні потрібні на об’єкти пошуку. Такі датасети налічують від 100 000 зображень і можуть досягати мільйонів, де потрібні образи знаходяться у різних позиціях. Датасети для тестування створені для перевірки нейронної мережі на придатність та ефективність. Тобто за допомогою таких датасетів ми

тестуємо на скільки добре наша нейронна мережа навчилася розпізнавати зображення. Таким чином після тренування ми можемо проаналізувати ефективність розпізнавання згорткової нейронної мережі.

Саме такий метод розпізнавання за допомогою згорткової нейронної мережі є найбільш підходящим для розпізнавання частин тіла людини. За допомогою такого відкритого інтернет ресурс як COCO dataset, що має велику кількість датасетів із зображеннями для тренування нейронних мереж, ми можемо отримати великий датасет для частин людського тіла.

2.4. Переваги та недоліки різних архітектур згорткових нейронних мереж

Як було вище розглянуто, найефективнішими метотодами розпізнавання частин людського тіла є саме згорткові нейронні мережі. Але й сама структура нейронних мереж буває досить різною. Від архітектури залежить швидкість та ефективність розпізнавання образів. Існує багато видів вже створених архітектур для нейронних мереж з різною послідовністю дій. Далі буде розглянуто вже створені архітектурні рішення згорткових нейронних мереж.

LeNet-5, новаторська 7-рівнева згорткова мережа (рис 2.5). В 1998 році, яка класифікує цифри, була застосована кількома банками для розпізнавання рукописних цифр на чеках (чеках), оцифрованих в 32 x 32 пікселях у градаціях сірого. Можливість обробки зображень з високою роздільною здатністю вимагає більших і більше згорткових шарів, тому ця техніка обмежується наявністю обчислювальних ресурсів.

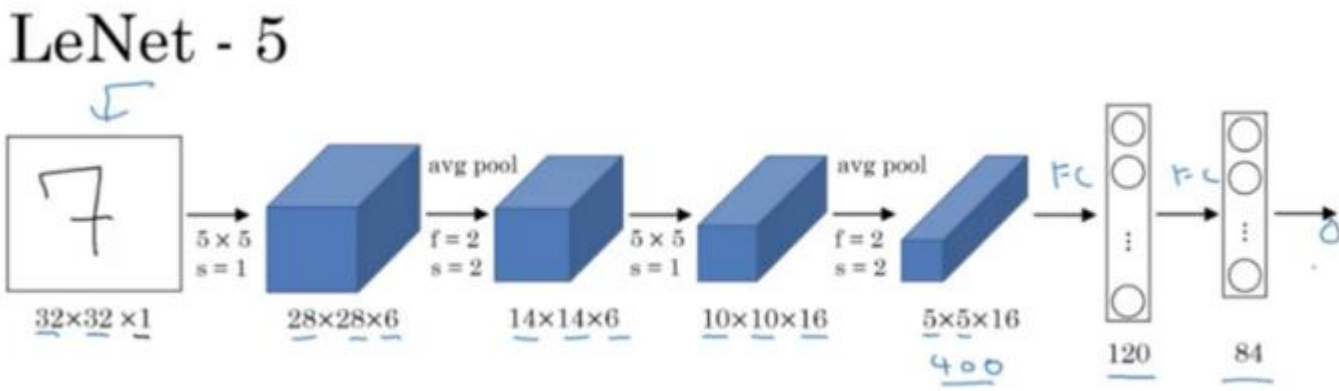

Рисунок 2.5 - Архітектура LeNet-5

Так наприклад, LeNet-5 отримує вхідний образ 32 x 32 x 1 (зображення сірого кольору) і метою є визначити схеми рукописних цифр. Система використовує 5 x 5 фільтри. Застосовуючи формулу розрахунку вищезгаданого поля, і результат результату виводу становить 28 × 28.

W x H → 32 x 32 (ширина на висоту)

F (w x h) → 5 x 5 (фільтр)

S → 1 (Stride)

P → 0 (об'єднання)

$$\left(\frac{W - Fw + 2P}{Sw}\right) + 1 => \left(\frac{32 - 5 + 0}{1}\right) + 1 => 27 + 1 => 28$$

$$\left(\frac{H - Fh + 2P}{Sh}\right) + 1 => \left(\frac{32 - 5 + 0}{1}\right) + 1 => 27 + 1 => 28$$

Таким чином результат отриманих даних дорівнює матриці 28 x 28. Наступний шар - це шар зменшення даних (Poll) для того аби працювати із зобраенням меншого розміру.

IM → 28 (Матриця вводу → згортання даних із попереднього кроку).

P → 0 (об'єднання)

S → 1 (Stride)

$$\frac{IM + 2P - 2}{S} + 1 => \frac{28 + 2*0 - 2}{2} + 1 => \frac{28 - 2}{2} + 1 => 14$$

На цьому етапі матриця даних має розмірність 14 x 14. Нарешті, відбувається зрівняння отриманих даних із класами для розпізнавання – fully connected layer (FC Layer).

У 2012 році була створена нова архітектура згорткової нейронної мережі – AlexNet (рис 2.3.2), що значно перевершила всіх попередніх конкурентів, зменшивши помилки з 26% до 15,3%. До цього кількість помилок при розпізнаванні складало 26,2%. Ця архітектура була однією з перших глибинних мереж, які значно підштовхували точність класифікації ImageNet у порівнянні з традиційними методологіями. Архітектура складається з 5 згорткових шарів, за якими слідують 3 повністю об'єднаних шари класифікації, як показано на рисунку 2.6.

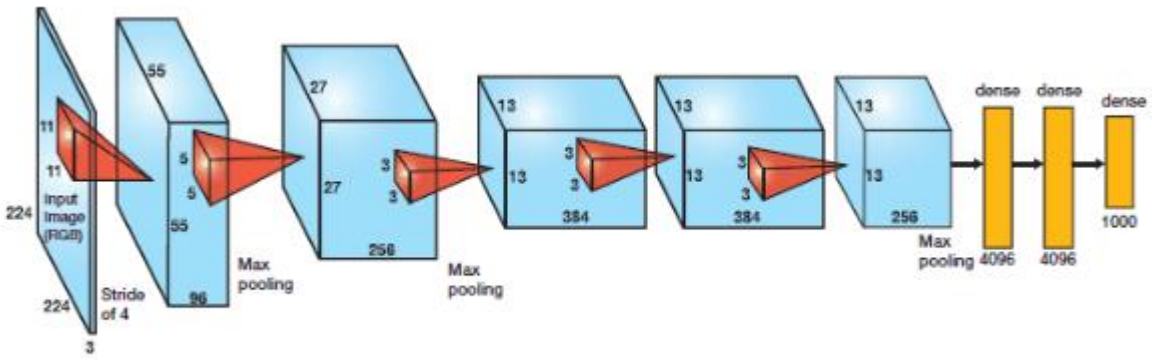

Рисунок 2.6 - Послідовність кроків AlexNet

AlexNet, запропонований Алексом Крижевським, використовує ReLu (лінійний блок виправлення) для нелінійної частини. ReLu виконує наступну функцію для всіх елементів f (x) = max (0, x). Перевага ReLu полягає в тому, що він тренується набагато швидше, ніж попередні типи шарів. У мережі ReLu поміщається після кожного кроку згортання.

Ще однією проблемою, яку вирішила ця архітектура, було зменшення надмірності, використовуючи шар Dropout після кожного FC-рівня. Його суть полягає в тому, що він навмисно виключає певні нейрони для того аби спонукати систему шукати нові шляхи для обробки даних. Таким чином нейронна мережа стає більш гнучка та ефективніша.

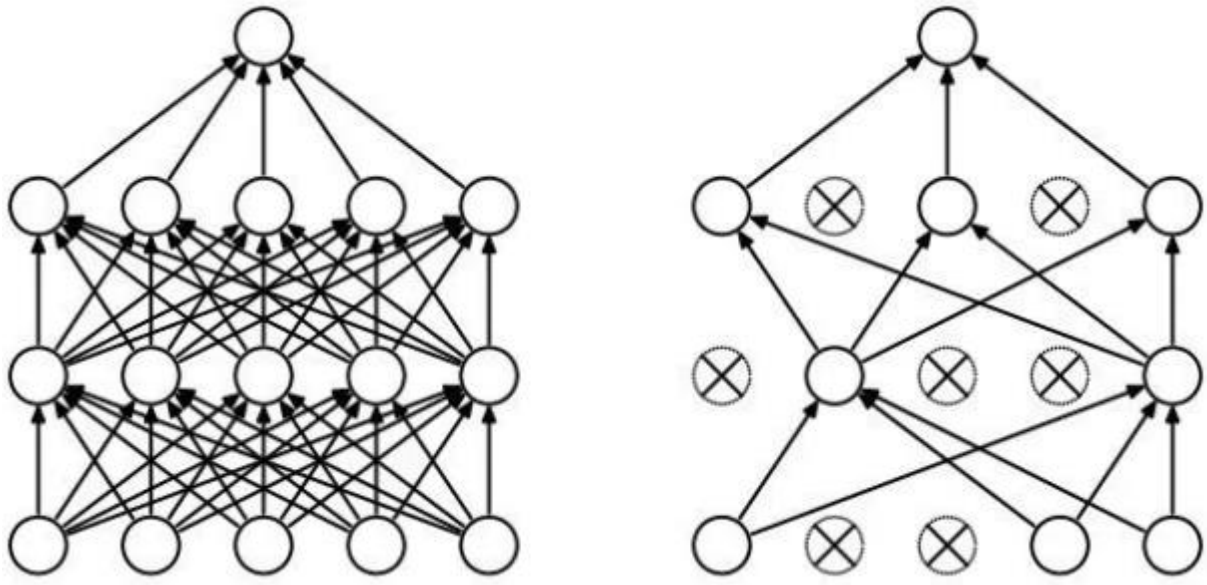

Рисунок 2.7 - Взаємодія нейронів з та без використання кроку DropOut.

Ідея відсіву полягає в наступному. Через відсівний шар, різні набори нейронів, які вимикаються, являють собою іншу архітектуру, і всі ці різні архітектури навчаються паралельно зі значенням. Для n нейронів, прикріплених до DropOut, кількість архітектур підмножини становить 2 ^ n. Таким чином, це означає, що прогнозування оптимізується. Це забезпечує регуляризацію структурованої моделі, яка допомагає уникнути перенавчання на одних і тих самих характеристиках. Інша думка про користь DropOut полягає в тому, що оскільки нейрони вибираються випадковим чином, вони, як правило, уникають розробки схожості між собою, тим самим дозволяючи їм розвивати важливі риси, незалежні від інших.

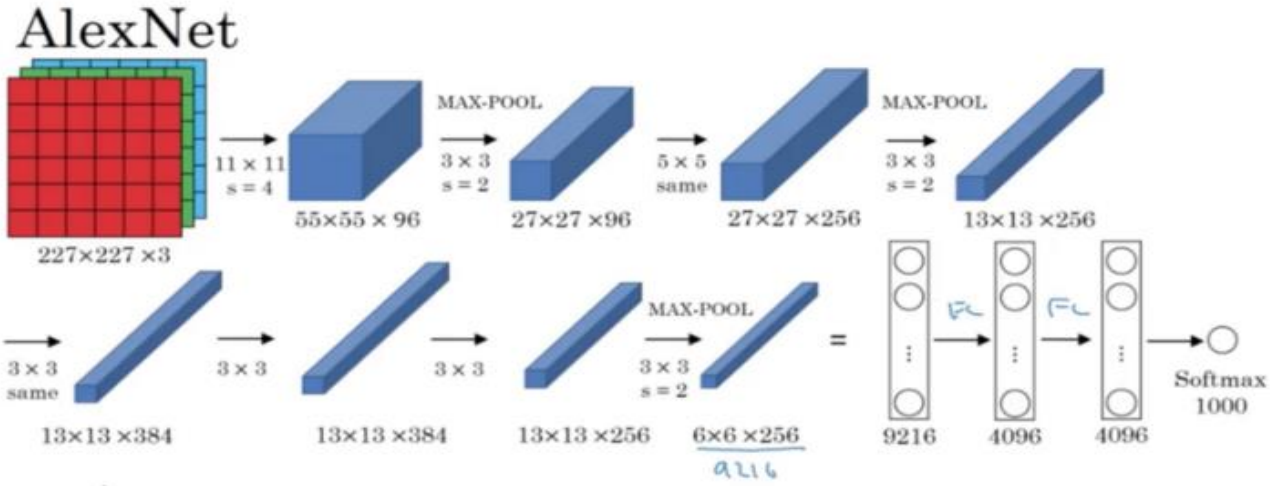

Рисунок 2.8 - Архітектура AlexNet

Також існує архітектура мережі, що була дуже схожа на LeNet. Але вона була більш глибокою, з більшою кількістю фільтрів на шар і зі стековими згортовими шарами. Вона складалася з 11x11, 5x3, 3x3 шарів і яка мала вже повний набір кроків згортки зображення. Такий метод має етап активації ReLU після кожного кроку згортання. Тести показували, що AlexNet навчався 6 днів одночасно на двох Nvidia GeForce GTX 580 GPU, що є причиною того, чому їх мережа розподілена. AlexNet був розроблений групою SuperVision.

Однією з добре ефективних систем була ZFNet (рис 2.4.4). Вона досягла верхньої помилки 14,8%. Це було в основному досягненням шляхом налаштування гіперпараметрів AlexNet при збереженні тієї ж структури з додатковими елементами глибокого навчання, як обговорювалося раніше.

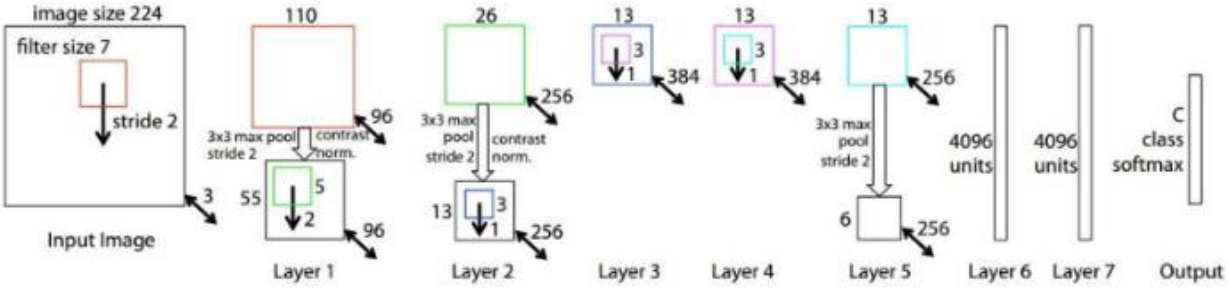

Рисунок 2.9 - Архітектура ZFNet

Однією з найпопулярніших архітектур згорткових мереж є VGGNet (рис 2.9). VGGNet складається з 16 згорткових шарів і дуже привабливою рівномірною архітектурою, що легко реалізувати на простих обчислювальних пристроях.. В даний час це найкращий вибір у спільноті для вилучення функцій із зображень. Вага конфігурації VGGNet є загальнодоступною і використовувалася в багатьох інших додатках. Однак VGGNet складається з 138 мільйонів параметрів, які можуть бути трохи складними для обробки.

Основною особливістю є те, що така структура використовує велику кількість фільтрів розміром 3 x 3, що дає змогу зменшити кількість параметрів для обробки зоброжень, а отже зменшити час та збільшити ефективність розпізнавання.

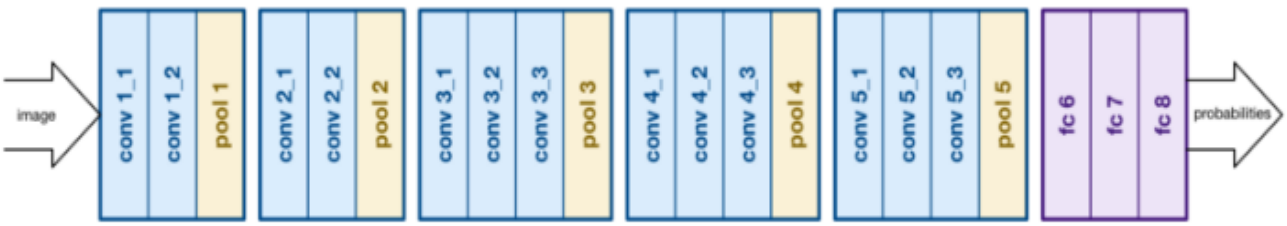

Рисунок 2.10 - Етапи обробки VGGNet

Хоча VGG досягає феноменальної точності на наборі даних ImageNet,

його розгортання на самих простих розмірах графічних процесорах є проблемою через величезні обчислювальні вимоги, як з точки зору пам'яті, так і часу. Це стає неефективним завдяки великій кількості згорткових шарів.

Наприклад, згортковий шар із розміром ядра 3 x 3, який приймає 512 каналів у якості вхідних і вихідних 512 каналів, порядок обчислень буде 9 x 512 x 512, що є досить великим значенням.

У згортковій операції в одному місці кожний вихідний канал, підключений до кожного вхідного каналу, і тому ми називаємо це щільною зв’язною архітектурою. GoogLeNet спирається на те, що більшість активацій у глибокій мережі є або непотрібними (нульовими значеннями), або надмірними. Тому найефективніша архітектура глибокої мережі матиме рідкісний зв'язок між активаціями, що означає, що всі 512 вихідних каналів не будуть мати з'єднання з усіма 512 вхідними каналами. Існують способи зрізати такі з'єднання, що призведе до незначного полегшення при обробці. Але ядра для паралельного матричного множення не оптимізуються в CUDA для GPU пакеті, а навпки, розпалалелювання робить їх навіть більш повільними, ніж їх щільні аналоги.

Таким чином, GoogleNet розробив модуль, що називається модулем запуску, який наближається до розрідженої CNN з нормальною щільною конструкцією. Оскільки лише невелика кількість нейронів ефективна, як зазначалося раніше, число згорткових фільтрів певного розміру ядра зберігається мало.

Перший початковий модуль GoogLeNet як приклад, який має 192 канали у якості вхідних даних. Він має всього 128 фільтрів розміру ядра 3 x 3 і 32 фільтра розміром 5 x 5. Порядок обчислень для фільтрів 5 x 5 становить 25 x 32 x 192, що може підірвати, коли ми йдемо глибше в мережу, коли ширина мережі та кількість фільтрів 5 x 5 збільшується. Для того, щоб уникнути цього, додатковий модуль використовує згортання 1x1 перед застосуванням ядра більшого розміру, щоб зменшити розмір вхідних

каналів. Це зменшує обчислення до 16 x 192 + 25 x 32 x 16. Всі ці зміни дозволяють мережі бути більш гнучкою.

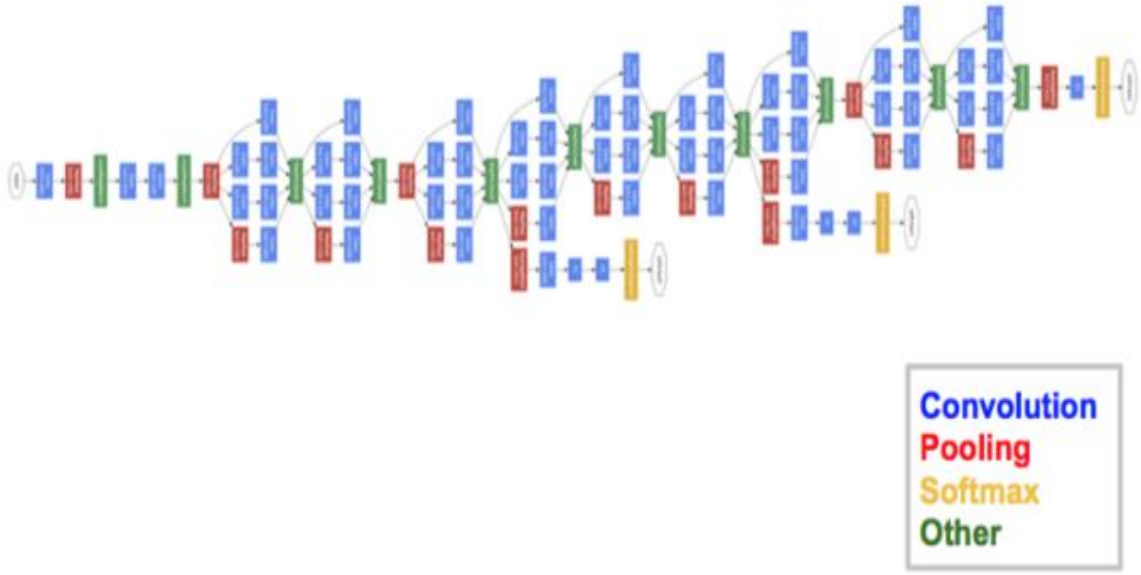

Рисунок 2.11 - Архітектура GoogleNet

Residual Neural Network представиляэ архітектуру з "пропуском з'єднання" і має важку нормалізацію пакетів. Такі пропущені з'єднання також відомі як замкнені блоки або закриті повторювані блоки та мають сильну схожість з останніми успішними елементами, що застосовуються в RNN. Архітектура таких згорткових мереж потребуэ важкий обчислювальних русурсів та складноъ побудови. Але така нейронна мережа показує високі результати на різних наборах даних. Сама а рхітектура схожа на VGGNet, що складається переважно з фільтрів 3 x 3.

Подібно до GoogLeNet, мережа використовує глобальне об'єднання, а потім клас класифікації. ResNets має неабиякі можливості і завдяки подібній архітектурі мережа має можливості навчитися із 152 шарами при великих обчислювальних можливостях. Мережа досягає кращої точності ніж VGGNet та GoogLeNet, а також має високу ефективність у швидкості обробки даних.

2.5. Згорткова мережа VGGNet

Як було вище описано згорткова мережа VGGNet має велику популярність у розпізнаванні об’єктів через свою ефективність та простоту.

По-перше, використання 3 × 3 фільтрів, оскільки 2 шари 3 × 3 фільтрів вже охоплюють область 5 × 5. Використовуючи 2 шари 3 × 3 фільтрів, ми маємо покриття області 5 × 5 (рис 2.4.1). Використовуючи вже 3 шари 3 × 3 фільтрів, ми маємо покриття області 7 × 7. Таким чином, великогабаритні фільтри, такі як 11 × 11 в AlexNet та 7 × 7 у ZFNet, дійсно не потрібні.

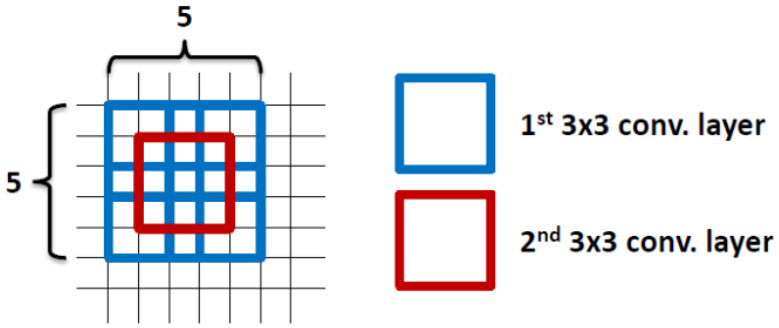

Рисунок 2.12 - Покриття фільтрами 3 на 3 області 5 на 5

Інша причина полягає в тому, що кількість параметрів через фільтри малого розміру менше. Припустімо, для кожного шару є лише 1 фільтр, 1 шару на вході, і виключити зміщення:

- 1 шар 11 × 11 фільтр, кількість параметрів = 11 × 11 = 121.
- 5 шару 3 × 3 фільтра, кількість параметрів = 3 × 3 × 5 = 45.

Кількість параметрів зменшується на 63%.

- 1 шар фільтра 7 × 7, кількість параметрів = 7 × 7 = 49.
- 3 шари 3 × 3 фільтрів, кількість параметрів = 3 × 3 × 3 = 27.

Кількість параметрів зменшено на 45%

- Використовуючи 1 шару фільтра 5 × 5, кількість параметрів = 5 × 5 = 25.
- Використовуючи 2 шари 3 × 3 фільтрів, кількість параметрів = 3 × 3 + 3 × 3 = 18.

Кількість параметрів зменшено на 28%.

Таким чином ми бачимо, що фільтри 3 х 3 мають дуже високу ефективність. Оскільки велика кількість параметрів має негативний вплив на продуктивність методу.

Також ефективність такої нейронної мережі залежить від кфлькості шарів та певних етапів, що покращують обробку зображень.

Наприклад, як ми бачимо VGG-11(LRN) має середній коефіцієнт помилки 10,5%, що містить додаткову операцію локальної відповіді на ліквідацію (LRN), запропоновану AlexNet. Порівнюючи VGG-11 та VGG-11 (LRN), частота помилок не поліпшується, що означає, що LRN не є корисним.

VGG-13 має 9,9% коефіцієнта помилки, що означає, що додаткова конфігурація допомагає точності класифікації.

VGG-16 (Conv1) має 9,4% коефіцієнта помилки, що означає, що додаткові три 1 × 1 конфігураційні шари допомагають точності класифікації. 1 × 1 згортання фактично допомагає збільшити нелінійність функції прийняття рішення. Не змінюючи розміри вводу та виводу, 1 × 1 згортання виконує проектування відображення в тій же високій розмірності.

VGG-16 має 8,8% коефіцієнта помилки, що означає, що глибока мережа навчання все ще покращується, додавши кількість шарів.

VGG-19 отримує 9,0% коефіцієнта помилки, що означає, що глибока мережа навчання не покращується, додавши кількість шарів, але не має високу різницю від попередньої структури.

Спостерігаючи додавання шарів один за іншим, можна помітити, що VGG-16 і VGG-19 починають зближуватися, а покращення точності сповільнюється. Таким чином ці два методи побудови є основними для задячі розпізнавання, щоб отримувати оптимальних результат (рис 2.13).

Багатомасштабне навчання також може вплинути на покращення ефективності при розпізнаванні. Оскільки об'єкт має інший масштаб у зображенні, якщо ми тренуємо мережу в тій же шкалі, ми можемо пропустити виявлення або мати неправильну класифікацію об'єктів з

іншими масштабами. Щоб вирішити це запропонована наступна система обробки. Для одномасштабного тренування зображення масштабується меншим розміром, рівним 256 або 384, тобто S = 256 або 384. Оскільки мережа приймає тільки 224 × 224 вхідних зображень. Масштабоване зображення буде обрізано до 224 × 224.

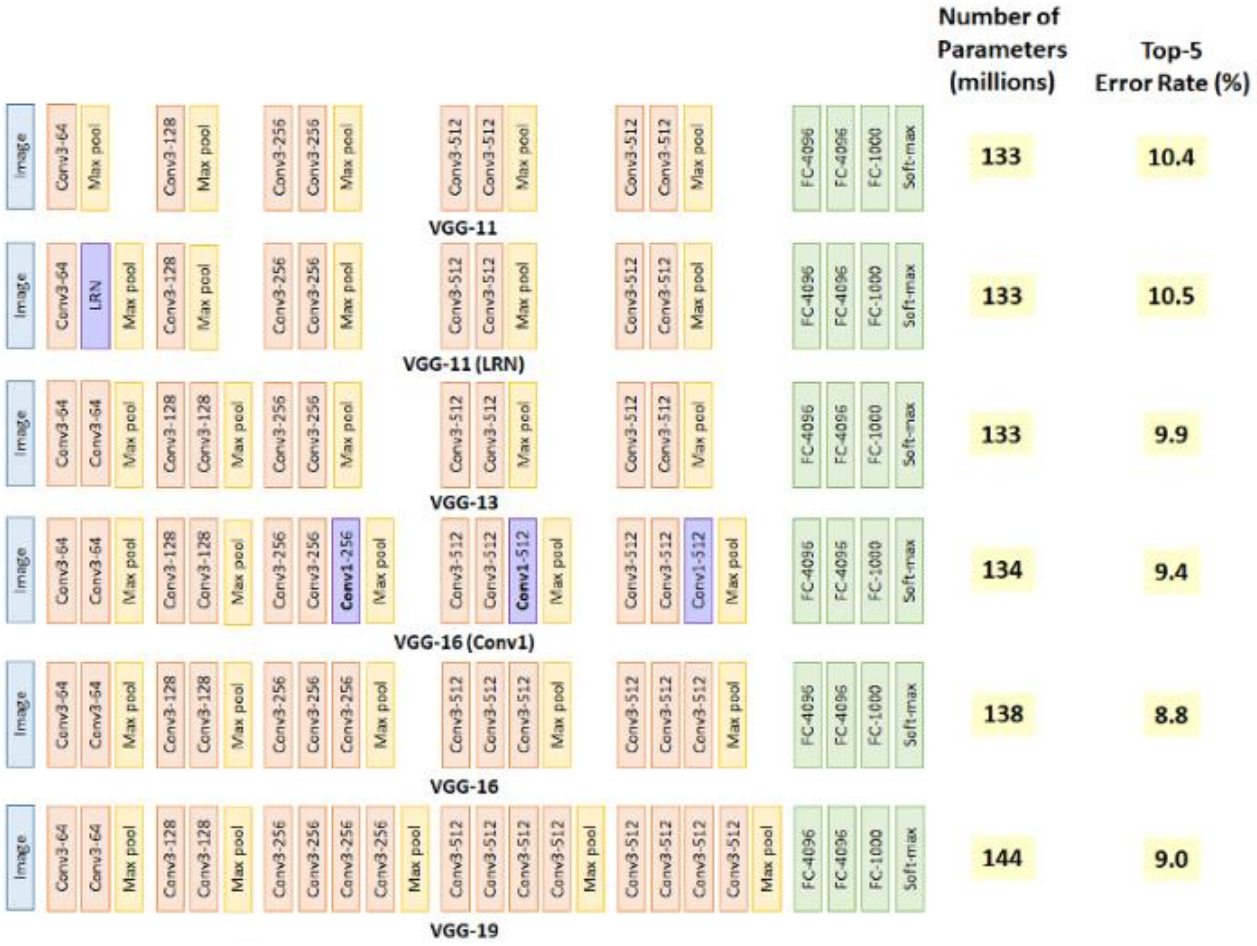

Рисунок 2.13 - Порівняльна характеристика структури VGG згорткової нейронної мережі

Концепція полягає в наступному у тому, що для багатомасштабного тренування зображення масштабується меншим розміром, рівним діапазону від 256 до 512, тобто S = [256; 512], потім обрізається до 224 × 224. Тому з діапазоном S вводять в мережу різного масштабу навчальні об'єкти (рис 2.14).

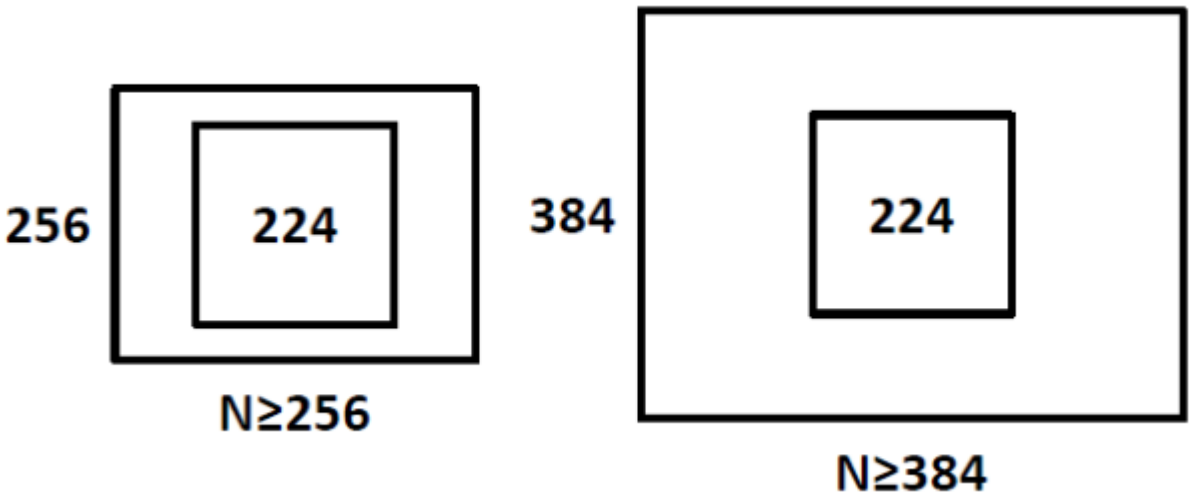

Рисунок 2.14 Масштабування зображення меншим розміром

За допомогою багатомасштабної підготовки результати розпізнавання для зображень різними розмірами об'єктів кількість помилок зменшилося. Так ми можемо побачити для різних стркутру згорткової нейронної мережі:

- VGG-13 зменшив рівень помилок з 9,4% (9,3%) до 8,8%.
- VGG-16 зменшив рівень помилок з 8,8% / 8,7% до 8,1%.
- VGG-19 зменшив рівень помилок з 9,0% / 8,7% до 8,0%.

Подібно до багатомасштабного навчання, багатошарове тестування також може зменшити частоту помилок, оскільки ми не знаємо розмір об'єкта в тестовому зображенні. Якщо будже відбуватися масштабування тестового образ різного розміру, ми можемо збільшити шанси правильної класифікації. Тобто наше зображення буде змінювати свої розміри для того аби збільшити можливість розпізнавання потрібного нам образу.

Використовуючи багатомасштабне тестування, але одномасштабне навчання, частота помилок зменшується.

У порівнянні з одномасштабними тренінгами одномасштабного тестування:

- VGG-13 зменшив рівень помилок з 9,4% (9,3%) до 9,2%.
- VGG-16 зменшив рівень помилок з 8,8% / 8,7% до 8,6%.
- VGG-19 зменшив рівень помилок з 9,0% / 8,7% до 8,7 / 8,6%.

Використовуючи як багатошарове навчання, так і тестування, частота помилок зменшується.

У порівнянні з тільки багатомовним тестуванням:

- VGG-13 зменшив рівень помилок з 9,2% (9,2%) до 8,2%.
- VGG-16 зменшив рівень помилок з 8,6% / 8,6% до 7,5%.
- VGG-19 зменшив рівень помилок з 8,7% / 8,6% до 7,5%.

Усі попередні дії забезпечують згортковій нейронній мережі VGG можливість показувати високі результати у розпізнаванні образів для пошуку. Саме на основі цього методу розпізнавання образів і будут відбуватися дослідження у розпізнаванні частин людського тіла та пошуки модифікації методу.

Основною та найефективнішою нейронною мережею є згорткова нейронна мережа VGG-19, що має 19 шарів обробки (кожен шар має по три кроки). Згорткова нейронна мережа VGG-19 має 16 кроків «Convolution» (згортки), 18 кроків «Relu», 5 кроків «Max Pollling», крок «Input», 2 кроки «Dropout», крок «Softmax», 3 кроки «Fully Connected Layer» (класифікації об’єкта) та крок «Classification output» (вивід даних). Як ми бачимо усі ці кроки забезпечуються повну обробку зображення і виведення основних характеристик знайдених за допомогою фільтрів. Велика кількість кроків «Relu» забезпечує фільтр непотрібних нам візуальних елементів, які заважають у розпізнаванні. Таким чином ми отримуємо тільки основні елементи та риси об’єкту для пошуку (рис 2.15).

```
ans = 47x1 Layer array with layers:

     1   'input'     Image Input             224x224x3 images with 'zerocenter' normalization
     2   'conv1_1'   Convolution             64 3x3x3 convolutions with stride [1  1] and padding [1  1  1  1]
     3   'relu1_1'   ReLU                    ReLU
     4   'conv1_2'   Convolution             64 3x3x64 convolutions with stride [1  1] and padding [1  1  1  1]
     5   'relu1_2'   ReLU                    ReLU
     6   'pool1'     Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
     7   'conv2_1'   Convolution             128 3x3x64 convolutions with stride [1  1] and padding [1  1  1  1]
     8   'relu2_1'   ReLU                    ReLU
     9   'conv2_2'   Convolution             128 3x3x128 convolutions with stride [1  1] and padding [1  1  1  1]
    10   'relu2_2'   ReLU                    ReLU
    11   'pool2'     Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    12   'conv3_1'   Convolution             256 3x3x128 convolutions with stride [1  1] and padding [1  1  1  1]
    13   'relu3_1'   ReLU                    ReLU
    14   'conv3_2'   Convolution             256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    15   'relu3_2'   ReLU                    ReLU
    16   'conv3_3'   Convolution             256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    17   'relu3_3'   ReLU                    ReLU
    18   'conv3_4'   Convolution             256 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    19   'relu3_4'   ReLU                    ReLU
    20   'pool3'     Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    21   'conv4_1'   Convolution             512 3x3x256 convolutions with stride [1  1] and padding [1  1  1  1]
    22   'relu4_1'   ReLU                    ReLU
    23   'conv4_2'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    24   'relu4_2'   ReLU                    ReLU
    25   'conv4_3'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    26   'relu4_3'   ReLU                    ReLU
    27   'conv4_4'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    28   'relu4_4'   ReLU                    ReLU
    29   'pool4'     Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    30   'conv5_1'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    31   'relu5_1'   ReLU                    ReLU
    32   'conv5_2'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    33   'relu5_2'   ReLU                    ReLU
    34   'conv5_3'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    35   'relu5_3'   ReLU                    ReLU
    36   'conv5_4'   Convolution             512 3x3x512 convolutions with stride [1  1] and padding [1  1  1  1]
    37   'relu5_4'   ReLU                    ReLU
    38   'pool5'     Max Pooling             2x2 max pooling with stride [2  2] and padding [0  0  0  0]
    39   'fc6'       Fully Connected         4096 fully connected layer
    40   'relu6'     ReLU                    ReLU
    41   'drop6'     Dropout                 50% dropout
    42   'fc7'       Fully Connected         4096 fully connected layer
    43   'relu7'     ReLU                    ReLU
    44   'drop7'     Dropout                 50% dropout
    45   'fc8'       Fully Connected         1000 fully connected layer
    46   'prob'      Softmax                 softmax
    47   'output'    Classification Output   crossentropyex with 'tench' and 999 other classes
```

Рисунок 2.15 - Кроки реалізації згорткової мережі VGG19

2.6. Обробка частин людського тіла при побудові каскаду

Розпізнавання людини або частин людини на зображенні дуже важка задача комп'ютерного зору. Алгоритми мають справу з дуже великим числом можливих людських положень, великі зміни зовнішнього вигляду людини (наприклад колір шкіри, одяг) та наявність великої множини людей в безпосередній близькості один від одного.

Таким чином метод повинен мати змогу вирішувати такі проблеми. Також одніє з проблем є саме визначення частин тіла людини, оскільки вони

можут бути приховані за певними перешкодами. Оскільки ми знаємо, що людське тіло має певний каскад, тобто тіло, голова, руки, ми можемо передбачити по каскаду чи це людина на зображенні і таким чином знайти ті чи інші частини тіла. Існує два шляхи обробки частин тіла на зображенні. При першому ми знаходимо образ людського тіла і після чого все працюємо із зменшенним зображенням. Таке зображення вже потенційно має всі потрібні частини тіла. Пісял пошуку всіх потрібних точок (голови, тулуба та кінцівок) ми формуємо каскад, з'єднуючи усі потрібні частини тіла. Для цього наша нейронна мережа повинна вміти розрізняти на зображенні усі потрібні нам елементи частин людського тіла. Такий підхід має перевагу в тому, що з великою вирогідністю не буде плутанити між частинами тіла різних людей, що знаходяться близько один до одного на зображенні. Другий підхід являє собою знаходження всіх кінцівок на зображенні одночасно. Тобто ми знаходимо усі можливі на зображенні ноги, руки, голови тощо. І пісял цього аналізуємо та з'єднуємо їх у каскад, що генерує повноцінне людське зображення (рис 2.16).

Але в такому випадку звичайно проблемою є саме з'єднаянн потрібних точок у повноцінний каскад, оскільки ми знаходемо усі частини окремо. Для цього нам потріба певний елемент асоціації меж точками, тобто характеристика, при якій ми будемо впевнені, що частини тіла належать до однієї й тієї самої людини. Один із можливих способів вимірювання асоціації – це виявити додаткову середню точку між кожною парою частин на кінці кінцівок, і перевірити наявність його позиції між кандидатами частина виявлення, як показано на рисунку 2.17.

Рисунок 2.16 – Розпізнавання чатин тіла людини на зображенні

Рисунок 2.17 – Розпізнавання чатин тіла людини на зображенні

Подібний елемент дає змогу визначити потрібні нам основні частини, на основі який, завдяки каскаду, ми приєднуємо інші, таким чином елементи

людського тіла з’єднуються. Але звичайно ці методи є зовсім недосконалими і серйогзні зміни у положенні людських тіл або при поганій якості зображення призведуть до поганих результатів при розпізнаванні.

2.7. Аналіз методу розпізнавання части людського тіла на основі VGG19

Як ми знаємо, одними з найефективніши і популярніших методів розпізнавання частин людського тіла є методи на основі згорткових нейронних мереж. Нейронна мережа VGG19 має досить високу ефективність та швидкість при навчанні, тому вона є дуже популярною ми розробці методів розпізнавання різних об'єктів і часто використовується для методів розпізнавання частин тіла людини.

Завдяки великому датасету COCO для розпізнавання части тіла людини методи розпізнавання мають можливість отримати потрібні елементи та характеристи для розпізнавання людей на зображенні. Отримавши класи для класифікації образів при розпізнаванні для нейронної мережі з’являється можливість розпізнавати частини людського на більшості зображеннях. Звичайно недоліком є те, що на розпізнавання, а особливо на навчання потрібно виликі обчислювальні ресурси та багато часу.

Таким чином метод розпізнавання за допомогою нейронної мережі показує досить високі результати я ми можемо побачити на рисунку 2.18 та 2.19.

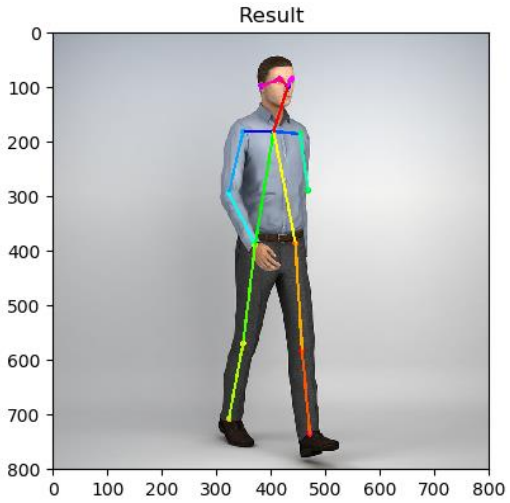

Рисунок 2.18 – Розпізнавання однієї людини методом розпізнавання части

тіла людини на основі VGG19

Як видно із зображень, при хорошому освітленні та деталізації, елементи людського тіла знаходяться досить точно. Також, як ми знаємо, згорткова нейронна мережа при розпізнаванні видає певне число – оціну або «score», що являє собою вппевненість нейронної мережі в тому, що ця точка є плечем, стопою чи іншою частиною тіла. Завдяки цьому ми можемо проаналізувати на скільки нейронна мережа має вискокі результати на різних хображеннях. Якщо певна частина тіла не була знайдена значення дорівнюють нуля. Оцінка може бути від 0 до 1, де 0 означає що елемент не був знайдений, а 1 що елемент був знайдений і мережа на 100 впевнена, що ця точка на зображенні є певною частиною тіла.

Значення оцінки у розрізі від 0 до 1, наприклад 0.51 або 0.67 дають змогу нам зрозуміти, що данний елемент є мало помітним, а не дуже схожим на зображенні на об’єкт нашого пошуку. Таким чином певними маніпуляціями ми можемо збільшувати кількість знайдених нами частин тіла або збільшувати значення оцінки потрібних об’єктів пошуку.

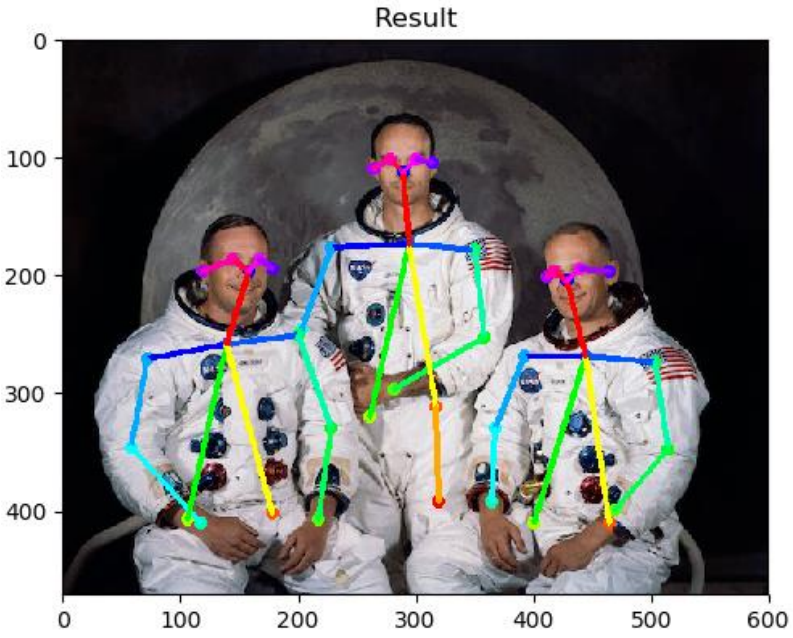

Рисунок 2.19 – Розпізнавання людей на зображенні високої якості

2.7. Висновки за розділом

Як ми розглянули існує велика кількість методів для розпізнавання образів на зображеннях х різними архітектурами. Також було проаналізовано метод розпізнавання зображень на основі згорткової нейронної мережі VGG19. Сам метод має можливість розпізнавати тіла людей та їх частини на зображеннях досить чітко. Кількість кроків такої нейронної мережі дає можливість знаходити характерні образи для пошуку різних образів, які нейронна мережа навчилася розпізнавати. Звичайно такий метод має і велику кількість недоліків, які в основному пов'язані з обробкою зображень. Часто саме погана якість зображень не дає змогу чітко знаходити образи, які доступні людському оку.

3. МОДИФІКАЦІЯ МЕТОДУ РОЗПІЗНАВАННЯ ЛЮДСЬКОГО ТІЛА

3.1. Недоліки методу розпізнавання об’єктів

Більшість проблем при розпізнаванні образів за допомогою згорткових нейронних мереж пов’язані або з поганою якістю і деталізацією зображення, або з поганим аналізом зображення для пошуку потрібних характеристик і визначенням об’єктів.

При розпізнаванні предметів за допомогою згорткових нейронних мереж велику роль відіграють датасети. Оскільки від якісного і великого датасету залежить точність розпізнавання потрібного образу. Так наприклад для пошуку собк або котів нам будет потрібно мати великий датасет розміром більше 100 тисяч зображень з усіма типами пород, у різних позіціях та на різному фоні. Таким чином наша мережа буде мати можливість працювати з різними видами зображень. Якісний датасет повинен мати більше 120 тисяч зображень з різним наповненням, але звичайно кожне зображення повинне мати об’єкт пошуку. Не маючи такого датасету якість розпізнавання буде досить низька. Але зараз існує вже створенні великі датасети для більшості основних об’єктів. Таким чином ми маємо змогу навчати мережу розпізнавати велику кількість потрібних нам образів.

Також одним із недолік при розпізнаванні образів за допомогою нейронних мереж є потреба у великих обчислювальних ресурсах. Усі інструменти розпізнавання маніпулюють великими векторами та матрицями даних. Іноді для того аби навчити нейронну мережу на хорошому датасеті вам будет потрібно кулька днів, а то й тиждень, для того аби були оброблені усі зображення та створені класифікатори. У більшості випадків інструменти для написання та роботи з нейронними мережами використовують GPU з великою кількістю пам’яті, тому на простих настільних комп'ютерах навчання буде неможливим. Також для розпізнавання і обробки зображень буде потрібен також і велика кількість

оперативної пам'яті та ресурси процесора. Таким чином для роботи із розпізнаванням потрібно мати досить великі обчислювальні ресурси для отримання хорошої швидкості обробки даних. Для отпимізації цього процесу часто використовують зображення малих розмірів або штучно їх зменшують при цьому втрачаючи якість зображення. Але завдяки цьому швидкість обробки даних збільшується у кілька разів. Так наприклад можна побачити на рисунку 3.1 швидкість обробки даних у різних , але досить сильних процесорів та зображень зменшеного розміру (стандартне 1080x720).

~0.6 FPS	~4.2 FPS @ 368x368	~10 FPS @ 368x368
2.8GHz Quad-core i7	2.8GHz Quad-core i7	Jetson TX2 Embedded Board

Рисунок 3.1 – Швидкість обробки даних для різних процесорів з показана у кадрах в секунду

Таким чином можна побачити, що зменшення справді збільшує швидкість обробки, але навість при використанні таких потужних систем як Jetson TX2 Embedded Board ми маємо всього 10 кадрів в секунду при розпізнаванні на живому відео.

Однією з найбільших проблем розпізнавання є робота із зображеннями в поганій якості та деталізації. В таких випадках часто втрачаються контури зображень, що є важливими ознаками і характеристами при розпізнаванні образів. Згорткова нейронна мережа завдяки фільтрам визначає певні характеристики і основні контрури, за якими вона може розпізнати певний образ.

Код знизу має позначення класу частин тіла людини завдяки якому ми розпізнаємо ті чи інши елементи нашого каскаду.

```
class CocoPart(Enum):
    Nose = 0
    Neck = 1
    RShoulder = 2
    RElbow = 3
```

```
RWrist = 4
LShoulder = 5
LElbow = 6
LWrist = 7
RHip = 8
RKnee = 9
RAnkle = 10
LHip = 11
LKnee = 12
LAnkle = 13
REye = 14
LEye = 15
REar = 16
LEar = 17
```

У випадку коли зображення має погану деталізацію, якість і погану яскравість контури на об’єктах стають не такими помітними таким чином нейронна мережа може не розпізнає певні характеристики образу. У таких випадках розпізнавання дає дуже низькі результати при пошуку частин тіла. У більшості знайдених частин тіла часто не точна позиція та оцінка дуже низька (рис. 3.2), а якщо зображення вхагалі темне тоді пошук взагалі не дає жодних результатів.

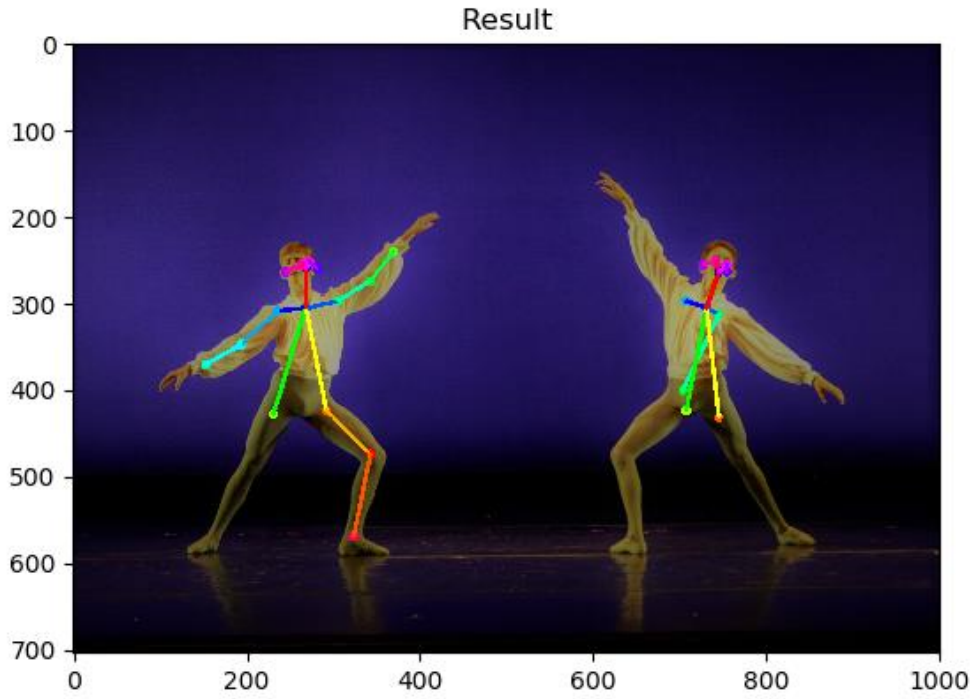

Рисунок 3.2 – Пошук на зображенні поганої якості

Як ми бачимо на такому зображенні були знайдені далеко не всі частини тіла. Також наша нейронна мережа показує які характеристи вона знайшла на зображенні по вектору X та У. Ознаки розпізнаються таким чином, що контури розпізнаються по векторам X та У завдяки фільтрам. Якщо ми подивимося на рисунок 3.2 та 3.3 ми зможемо побачити можливі елементи частин тіла як образи, на які орієнтується нейронна мережа. Завдяки таким образам і відбувається пошук елементів.

Також важливим елементом є оцінка кожної точки окремо. На основі оцінок виноситься вирок чи ця точка є насправді тим місцем що було ціллю пошуку, чи можливо це лише ілюзія. Результати з оцінками близькими до нуля завжди відкидаються. Оцінки з результатом 0.20 та меньше важаються дуже сумнівними. Так я це зображення має досить погану деталізацію результати більшості точок досить низькі. А деякі точки мають оцінку 0 (такі точки не записуються до класу людини).

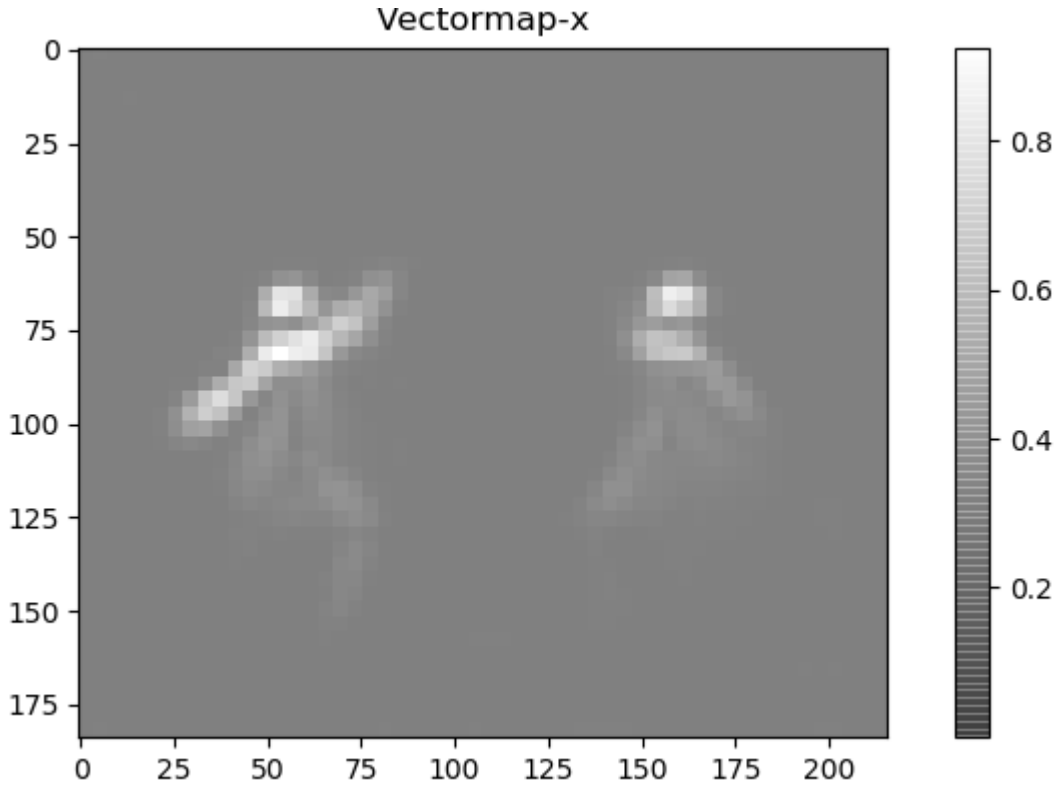

Рисунок 3.3 – Основні характеристики по вектору X

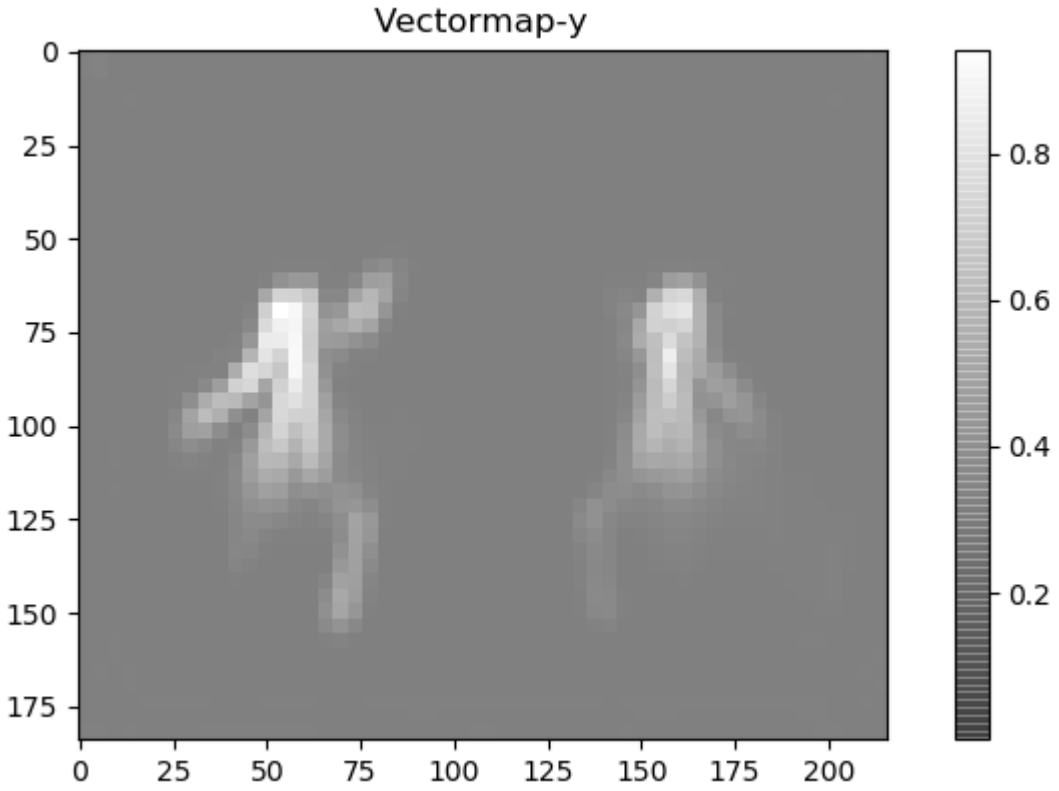

Рисунок 3.4 – Основні характеристики по вектору У

Як ми бачимо на рисунку 3.5 більшість частин тіла мають досить низькі результати. Середній результат оцінки близько 0.50, що є зовсім невисоким. Деякі точки (частини тіла 4, 7, 9, 10) взагалі не були знайдені (отримали оцінку 0), через нечіткі характеристики.

На основі оцінки результатів є можливість аналізувати ті чи інші моменти роботи з модифікацією. Таким чином робота з попередньою обробкою зображення на етапі “INPUT” може дати покращення при роботі із погано деталізованими зображеннями.

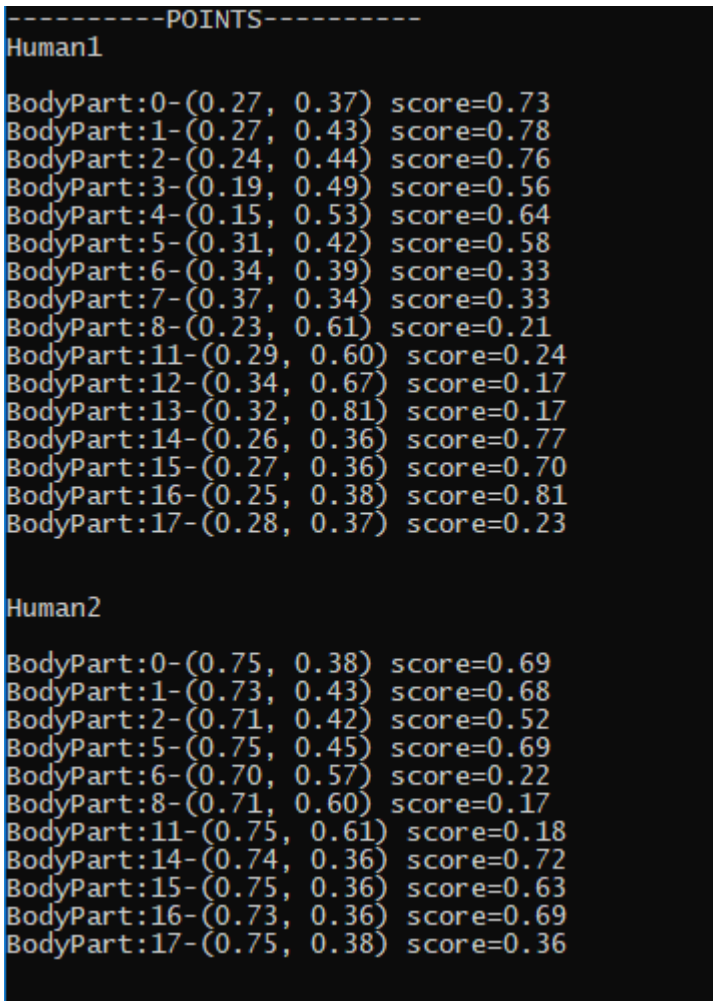

Рисунок 3.5 – Оцінка точок при розпізнаванні частин тіла людини

3.2. Модифікація методу розпізнавання за допомогою згорткової нейронної мережі

Як було розглянуто у попередньому розділі важливим елементом у розпізнаванні відіграє саме хороша деталізація зображення. Погано деталізовані зображення показують погані результати через важкість аналізу даних, пошуку характеристик тощо. Оскільки найчастіше датасети мають хороші деталізовані зображення з чітко вираженими об’єктами для пошуку, ситуація з погано деталізованими зображеннями або зображеннями, що мають велику кількість темних відтінків, які зливаються, досить непроста при розпізнаванні.

Тому було вирішено провести модифікацію на етапі “INPUT” згорткової нейронної мережі. На цьому етапі зображення, яке надходить до

програми проходить певну обробку і дані приводяться до вигляду коли з ними можуть працювати наступні кроки. Тобто зображення, яке має кольорову гамму RGB (R – червони, G – зелений, B – синій), розбивається на 3 шари – 3 матриці даних. Кожна матриця має значення від 0 до 255. Таким чином ми отримуємо 3 матриці із значеннями кожного пікселя і модифікуємо наступними кроками. Тому після кроку “INPUT” є можливість додати етпа попередньої обробки зображення перед тип як передходити до кроку згортки. Етап попередньої обробки буде отримувати потрібні дані із зображення та збільшувати його чіткість, контраст і яскравість. Елементи матриці змінюються таким чином, що зображення отримує більшу деталізацію. Збільшення деталізації відбувається завдяки накладанню Lab кольорової моделі. LAB — система задання кольорів, що використовує як параметри світлосилу, відношення зеленого до червоного та відношення синього до жовтого. Ці три параметри утворюють тривимірний простір, точки якого відповідають певним кольорам.

LAB має три параметри для опису кольору: світлосила *L* — рівень освітлення сцени та два хроматичні параметри. Перший (умовно позначений латинською літерою *a*) вказує на співвідношення зеленої і червоної складової кольору, другий (позначений літерою *b*) — співвідношення синьої та жовтої складової. Колірна модель L*a*b розроблялась як апаратно-незалежна, тобто вона задає кольори без врахування особливостей відтворення кольорів (рис 3.6).

```
clahe = cv2.createCLAHE(clipLimit=3., tileGridSize=(8,8))

lab = cv2.cvtColor(image, cv2.COLOR_BGR2LAB)  # convert from BGR to LAB color space
l, a, b = cv2.split(lab)  # split on 3 different channels

l2 = clahe.apply(l)  # apply CLAHE to the L-channel

lab = cv2.merge((l2,a,b))  # merge channels
```

Рисунок 3.6 – Код покращення деталізації зображення

На відміну від кольорових просторів RGB чи CMYK, які є, по суті, набором апаратних даних для відтворення кольору на папері чи на екрані

монітора, Lab однозначно визначає колір. Тому Lab широко використовується в програмному забезпеченні для обробки зображень як проміжного кольорового простору, через який проходить конвертування даних між іншими кольоровими просторами (наприклад, з RGBсканера в CMYK печатного процесу). При цьому особливі властивості Lab зробили редагування в цьому просторі потужним інструментом корекції кольору [10].

У колірному просторі Lab значення світлості відокремлено від значення хроматичної складової кольору (відтінок, насиченість). Світлість задана координатою L (змінюється від 0 до 100, тобто від найтемнішого до найсвітлішого), хроматична складова — двома декартовими координатами a и b. (Перша позначає положення кольору в діапазоні від зеленого до червоного, друга — від синього до жовтого.)

Завдяки характеру визначення кольору в Lab з'являється можливість окремо впливати на яскравість, контраст зображення і на його колір. У багатьох випадках це дозволяє прискорити обробку зображень. Lab надає можливість вибіркового впливу на окремі кольори в зображенні, посилення кольорового контрасту, незамінними є можливості, які цей колірний простір надають для боротьби із шумом на цифрових фотографіях.

До тестів на вечірніх зображеннях з дуже поганим освітленням результати розпізнавання були дуже низькими.

Рисунок 3.7 – Результати пошуку людини на темному зображенні

Як видно на рисунку 3.1.7 програма розпізнавання не змогла знайти потрібні нам частини тіла і взагалі знайти образ людини. Робота з таким зображенням не дає жодних результатів.

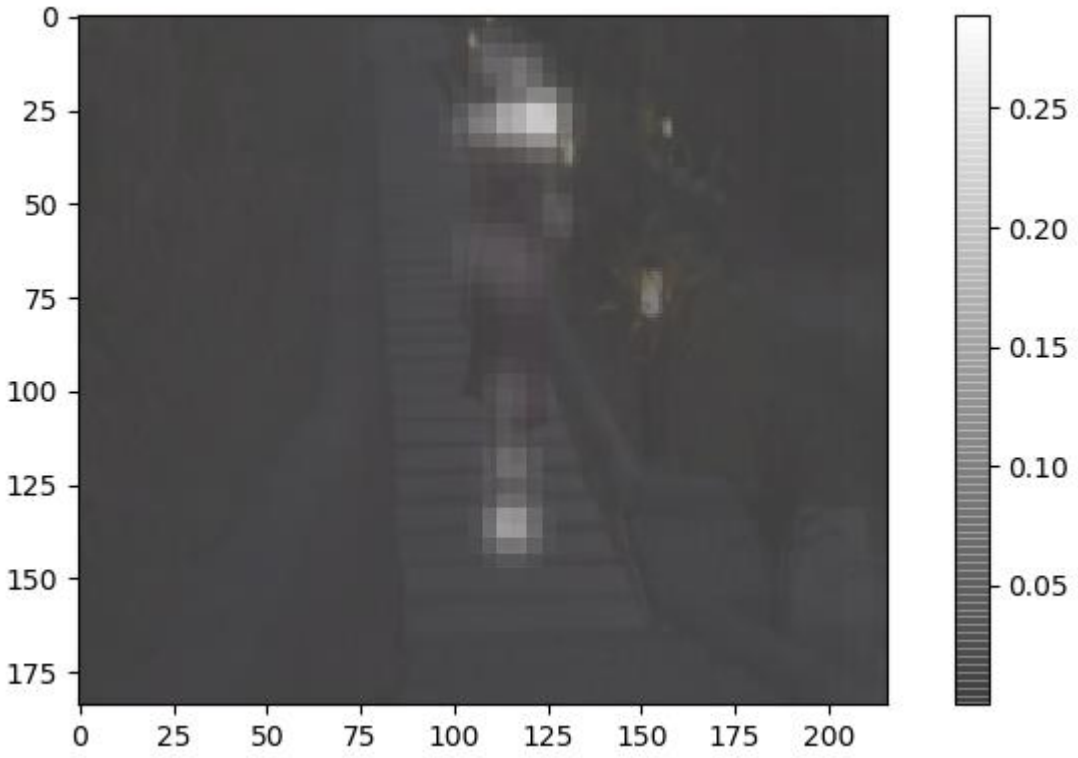

Рисунок 3.2.7 – Оцінка частин тіла людини на зображенні

Після додавання кроку попередньої обробки результати на певних

зображеннях стали кращими. На темних зображеннях де були знайдено тільки пара невелика кількість частин тіла або взагалі не було розпізнано людське тіло результати стали кращими (рис. 3.1.8).

Рисунок 3.8 – Результати пошуку людини на темному зображенні

Так на вечірньому зображенні було знайдено людину, де до модифікацій не було позитивних результатів. Як видно було знайдено вже основні характерні людині частини тіла. Результати із нульових оцінок перетворились на певні результати. Вже були знайдені точки № 1,2,5,6,8,11,16,17 (рис. 3. 9). Майже всі значення мають досить низький результат оскільки зображення має погану якість.

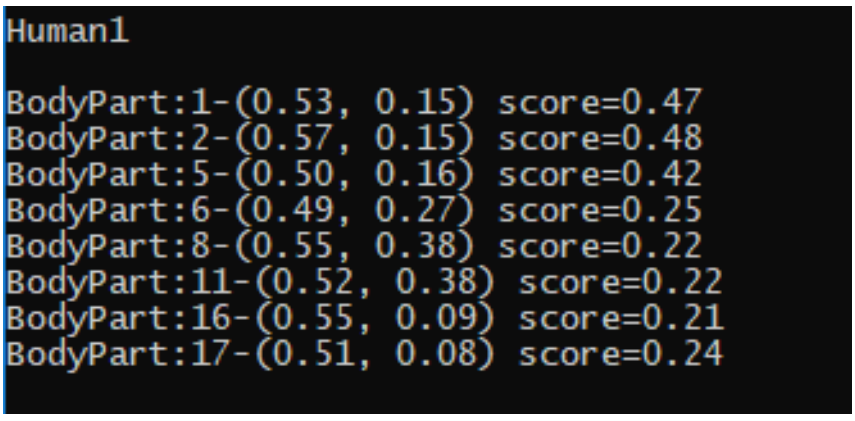

Рисунок 3.9 – Оцінка пошуку частин тіла після модифікації

Зображення проходить кілька кроків модифікації після чого обирається найкращий результат. Таким чином було досягнуто результату де було

знайдено на такому зображенні 9 різних точок – частин тіла людини (№ 1,2,5,6,8,11,12,16,17). Оцінка частин тіла зрівняно з попередньою теж стала вищою. Так середня оцінка попереднього результату була близько 0.32, в той час я оцінка нових результатів не тільки дала приріст у впевненості згорткової нейронної мережі у знайдених точках, але й були знайдені додаткові частини людського тіла, які до цього не були розпізнані (рис. 3. 10).

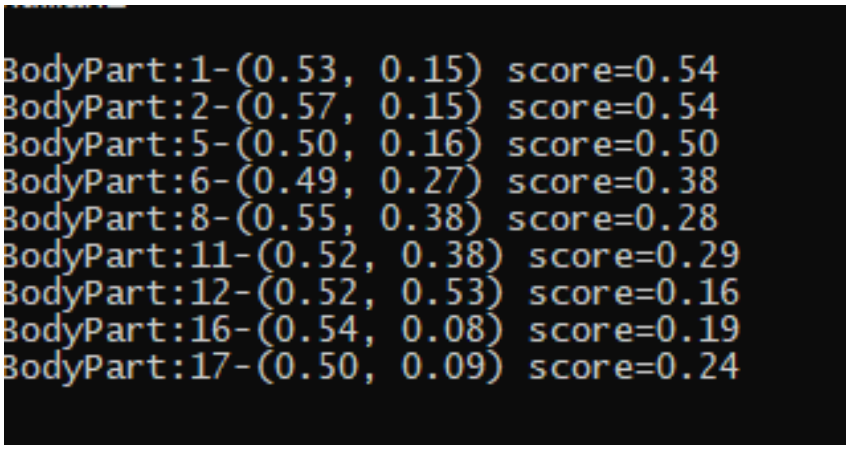

Рисунок 3.10 – Оцінка пошуку частин тіла після модифікації

Обробка даних зображення у самій програмі проходить у кілька етапів як показано на рисунку 3.11. Кілька етапів обробки потрібно для того, аби знайти оптимальний результат. Оскільки кожне зображення має свої характеристики, для кожного потрібна різна інтенсивність обробки. Після кількох етапів модифікації вхідних даних зображення, які спочатку розбиваються на 3 шари даних, в залежності від кольорової моделі, обирається оптимальна на основі кількості знайдених точок та оцінки. Точки на частинах тіла є більш важливими, тому іноді обираються результати з меншими оцінками але більшою кількістю точок (частин тіла людини).

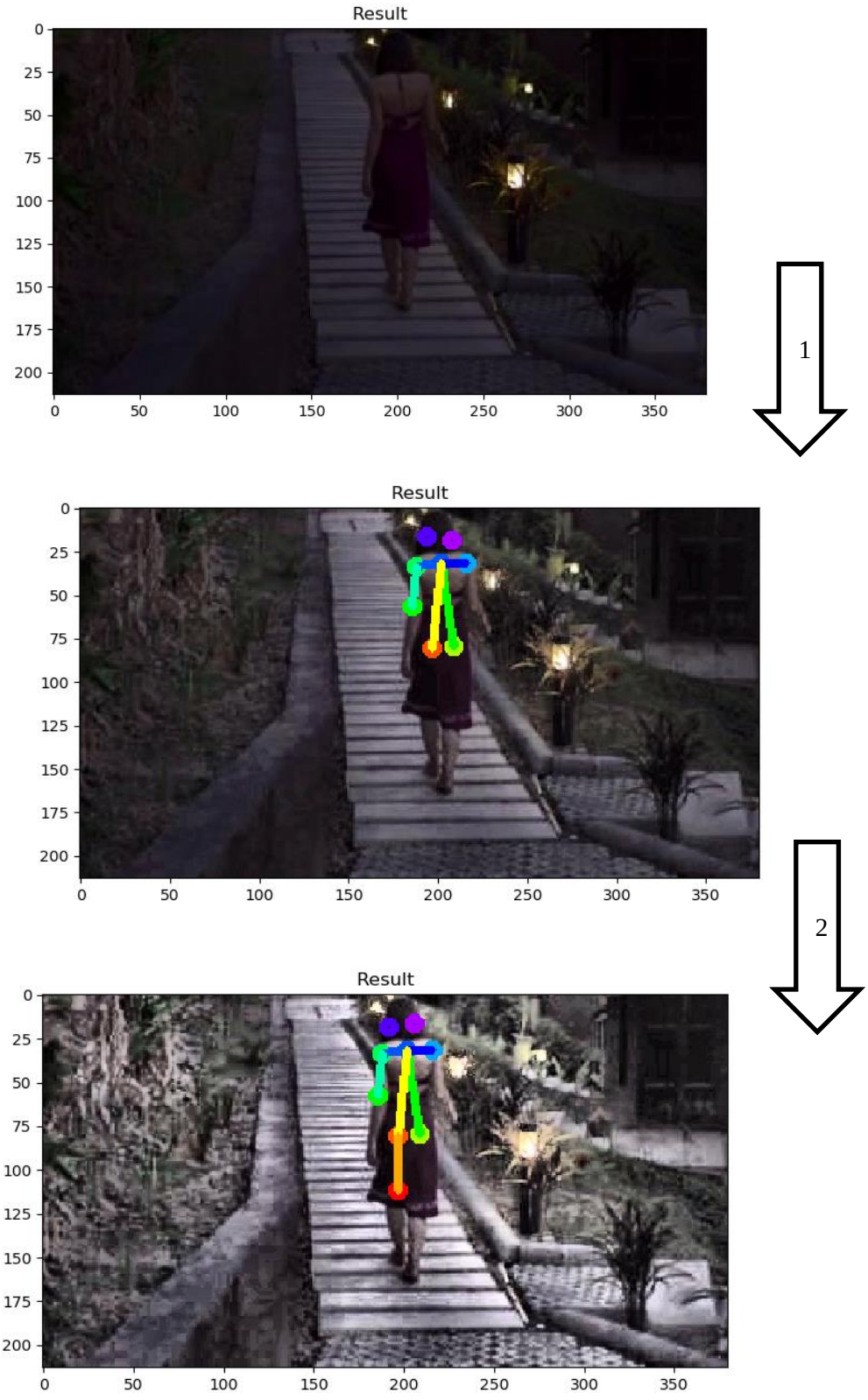

Рисунок 3.11 – Кроки обробки зображення

Таким чином ми маємо три кроки після яких отримуємо зображення із найкращим результатом розпізнавання (рис 3.12).

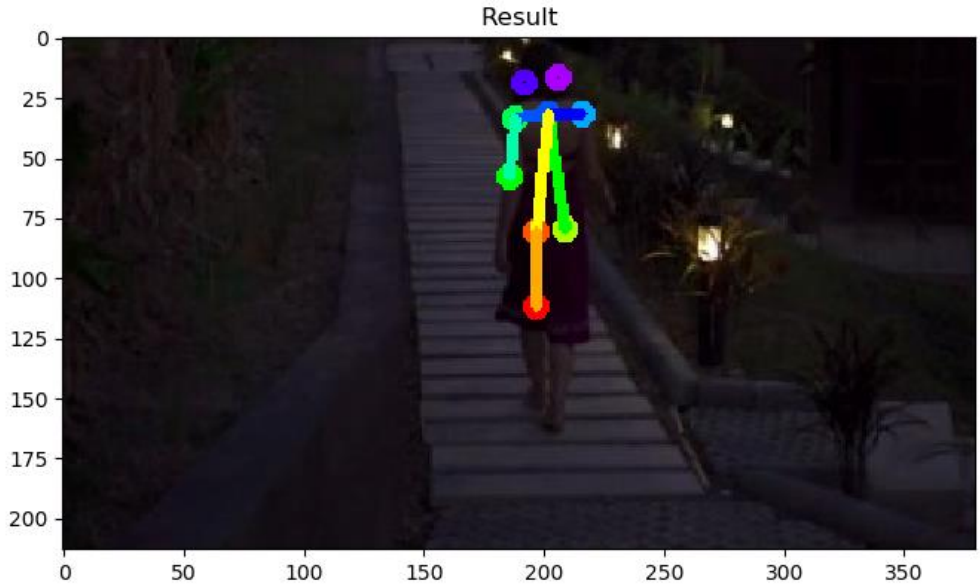

Рисунок 3.12 – Кінцевий результат виведення зображення

3.3. Аналіз модифікованого методу розпізнавання частин людського тіла

Як ми вже побачили модифікований метод розпізнавання частин тіла зміг знайти точки, які попередньо не було знайдено і вивів початкове зображення з певним каскадом. Також важливим елементом є оцінка кожної точки після отримання каскаду. Тести показали, що на багатьох зображеннях з поганим освітленням і деталізацією, на яких попередньо не було знайдено жодних точок на тілі людини, було знайдено точки основного каскаду. На тих зображеннях де були знайдені частини тіла оцінка стала вищою. Якщо порівнювати оцінки, ми можемо прачити як змінюваась оцінка на різних етапах обробки.

Перший етап обробки зміг дати результат у 8 знайдених кінцівок:

- BodyPart:1 score=0.47
- BodyPart:2 score=0.48
- BodyPart:5 score=0.42
- BodyPart:6 score=0.25
- BodyPart:8 score=0.22
- BodyPart:11 score=0.22

- BodyPart:16 score=0.21
- BodyPart:17 score=0.24

На більшості точок результат досягав від 0.21 до 0.25, деякі точки мали оцінку близько 0.42-0.48, але такі результати все одно дуже низькими. Іноді такі дані можна сприйняти за ілюзію або силует і відкинути, оскільки на зображеннях часто можуть виникати певні структури схожі на постаті людей і низьким результатам не завжди можна довіряти.

Тому для тестів був доданий ще один етап попередньої обробки зображень. Другий етап обробки:

- BodyPart: 1 score=0.54
- BodyPart: 2 score=0.54
- BodyPart: 5 score=0.50
- BodyPart: 6 score=0.38
- BodyPart: 8 score=0.28
- BodyPart: 11 score=0.29
- BodyPart: 12 score=0.16
- BodyPart: 16 score=0.19
- BodyPart: 17 score=0.24

Як видно з другого етапу обробки зображення оцінка точок стала вищою. По перше була знайдена додаткова точка, що позначає ліве коліно, яке до цього не було розпізнано на зображенні. Також видно, що більшість точок стали мати бульшу оцінку. Значення змінились наступним чином:

- BodyPart: 1 score=0.47 => score=0.54
- BodyPart: 2 score=0.48 => score=0.54
- BodyPart: 5 score=0.42 => score=0.50
- BodyPart: 6 score=0.25 => score=0.38
- BodyPart: 8 score=0.22 => score=0.28
- BodyPart: 11 score=0.22 => score=0.29
- BodyPart: 12 score=0.00 => score=0.16

- BodyPart: 16 score=0.21 => score=0.19
- BodyPart: 17 score=0.24 => score=0.24

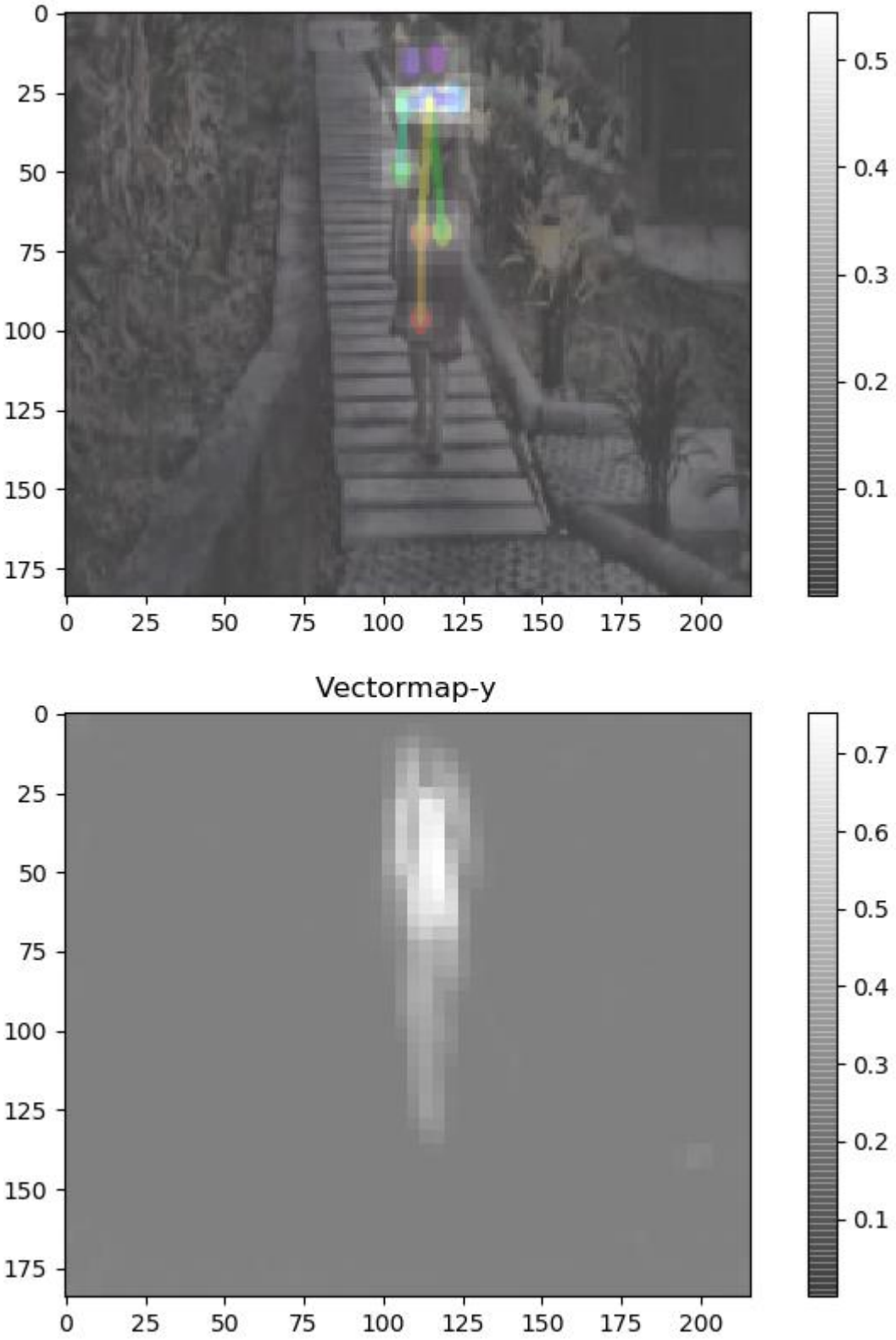

Рисунок 3.14 – Розпізнавання частин тіла після модифікації

Як видно оцінка змінилась в середньому на 0.06-0.09 у більшу сторону (рис 3.14). Таким чином можна сказати, що були отримані певні результати

порівняно з першим кроком обробки. Такий результат, хоча і невеликий збільшує точність визначення каскаду людського тіла і впевненість в тому, що ми знайшли саме людину, а не ілюзію чи силует. Також можна побачити елементи розпізнавання на рисунку 3.3.1.

Як видно із зображення, особливо по вектору У, силует людини має більш чітку фарму. Також на зображені видно як виділяються основні точки розпізнавання на тілі людини. З цього зображення візуально можна зрозуміти на скільки покращелося розпізнавання елементів.

Також важливо є сказати те, що іноді оцінка певних точок на тілі людини (частин тіла) показує дещо нижчі результати порівняно із попереднім кроком. Це зумовлено тим, що іноді пікселі змінюють свої значення і точка не так віражена, як була раніше. Але часто такі зміни бувають незначними. Так, якщо розглядати попредній приклад, ми ммможемо побачити, що точка 16 мала оцінку 0.21, а після модифікації оцінка впала до 0.19. Така зміна є допустимою, якщо загальний результат значко кращий попереднього кроку.

Таким чином модна зробти висновок, що іноді дозволяється пожертвувати оцінкою певних точок, але при цьому збільшити оцінку інших, якщо різниця значна. А іноді в таких випадках можуть бути знайдені додаткові точки на тілі людини, тому така невелика втрата оцінки є дозволеною. Така втрата оцінки є взагалі незначною. Інший випадок коли після модифікації ми втрачаєму одну з точок, але знаходимо іншу. В такому випадку є можливість об'єднати результати попереднього тесту і додати його до наступного кроку для того аби побачити більш повну картину.

Для кращих тестів були обрані зображення хорошої якості зроблені увечері. Такі зображення майже завжди показують хороші результати. Основна різниця між ними у оцінках знайдених точок. Для таких зображень виконується тільки один етап попередньої обробки. Оскільки надмірна маніпуляція над даними зображення також може зіграти генативно і зображення стане нечитабельним. Оскільки такі зміни призведуть до

великої кількості шумів.

Як ми бачимо із рисунку 3.15 на темному зображенні було знайдено майже усі точки до модифікацій. Кожна з точок має досить непогані результати оцінки:

- BodyPart:1-(0.54, 0.43) score=0.76
- BodyPart:2-(0.56, 0.43) score=0.79
- BodyPart:3-(0.56, 0.52) score=0.58
- BodyPart:4-(0.57, 0.54) score=0.40
- BodyPart:5-(0.50, 0.43) score=0.69
- BodyPart:6-(0.50, 0.52) score=0.62
- BodyPart:7-(0.49, 0.58) score=0.63
- BodyPart:8-(0.55, 0.58) score=0.65
- BodyPart:9-(0.55, 0.69) score=0.44
- BodyPart:10-(0.53, 0.83) score=0.30
- BodyPart:11-(0.51, 0.58) score=0.63
- BodyPart:12-(0.51, 0.70) score=0.65
- BodyPart:13-(0.52, 0.82) score=0.31
- BodyPart:16-(0.55, 0.38) score=0.37
- BodyPart:17-(0.52, 0.38) score=0.66

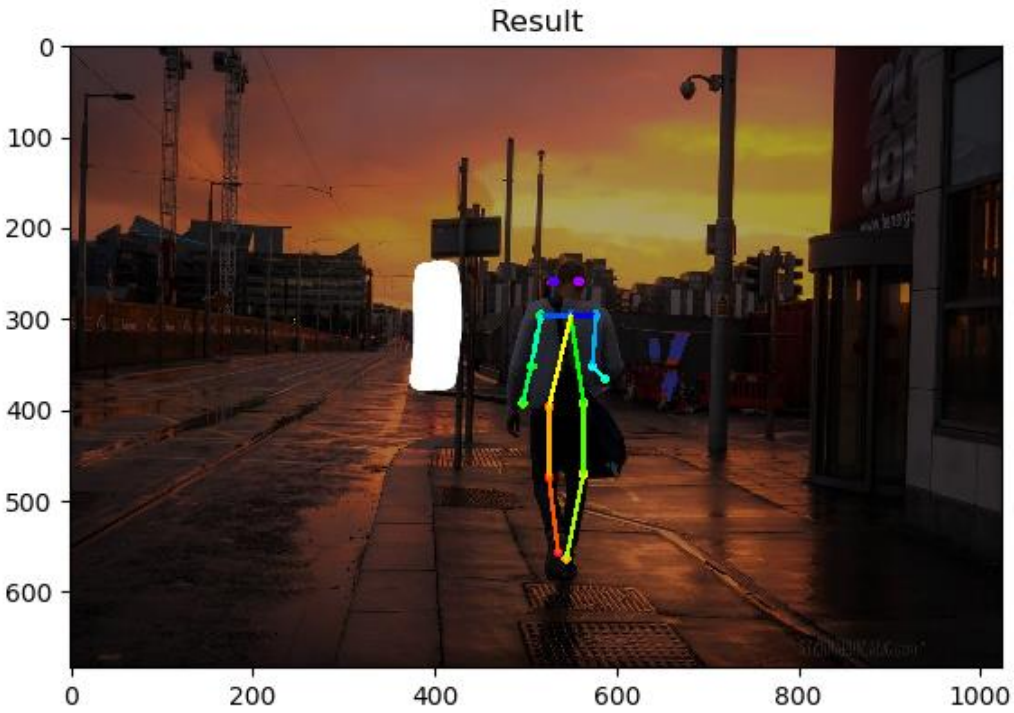

Рисунок 3.15 – Розпізнавання частин тіла після модифікації на якісному темному зображенні

Як видно середній результат оцінки розпізнавання близько 0.57, що для таких зображень досить непоганим. Після додавання кроку модифікації значення оцінки дещо змінилися. Після модифікації результати стали наступними:

- BodyPart:1-(0.54, 0.43) score=0.77
- BodyPart:2-(0.56, 0.43) score=0.80
- BodyPart:3-(0.57, 0.52) score=0.56
- BodyPart:4-(0.58, 0.53) score=0.51
- BodyPart:5-(0.50, 0.43) score=0.81
- BodyPart:6-(0.48, 0.52) score=0.64
- BodyPart:7-(0.48, 0.58) score=0.65
- BodyPart:8-(0.55, 0.57) score=0.67
- BodyPart:9-(0.56, 0.67) score=0.44
- BodyPart:10-(0.52, 0.86) score=0.36
- BodyPart:11-(0.51, 0.57) score=0.67

- BodyPart:12-(0.51, 0.69) score=0.62
- BodyPart:13-(0.52, 0.83) score=0.32
- BodyPart:16-(0.55, 0.39) score=0.38
- BodyPart:17-(0.52, 0.39) score=0.65

Як видно з цих результатів більшість оцінок розпізнавання мають близькі результати, але в середньому вищі на 0.01-0.05. Деякі з них стали вищими на 0.10-0.12, що є досить високим результатом. Так наприклад, точки 4 і 5 підвищи значно свої значення, близько 20% від початкового. Так після модифікації ми можемо побачити результати розпізнавання точок (рис 3.16).

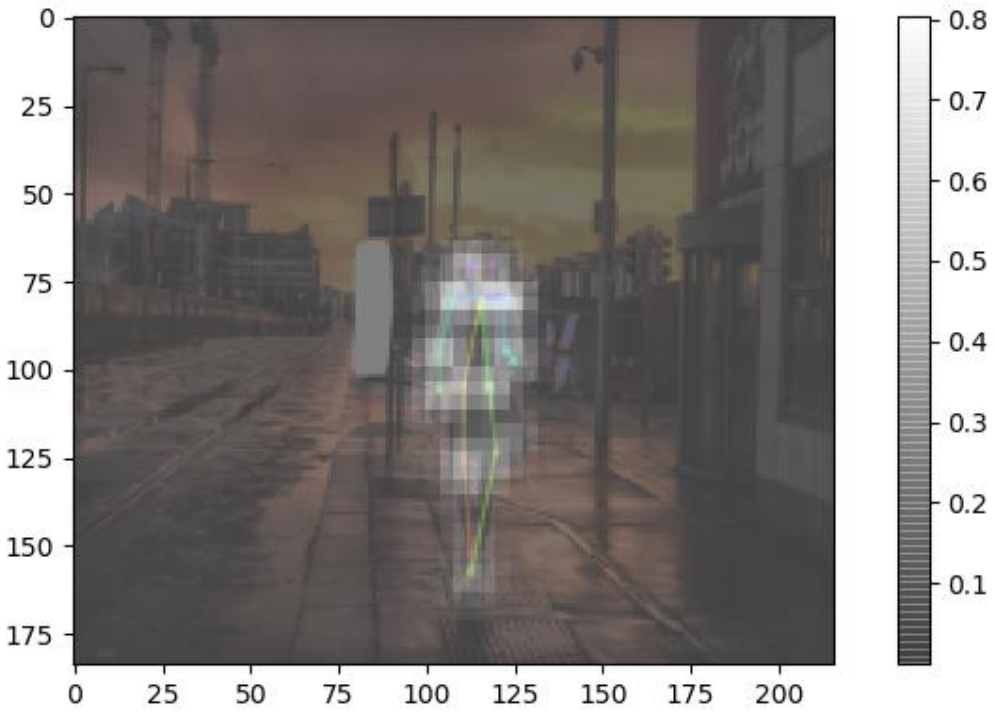

Рисунок 3.16 – Точки та оцінка розпізнавання частин тіла

Тому навіть невелике збільшення оцінки кожної точки є важливим. Іноді згорткова нейронна мережа сприймає силуети за постатті людей і визначає їх скелет, хоча і з малою кількістю точок. Звичайно навіть людське око часто може помилятися у поганих умовах. Так часто може комусь здаватися певні образи чи силуети, які будуть сприйті за людей. Таким

чамим чином і згорткова нейронна мережа може сприйняти певний образ за людський, хоча він буде помилковим.

3.4. Висновки за розділом

Однією із основних проблем розпізнавання у більшості методів є робота із погано деталізованими та темними зображеннями. Часто нічні камери та відеокамери не мають технологій якісної обробки даних у поганих умовах. Таким чином розпізнавання на відео та зображеннях майже неможливе.

Згорткові нейронні мережі мають етап обробки даних для подальшої передачі їх на наступні кроки. Додавши до цього етапу попереднб обробку даних зі збільшенням деталізації і збільшенням чіткості елементів дає змогу краще обролювати дані на наступних етапах.

Тести показали, що попередня обробка даних покращує зображення і таким чином розпізнавання давало значно кращі результати. В тестах використовувалися темні зображення поганої якості на яких попередньо не було знайдено всіх точок частин тіла. Після модифікації результати стали кращими і більшість основних точок для каскаду тіла людини були знайдені.

ВИСНОВКИ

Метою магістерської дисертації був аналіз існуючих методів розпізнавання людського тіла та його частин, а також пошук можливих можифікацій методу розпізнавання.

Дослідження в даній магістерській роботі дали змогу зробити наступні висновки:

- Існує велика кількість методів розпізнавання людського тіла, але найпоширенішими та найуспішнишими є методи з використанням нейроних мереж.

- Згорткові нейронні мережі є сильним інструментом розпізнавання оскільки має можливість розпізнавати велику кількість об'єктів в залежності від тих на яких вона навчиться, а також гнучка ахітектура обробки зображень та пошуку рис і характеристик на них дає можливість чітко визначати потрібні образи.

- Неділоком методу розпізнавання є робота із зображенями поганої якості та деталізації, саме тому додавання кроку попередньої обробки дає можливість покращити дані для пошуку потрібних характеристик образу.

- Модифікація попередньої обробки дала можливість знаходити на багатьох зображенях поганої якості частини людського тіла, які до цього не були знайдені, а також підвищити оцінку впевненості нейронної мережі в коректності розпізнавання.

Таким чином за результатами магістерської роботи можна зробити висновок, що модифікований метод розпізнавання частин тіла людини дає можливість краще знаходити потрібні образи на зображеннях поганої якості та деталізації.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. CNN Acchitectures [Електронний ресурс]. – 2018. – Режим доступу: https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848

2. Review - VGGNet [Електронний ресурс]. – 2018. – Режим доступу: https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvlc-2014-image-classification-d02355543a11

3. Combining Local Appearance and Holistic View: Dual-Source Deep Neural Networks for Human Pose Estimation [Електронний ресурс]. – 2016. – Режим доступу: https://arxiv.org/pdf/1504.07159.pdf

4. Self Adversarial Training for Human Pose Estimation [Електронний ресурс]. – 2015. – Режим доступу: https://arxiv.org/pdf/1707.02439.pdf

5. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields [Електронний ресурс]. – 2017. – Режим доступу:http://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf

6. VGG19 [Електронний ресурс]. – 2011. –Режим доступу: https://www.researchgate.net/figure/Example-network-architectures-for-ImageNet-Left-the-VGG-19-model-196-billion-FLOPs_fig3_309392322

7. РОЗПІЗНАВАННЯ ОБРАЗІВ ТА ЇХ КЛАСИФІКАЦІЯ [Електронний ресурс]. – 2017. – Режим доступу: https://studopedia.com.ua/1_42779_zagalna-harakteristika-zadach-rozpiznavannya-obraziv-ta-matematichna-model-zadachi.html

8. Практичне застосування нейронних мереж [Електронний ресурс]. – 2017. – Режим доступу: https://habr.com/post/322392/

9. Згорткові нейронні мержі [Електронний ресурс]. – 2017. – Режим доступу: http://ru.datasides.com/code/cnn-convolutional-neural-networks/

10. Кольорова модель LAB [Електронний ресурс]. – 2017. – Режим доступу: https://uk.wikipedia.org/wiki/Lab