


Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА


на тему:

Інтелектуальна система автоматизованого детектування пози оператора

Виконав: студент 2 курсу, групи 2АКІТ-22м
спеціальності 151 - Автоматизація та
комп'ютерно-інтегровані технології


 Дмитро ЧЕРНАТА
Ім'я ПРІЗВИЩЕ

Керівник: к.т.н., доцент, доцент кафедри КСУ
ступінь, звання, посада

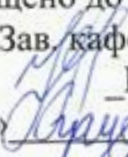
 Тетяна ГРИЦУК
Ім'я ПРІЗВИЩЕ

« 01 » 12 2023 р.

Опонент: к.т.н., доцент, доцент кафедри АІТ
ступінь, звання, посада

 Ілона БОГАЧ
Ім'я ПРІЗВИЩЕ

« 05 » 12 2023 р.

Допущено до захисту
Т.в.о.Зав. кафедри КСУ
 Марія ЮХИМЧУК
« 07 » Квітень 2023

Вінницький національний технічний університет
 Факультет інтелектуальних інформаційних технологій та автоматизації
 Кафедра комп'ютерних систем управління
 Рівень вищої освіти другий (магістерський)
 Галузь знань - 15 - Автоматизація та приладобудування
 Спеціальність - 151 - Автоматизація та комп'ютерно-інтегровані технології
 Освітньо - професійна програма - Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
 Т.в.о.Зав. кафедри КСУ

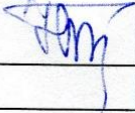
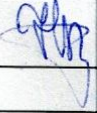
Марія ЮХИМЧУК

“09” жовтня 2023 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ студенту Черната Дмитро Вадимович.

(прізвище, ім'я, по батькові)

1. Тема роботи. Інтелектуальна система автоматизованого детектування пози оператора керівник роботи Грищук Тетяна Вікторівна затверджені наказом ВНТУ від “18” вересня 2023 року №247
2. Термін подання студентом роботи “1” грудня 2023 року
3. Вихідні дані до роботи: координати x і y кожної ключової точки, рейтинг надійності кожної ключової точки, головний клас (main), що забезпечує роботу системи, класи для попередньої обробки даних, нейронної мережі та масштабування, основний програмний модуль (mainprogram), який керує всіма компонентами системи, відеодані, зображень для оцінки пози людини, Poselet conditioned pictorial structures / L. Pishchulin [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2013
4. Зміст текстової частини огляд методів розпізнавання пози людини, опис розробки системи моделювання процесів визначення пози людини на відеофрагменті, постановка цілей та задач дослідження, обґрунтування використання згорткової нейронної мережі до задачі розпізнавання пози людини, математичний опис роботи нейронної мережі, опис шарів згорткової нейронної мережі, обґрунтування базової моделі нейромережі, опис інструментів розробки, опис системи розпізнавання пози людини
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень) схема роботи алгоритму системи розпізнавання пози людини, архітектура системи розпізнавання пози людини, візуалізація компонентів карт інтенсивності для лівого плеча, візуалізація полів асоціації частин(ліве плече - ліве стегно), візуалізація розробленої системи та порівняння із OpenPose на наборі даних nuScenes, візуалізація розробленої системи та порівняння із Mask RCNN на наборі даних nuScenes

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	викона прийм
1-4	Ковтун В.В. д.т.н., професор, зав. кафедри КСУ		
5	Буреннікова Н. В., д.е.н., професор кафедри ЕПВМ.		

1. Дата видачі завдання “09” жовтня 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		П
		початок	закінчення	
1	Аналіз методів розпізнавання пози людини	10.09.2023	11.09.2023	
2	Постановка цілей та задач дослідження	21.09.2023	25.09.2023	
3	Математичний опис роботи нейронної	10.10.2023	13.10.2023	
4	Розробка програмного забезпечення	18.10.2023	20.10.2023	
5	Розробка схем нейронної мережі	29.10.2023	01.11.2023	
6	Розробка та тестування програмного забезпечення	19.11.2023	20.11.2023	
7	Розрахунок економічної частини	21.11.2023	22.11.2023	
8	Графічні матеріали	25.11.2023	27.11.2023	

Студент
(підпис)



Дмитро Ч
(Ім'я ПІ)

Керівник роботи
(підпис)



Тетяна Г
(Ім'я ПІ)

АНОТАЦІЯ

УДК 004.942

Черната Д. В. Інтелектуальна система автоматизованого детектування пози оператора. Магістерська кваліфікаційна робота зі спеціальності 151 - Автоматизація та комп'ютерно-інтегровані технології, освітня програма - Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2023. 134 с.

На укр. мові. Бібліогр.: 52 назв; рис.: 31; табл. 9.

У магістерській кваліфікаційній роботі розроблено систему, яка визначає пози людини у відеопотоках, шляхом аналізу картини та вивчення ефективності цієї системи порівняно з існуючими методами. В результаті роботи проведено аналіз, розроблено алгоритм для розпізнавання поз у відео в реальному часі та оцінено його ефективність порівняно з існуючими методами. Розроблені алгоритми дозволяють підвищити точність та швидкість розпізнавання поз у відеопотоках. У практичній частині обгрунтовано інструментальні засоби реалізації системи, розроблено програмне забезпечення, наведено результати його тестування. У економічній частині проаналізований технічний рівень і розрахована собівартість реалізації розробки.

Ключові слова: автоматизована система, розпізнавання пози людини, нейронна мережа, ефективність, точність, оцінка пози.

ABSTRACT

UDC 004.942

Chernata D.V. Intelligent system of automated detection of the operator's posture. Master's thesis on specialty 151 - Automation and computer-integrated technologies, educational program - Intelligent computer systems. Vinnytsia: VNTU, 2023. 134 p.

In Ukrainian speech Bibliography: 52 titles; Fig.: 31; table 9.

In the master's thesis, a system was developed that determines human postures in video streams by analyzing the picture and studying the effectiveness of this system compared to existing methods. As a result of the work, an analysis was carried out, an algorithm was developed for recognizing poses in videos in real time, and its effectiveness compared to existing methods was evaluated. The developed algorithms allow to increase the accuracy and speed of recognition of poses in video streams. In the practical part, the instrumental means of implementing the system are substantiated, the software is developed, and the results of its testing are given. In the economic part, the technical level is analyzed and the cost of development implementation is calculated.

Keywords: automated system, human posture recognition, neural network, efficiency, accuracy, posture assessment.

ЗМІСТ

ВСТУП	8
Практична цінність роботи полягає у розробці програмного додатку для розпізнавання пози людини.	9
Апробація. . Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)»:	9
Публікації: Черната Д. В. «ІНТЕЛЕКТУАЛЬНА СИСТЕМА АВТОМАТИЗОВАНОГО ДЕТЕКТУВАННЯ ПОЗИ ОПЕРАТОРА», «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2023. URL https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19581/16328	10
1 СУЧАСНИЙ СТАН РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ	11
1.1 Огляд методів розпізнавання пози людини	11
1.2. Огляд існуючих рішень для реалізації проекту.....	19
1.3 Опис розробки системи моделювання процесів визначення пози людини на відеофрагменті.....	20
1.4 Постановка цілей та задач дослідження	21
2 ЗАСТОСУВАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДО ЗАДАЧ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ	23
2.1 Обґрунтування використання згорткової нейронної мережі до задачі розпізнавання пози людини	23
2.2 Математичний опис роботи нейронної мережі	24
2.2.1 Опис шарів згорткової нейронної мережі.....	25
2.2.2 Навчання згорткової мережі.....	32
2.3 Удосконалення методу розв’язання задачі розпізнавання пози людини.....	36
3 ОСОБЛИВОСТІ ПОБУДОВИ МОДЕЛІ ДЛЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ	43
3.1 Обґрунтування базової моделі нейромережі.....	43
3.2 Опис інструментів розробки	54
4. СИСТЕМИ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ	62
4.1 Опис функціональності системи.....	62
4.2 Варіанти використання системи	62
4.3 Етапи роботи системи розпізнавання пози людини	64
4.4 Архітектура згорткової нейронної мережі для вирішення задачі розпізнавання пози людини	67
4.5 Експерименти та оцінка роботи системи.....	76
5 ЕКОНОМІЧНА ЧАСТИНА	84
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки.....	84
5.2 Розрахунок узагальненого коефіцієнта якості розробки.....	88

5.3 Розрахунок витрат на проведення науково-дослідної роботи	89
5.3.1 Витрати на оплату праці	89
5.3.2 Відрахування на соціальні заходи	92
5.3.3 Сировина та матеріали	92
5.3.4 Розрахунок витрат на комплектуючі	94
5.3.5 Спецустаткування для наукових (експериментальних) робіт	94
5.3.6 Програмне забезпечення для наукових (експериментальних) робіт.....	95
5.3.7 Амортизація обладнання, програмних засобів та приміщень	96
5.3.8 Паливо та енергія для науково-виробничих цілей.....	97
5.3.9 Службові відрядження.....	98
5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації ...	98
5.3.11 Інші витрати.....	99
5.3.12 Накладні (загальновиробничі) витрати.....	99
5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	100
ВИСНОВКИ.....	106
ПЕРЕЛІК ПОСИЛАНЬ	107
ДОДАТКИ.....	112
Додаток А	113
Додаток Б	114

ВСТУП

Стрімкий розвиток сучасних комп'ютерних технологій призвів до появи потужних і доступних обчислювальних пристроїв, які дозволяють реалізовувати та виконувати складні алгоритми, що вимагають великих обсягів обчислювальної потужності. В результаті стає можливою практична реалізація різних методів штучного інтелекту, в тому числі ефективних методів машинного навчання, здатних замінити людину в різних сферах діяльності, що потребують тривалої монотонної обробки інформації.

З кожним роком збільшується кількість автономних технологій, технологій віртуальної та доповненої реальності, які використовують інформацію про положення тіла людини в просторі. Здатність визначити позу людини в певній області зображення або відео відіграє ключову роль. Хоча багато великих компаній IT-індустрії, такі як Google, Facebook і NVidia, активно проводять фундаментальні та прикладні дослідження в області машинного навчання, поки що немає готового загального рішення подібних проблем.

Значного прогресу досягнуто в розпізнаванні жестів людини завдяки використанню нейронних мереж. Однією з найпоширеніших моделей є перцептрон. Однак ця модель не підходить для вирішення проблеми розпізнавання жестів людини, оскільки великий розмір вхідних даних (зображень, відео) призводить до значного збільшення кількості синаптичних зв'язків, нейронів у мережі. В результаті значно зростає швидкість і обчислювальна складність розпізнавання. Ця модель нейронної мережі ігнорує топологію вхідних даних і не враховує унікальну двовимірну структуру вхідного зображення. Для усунення цих недоліків скористаємося згортковими нейронними мережами.

Існуючі моделі та методи розпізнавання поз відео є досить складними та трудомісткими для оцінки пози в реальному часі. Зараз активно ведеться розробка спрощених алгоритмів, які зможуть дуже точно та швидко розпізнавати жести людини у відеопотоках, що дозволить використовувати їх у безпілотних

автомобілях у реальному часі. Ця робота зосереджена на вдосконаленні існуючих алгоритмів для оцінки поз людини в переповнених неінсценованих сценах.

Метою дослідження є покращення виявлення та розуміння людської поведінки бортовими комп'ютерами шляхом удосконалення методів визначення поз людини в реальному часі. Для досягнення цієї мети необхідно виконати наступні завдання:

- проаналізувати існуючі алгоритмічні та програмні аналоги в предметній області;
- реалізувати алгоритми попередньої обробки вхідних даних для подачі в нейронні мережі;
- розробити систему розпізнавання жестів людини у відеопотоках у реальному часі;
- вивчити ефективність розробленого алгоритму та порівняти його з існуючими аналогічними алгоритмами на вибраних наборах даних.

Об'єктом дослідження є процес розпізнавання жестів людини в режимі реального часу з потокового відео.

Предметом дослідження є математичні моделі та методи розпізнавання поз людини на зображенні.

Новизна отриманих результатів:

- удосконалено метод розпізнавання поз людини, який, на відміну від існуючих, використовує карти інтенсивності та поля інтенсивності частин тіла для підвищення точності в режимі низької роздільної здатності та швидкості розпізнавання в реальному часі.

Практична цінність роботи полягає у розробці програмного додатку для розпізнавання поз людини.

Апробація. . Представлені в роботі результати апробовані в результаті участі в конференції Всеукраїнська науково-практична Інтернет-конференція студентів, аспірантів та молодих науковців «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ (МН-2024)»:

Публікації: Черната Д. В. «ІНТЕЛЕКТУАЛЬНА СИСТЕМА АВТОМАТИЗОВАНОГО ДЕТЕКТУВАННЯ ПОЗИ ОПЕРАТОРА», «МОЛОДЬ В НАУЦІ: ДОСЛІДЖЕННЯ, ПРОБЛЕМИ, ПЕРСПЕКТИВИ», 2023. URL <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19581/16328>

1 СУЧАСНИЙ СТАН РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

1.1 Огляд методів розпізнавання пози людини

Один із варіантів вирішення задачі розпізнавання пози людини був запропонований ще в 1973 році - моделі структури зображення [1]. Фреймворк базується на створенні математичної моделі та графічної форми людського скелета, де кістки є ребрами даного графа, а суглоби є вершинами.

В останні роки найсучасніші методи оцінки пози базуються на згорткових нейронних мережах. Вони перевершують традиційні методи, засновані на структурах зображень і деформуючих моделях частин. Популярність глибокого навчання почалася з DeepPose [2], який використовує каскадні згорткові мережі для оцінки пози всього тіла. Потім, замість прогнозування абсолютних положень людських суглобів, деякі автори використовують залежності між вхідними та вихідними просторами для покращення оцінки пози шляхом прогнозування виправлення помилок для кожної ітерації або використання мереж уточнення пози людини.

Усі ці методи оцінки пози людини можна розділити на методи «знизу вгору» та «зверху вниз». Почніть з оцінки кожного суглоба тіла, а потім згрупуйте їх, щоб створити унікальні пози. Останній спочатку запускає детектор тіла та оцінює суглоби тіла в межах визначених меж. Підхід зверху вниз особливо ефективний, коли пішоходи накладаються на інших пішоходів у обмеженому діапазоні зіткнень. Попередні методи знизу вгору не обмежують домен, але все ще містять приблизні карти функцій для локалізації. Цей метод не містить жодних обмежень локалізації швів на основі сітки та здатний оцінювати кілька позицій, що перекриваються.

Розглянемо докладніше існуючі методи розпізнавання людських жестів.

Відкрита поза

OpenPose [3] є одним із найпопулярніших методів знизу вгору для оцінки пози багатьох людей у кадрі. Як і більшість методів «знизу вгору», OpenPose

спочатку визначає частини (ключові точки), що належать кожній людині на зображенні, а потім призначає ці частини окремим особам. На рисунку 1.1 показана архітектура моделі OpenPose.

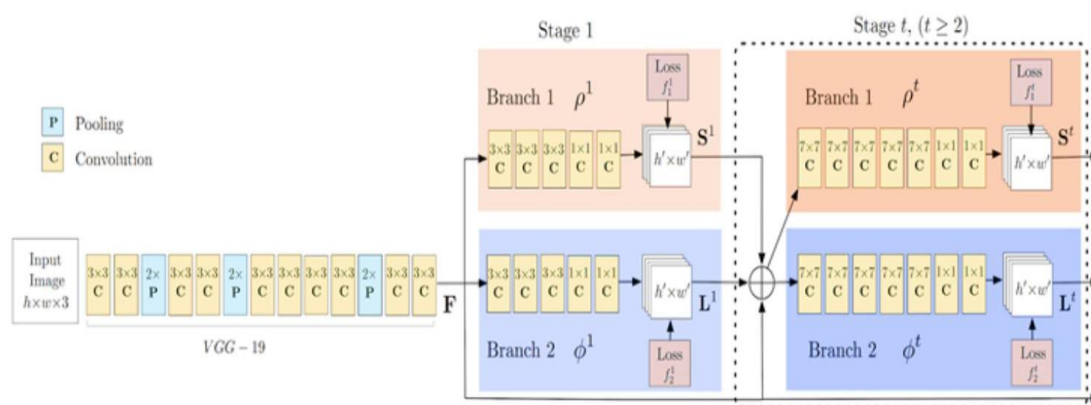


Рисунок 1.1 - Архітектура моделі OpenPose [3]

Мережа OpenPose спочатку витягує функції із зображення за допомогою кількох перших шарів (VGG-19 на блок-схемі вище). Потім ці об'єкти подаються у дві паралельні гілки згорткового шару. Перша гілка передбачає набір з 18 карт довіри, кожна з яких представляє певну частину людського скелета. Друга гілка надає набір із 38 полів асоціації частин, які представляють ступінь асоціації між частинами.

Послідовні етапи використовуються для уточнення прогнозів, зроблених кожним шаром. За допомогою карт довіри частин між двома частинами формується дводольний граф. Слабші ланки в дводольному графі скорочуються з використанням значень полів спорідненості для кожної частини. За допомогою наведених вище кроків людський скелет можна оцінити та призначити кожній людині на зображенні.

Глибокий

DeepCut [4] - метод оцінки пози людини знизу вгору. У своїй роботі автори визначили такі завдання:

1. Підготуйте групу кандидатів з частин тіла D . Цей набір представляє всі можливі розташування частин тіла кожної людини на зображенні. Виберіть підмножину частин тіла з наведеного вище набору частин тіла-кандидатів.

2. Позначте кожну вибрану частину тіла C одним із класів частин тіла. Клас частини тіла представляє тип частини тіла, наприклад «рука», «нога», «тулуб» тощо.

3. Відокремлення частин тіла, що належать людині.

Зазначену вище задачу розв'язують шляхом її моделювання як задачі цілочисельного лінійного програмування (ILP). Система моделюється шляхом розгляду трійки (x, y, z) бінарних випадкових величин, область визначення якої вказана в наступному рівнянні [4].

$$\begin{aligned} x &\in \{0,1\}^{D * C} \\ y &\in \{0,1\}^{\binom{D}{2}} \\ z &\in \{0,1\}^{\binom{D}{2} * C^2} \end{aligned} \quad (1.1)$$

Розглянемо дві частини тіла-кандидати d і d' з набору частин тіла-кандидатів D і класи c і c' з набору класів C . Кандидатів на частини тіла можна знайти через Faster RCNN або Dense CNN. Тепер ми можемо сформулювати наступний набір тверджень.

Якщо $x(d, c) = 1$, це означає, що частина тіла кандидата d належить до класу c .

Крім того, $y(d, d') = 1$ означає, що частини тіла кандидата d і d' належать одній людині.

Вони також визначають $z(d, d', c, c') = x(d, c) * x(d', c') * y(d, d')$. Якщо вказане значення рівняння дорівнює 1, це означає, що частина тіла кандидата d належить до категорії c , частина тіла кандидата d' належить до категорії c' і, нарешті, частини тіла кандидата d і d' належать одній особі.

Останнє речення можна використовувати для розрізнення пози тіла, що належить різним людям. Очевидно, що наведене вище твердження можна виразити як функцію (x, y, z) за допомогою лінійного рівняння. Таким чином

встановлюється повна лінійна програма, за допомогою якої можна оцінити пози кількох людей.

Ієрархічна RNN для каркасного виявлення дій

У роботі [5] автори розділили людський скелет на п'ять анатомічних частин (тулуб, дві руки та дві ноги) замість безпосереднього введення окремих кісток у нейронну мережу. У кожному з п'яти розділів виділяється кілька важливих точок і відстежується їх рух. Кожна з цих п'яти частин подається в двонаправлену нейронну мережу (Bidirectional Recurrent Neural Network - BRNN).

У міру збільшення кількості шарів зображення, вибрані підмережею, об'єднуються ієрархічно на вході наступних шарів. Нарешті, повністю підключені шари та шари softmax завершують архітектуру, генеруючи остаточне представлення для класифікації руху. На малюнку 1.2 показано архітектуру цієї ієрархічної нейронної мережі та описано частини тіла, відповідальні за кожен рівень.

Точність цього методу досить висока, але він має суттєвий недолік: в якості вхідних даних використовуються дані захоплення руху, що вимагає спеціального обладнання для отримання інформації про рух конкретної людини.

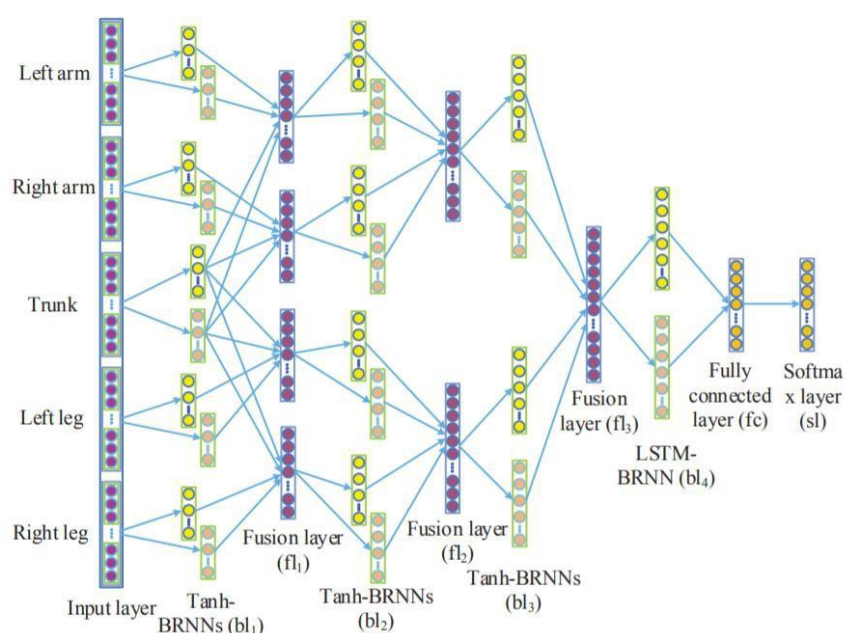


Рисунок 1.2 - Архітектура ієрархічної нейронної мережі [5]

Регіональна оцінка пози кількох людей (RMPE або AlphaPose)

RMPE [6] є популярним методом зверху вниз для оцінки пози людини. Автори стверджують, що низхідні методи часто залежать від точності детекторів людини, оскільки оцінка місця розташування виконується в регіоні, де знаходиться тіло людини. Таким чином, помилки локалізації та очікуване дублювання граничних полів можуть призвести до створення неоптимальних алгоритмів розпізнавання пози людини.

Щоб вирішити цю проблему, автори запропонували використовувати мережу симетричного просторового перетворення (SSTN) [6] для виділення окремих особин із неточних граничних полів. У цій області використовується оцінювач пози однієї людини (SPPE) для оцінки скелета людської пози людини. Мережа просторової детрансформації (SDTN) використовується для перетворення оціненої пози людини в вихідну систему координат зображення. Нарешті, технологія параметричного немаксимального придушення (NMS) використовується для вирішення проблеми надмірної генерації пози.

Крім того, автор також реалізував генератор керованих пропозицій пози, щоб розширити навчальні зразки для кращого навчання мереж SPPE та SSTN. Важливою особливістю RMPE є те, що цей метод можна розширити до будь-якої комбінації алгоритмів виявлення людини та SPPE.

R-CNN (регіон із CNN)

Архітектура мережі R-CNN [7] була розроблена командою з Каліфорнійського університету в Берклі для застосування згорткових нейронних мереж до завдань виявлення об'єктів. Процес виявлення цілі мережею R-CNN можна розділити на наступні етапи:

- використовуйте вибіркового пошук для вибору областей-кандидатів;
- перетворити регіон до розміру, прийнятого CNN CaffeNet;
- використовуйте CNN для отримання 4096-вимірного вектора ознак;
- виконайте N бінарних класифікацій для кожного вектора ознак, використовуючи N лінійних SVM;

- лінійна регресія параметрів кадру області для більш точного покриття об'єкта.

R-CNN в основному функціонує як класифікатор, він не передбачає межі об'єктів (за винятком уточнення за допомогою регресії обмежувальної рамки). Його точність залежить від продуктивності модуля регіонального котирування. У деяких роботах були запропоновані методи використання глибоких мереж для прогнозування обмежувальних рамок об'єктів.

Fast R-CNN [8] пропонує прискорити процес R-CNN шляхом внесення деяких модифікацій: замість проходження 2000 регіонів-кандидатів окремо, все зображення передається через CNN. Потім запропоновані регіони накладаються на згенеровану загальну карту об'єктів. Функції з різних регіонів перетворюються на фіксовані розміри за допомогою процесу RoIPooling. Вікно області шириною w і висотою h поділено на сітку з клітинками HW розміром h/H w/W . Для кожного такого блоку виконується максимальне об'єднання, щоб вибрати лише одне значення, що дає матрицю характеристик обладнання.

Після вдосконалення Fast R-CNN вузьке місце нейронної мережі стає механізмом створення регіону-кандидата. У 2015 році команда Microsoft Research значно пришвидшила цей етап. Вони пропонують обчислювати регіони не за початковим зображенням, а за картами ознак, отриманими від CNN (рис. 1.3).

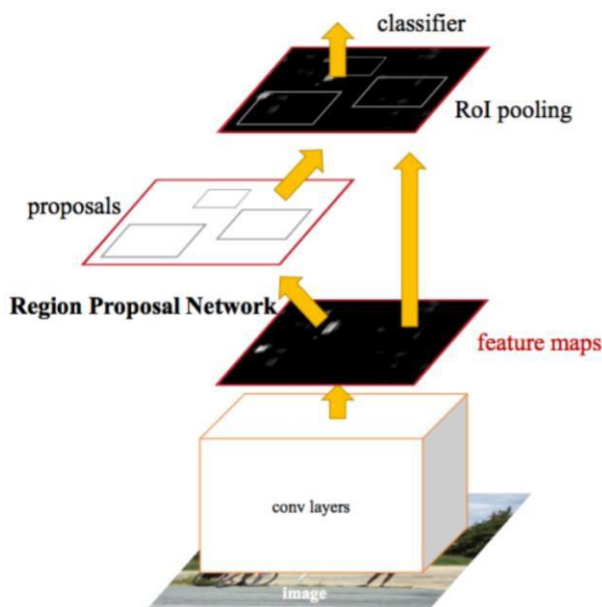


Рисунок 1.3 - Архітектура Faster R-CNN [8]

Для цього додано модуль регіональної мережі пропозицій (RPN). У рамках RPN витягнуті ознаки CNN передаються через «міні-нейронну мережу» з вікном 3x3. Отримані з його допомогою значення передаються на два паралельних повнозв'язаних шару: шар регресії (reg) і шар класифікації (cls).

Маска R-CNN

Mask R-CNN [7] продовжує архітектуру Faster R-CNN, додавши ще одну гілку, яка передбачає розташування маски, що покриває знайдений об'єкт, і вирішила завдання сегментації екземпляра (рис. 1.4).

Маска - це просто прямокутна матриця, у якій 1 у певній позиції вказує на те, що відповідний піксель належить до об'єкта даної категорії, а 0 вказує на те, що піксель не належить цьому об'єкту. Виділення маски здійснюється незалежно від класу: передбачте маску окремо для кожного класу без попереднього знання того, що представлено в діапазоні, і просто виберіть маску для класу, який виграє незалежний класифікатор. Дехто вважає такий підхід більш ефективним.

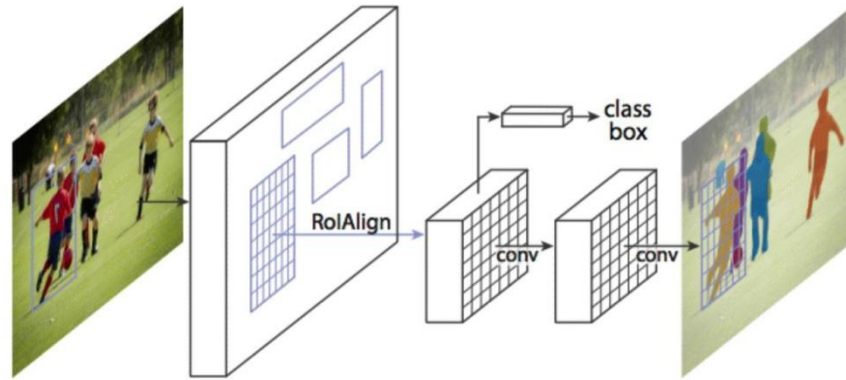


Рисунок 1.4 - Архітектура Mask R-CNN [7]

Однією з головних модифікацій, що впливають із необхідності прогнозування масок, є зміна процедури RoIPool (обчислення матриці характеристик регіонів-кандидатів) на так звану RoIAlign. Справа в тому, що карта функцій, отримана від CNN, менша за оригінальне зображення, і область на зображенні, яка охоплює цілу кількість пікселів, не може бути зіставлена з пропорційною областю карти з цілою кількістю функцій.

RoIPool вирішує цю проблему, просто округляючи невеликі значення до цілих чисел. Цей метод добре працює для вибору обмежувальних рамок, але маски, обчислені на основі таких даних, надто неточні. Навпаки, RoIAlign не використовує округлення, зберігає всі числа дійсними та використовує білінійну інтерполяцію до найближчих чотирьох цілих точок для обчислення власних значень.

У той же час алгоритми виявлення об'єктів можна навчити визначати місцезнаходження людей. Поєднуючи інформацію про положення людини та набір ключових точок, ми отримуємо скелет людської пози кожної людини на зображенні.

Цей підхід подібний до підходу зверху вниз, але етап виявлення людини виконується паралельно з етапом виявлення частини тіла. Іншими словами, етап виявлення ключової точки та етап виявлення особи незалежні один від одного.

1.2. Огляд існуючих рішень для реалізації проекту

Відеоролики використовуються як вхідні дані для системи, але перед тим, як ці дані будуть подані в нейронну мережу, їх необхідно обробити, тобто витягується лише та інформація, яка потрібна для вирішення поставленого завдання. У нашому випадку ця інформація являє собою координати тіла людини в двовимірному просторі - його кістки.

Щоб отримати рухомий скелет на зображенні, спочатку потрібно вибрати об'єкт, для якого існують відповідні алгоритми.

Існує два основних типи методів виділення рухомих об'єктів [9]:

- метод виділення границі;
- спосіб виділення всієї області об'єкта.

Основна ідея методу першого типу полягає в знаходженні різниці векторів оптичного потоку і подальшому побудові контуру рухомого об'єкта на зображенні. На відміну від цього, друга категорія методів базується на групуванні подібних векторів оптичного потоку в певній області та поданні їх як вихідних даних алгоритму як рухомих об'єктів.

Оптичний потік - це візуальне рухоме зображення, яке представляє зміщення кожної точки між двома зображеннями. Існує багато алгоритмів побудови оптичного потоку, деякі з яких представлені нижче.

Метод Лукаса-Канаді [10] є локальним методом, який обчислює оптичний потік в кожному пікселі без урахування значень оптичного потоку в інших пікселях. Це диференціальний метод, оскільки він використовує похідні для оцінки. Це досить швидкий метод, але через локальність він погано працює на ділянках з однорідною текстурою.

Алгоритм Хорна-Шанко [11] більш глобальний, ніж метод Лукаса-Канаді. Він базується на припущенні, що оптичний потік досить плавний по всьому зображенню. Алгоритм заснований на припущенні, що зміна проекції вектора оптичного потоку обмежена відповідно до визначеного рівняння. Існують також різні варіанти модифікації цих методів, які можуть підвищити їх ефективність і

зменшити кількість помилок, наприклад, поєднання методу Лукаса-Канаді з методом Хорна-Шанко [12].

Для фіксованих фонів у відеозаписах можна використовувати простіші методи - бінаризацію або віднімання фону [13]. де рух визначається пікселями, інтенсивність яких змінюється відносно інтенсивності фонових пікселів. Алгоритм бінаризації відстеження руху простий у реалізації та швидкий у обчисленні.

Скелет зображення підкреслює геометричні та топологічні властивості форми, такі як її з'єднання, топологія, довжина, орієнтація, ширина. Разом із відстанню його точок до меж фігури, скелети також можуть бути використані як представлення форми (вони містять всю інформацію, необхідну для відновлення форми) [14].

Скелети широко використовуються в комп'ютерному зорі, аналізі зображень, розпізнаванні образів і цифровій обробці зображень для таких цілей, як оптичне розпізнавання символів, розпізнавання відбитків пальців і візуальний контроль. Існує багато методів скелетування, серед яких найпростішим і найменш ефективним є шаблонний метод [15]. Він передбачає аналіз меж подвійних елементів зображення для послідовного виявлення та видалення екстремальних елементів об'єктів. Іншим методом скелетування є хвильовий [16]. Хвильові алгоритми часто використовуються в комп'ютерних іграх для визначення мінімальної відстані від одного об'єкта до іншого в обмеженому дискретному просторі. Для цього в початковій точці генерується хвиля, яка поширюється за певними правилами і позначає точки, через які вона проходить. Одним із найефективніших алгоритмів на даний момент є алгоритм Цзуна-Суна, заснований на діаграмах Вороного.

1.3 Опис розробки системи моделювання процесів визначення пози людини на відеофрагменті

Завдання моделювання процесу визначення положення об'єкта на динамічному наборі зображень обширне і має багато нюансів, які розробники повинні вивчити і узгодити з клієнтом. В результаті досліджень подібних систем та досліджень предметної області було вирішено створити систему, основною функцією якої є розпізнавання положення об'єктів на зображенні, ідентифікація точок скелета, формування відображення пози людини в режимі реального часу. у відео.

Для виконання поставлених завдань вибрано набір засобів реалізації та спроектовано архітектуру системи та її компонентів.

1.4 Постановка цілей та задач дослідження

Розроблено для розпізнавання 2D-поз людей у відеопотоках у реальному часі.

Основними цілями створення комплекту завдань є:

- підвищення швидкості і точності розпізнавання пози людини;
- підвищення ефективності аналізу відеопотоку бортовими комп'ютерами та камерами відеоспостереження безпілотних автомобілів для подальшої обробки та прийняття рішень.

Для досягнення поставлених цілей необхідно вирішити наступні завдання:

- проаналізувати існуючі алгоритмічні та програмні аналоги в предметній області;
- реалізувати алгоритми попередньої обробки вхідних даних для подачі в нейронні мережі;
- розробити систему розпізнавання жестів людини у відеопотоках у реальному часі;
- вивчити ефективність розробленого алгоритму та порівняти його з існуючими аналогічними алгоритмами на вибраних наборах даних.

Огляд існуючих систем показує, що задача автоматичного розпізнавання пози людини залишається актуальною і сьогодні. Після аналізу існуючих рішень у

цій галузі ми додатково модифікуємо підхід «знизу вгору», спочатку визначивши місця з'єднань на зображенні, а потім використовуючи нейронні мережі, щоб згрупувати конкретних людей на зображенні в пари.

2 ЗАСТОСУВАННЯ ЗГОРТКОВОЇ НЕЙРОННОЇ МЕРЕЖІ ДО ЗАДАЧ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ

2.1 Обґрунтування використання згорткової нейронної мережі до задачі розпізнавання пози людини

Найкращі результати серед сучасних методів машинного навчання досягаються за допомогою математичних моделей нейронних мереж. З розвитком апаратних компонентів багато великих компаній мають можливість навчати нейронні мережі на величезних наборах даних за короткий час. Ці інновації виділяють згорткові нейронні мережі та усувають проблему тривалого навчання мережі.

Проблема великих обсягів даних виникає, коли справа стосується обробки зображень, особливо розпізнавання об'єктів на фотографіях і відеокліпах. Наприклад, найпоширеніша база даних зображень містить мільйони картинок розміром $256*256$, записаних у форматі RGB. Навіть на надпотужних комп'ютерах звичайні нейронні мережі навчаються дуже довго і можуть страждати від проблем із перенавчанням, тобто нездатністю сприймати та класифікувати об'єкти. Згорткові нейронні мережі отримують на вхід вектори значень, які далі трансформуються через приховані шари.

Ці приховані шари складаються з пулів нейронів, де кожен нейрон з'єднаний з нейроном попереднього шару. Останній рівень нейронної мережі є вихідним. Як правило, це вектор, який може належати до певної категорії. Беручи зображення в ImageNet як приклад, у такій повністю пов'язаній архітектурі кожен нейрон матиме приблизно $256*256*3 = 196608$ ваг. Для повноцінної класифікації потрібно хоча б кілька таких нейронів.

Рішенням цієї проблеми є особлива архітектура штучних нейронних мереж - так звані згорткові нейронні мережі. Назва походить безпосередньо від функції, яку виконує ця згорткова мережа. Цей тип мережі є типом глибокого навчання, який використовує нелінійні перетворення для отримання результатів.

Згорткові нейрони мають більш інтелектуальну організацію для обробки зображень. Шари цієї нейронної мережі розташовуються в трьох вимірах, які є характеристиками зображення: висота, ширина і глибина (для зображень це червоний, зелений, синій канали).

2.2 Математичний опис роботи нейронної мережі

Уоррен МакКаллох і Уолтер Пітс запропонували модель штучного нейрона у своїй роботі з моделювання нейронної активності [17]. За основу своєї моделі автори взяли біологічні нейрони. Штучний нейрон McCulloch-Pitts має N вхідних двійкових значень x_1, \dots, x_n . Ці значення є спайками, що надходять на вхід нейрона (рис. 12). У нейроні спайк складається з ваги w_1, \dots, w_n .

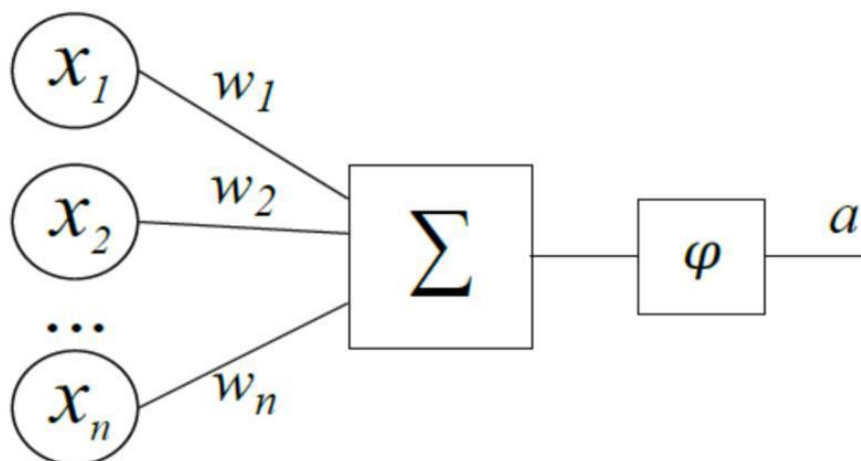


Рисунок 2.1 - Модель штучного нейрона Маккаллох-Піттса

Вихідний сигнал розраховується за такою формулою:

$$a = \varphi(\sum_{i=1}^n w_i x_i) \quad (2.1)$$

де нелінійна функція φ (функція активації) перетворює загальний сплеск у вихідне значення нейрона.

У моделі Мак-Каллоха-Піттса для цього використовується функція Хевісайда, у даній роботі буде розглянуто використання напівлінійних функцій активації (модифікованих лінійних одиниць).

Об'єднавши окремі нейрони, можна отримати штучну нейронну мережу. Для цього вихідний сигнал одного нейрона подається на вхід наступного нейрона (рис. 2.2). Нейронні мережі складаються з кількох шарів, кожен із яких може містити декілька нейронів. Рівень, який приймає сигнали із зовнішнього світу, називається вхідним. Рівень, який посилає сигнал у зовнішній світ, є вихідним шаром, а інші шари називаються прихованими.

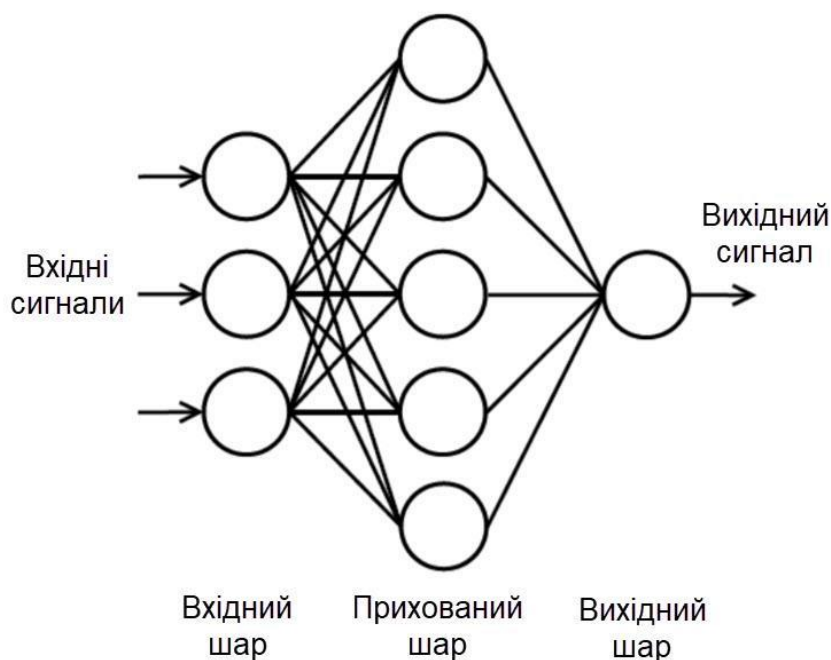


Рисунок 2.2 - Штучна нейронна мережа

Штучні нейронні мережі поділяються на мережі прямого зв'язку (в яких відсутні цикли) і рекурентні мережі (в яких допускаються цикли).

2.2.1 Опис шарів згорткової нейронної мережі

Згорткова нейронна мережа - це нейронна мережа, архітектура якої базується на чергуванні згорткових і підвибіркових шарів. Вперше це було запропоновано в

роботі [18]. Структура мережі одностороння (без зворотного зв'язку) і багатошарова.

Згортковий шар - це набір карт ознак (набір матриць). Кожна карта має ядро (так зване ядро сканування або фільтр).

Кількість таких карток змінюється залежно від завдання, наприклад, якщо взяти велику кількість карток, якість розпізнавання покращується, але збільшується обчислювальна складність. У більшості випадків рекомендується, щоб співвідношення кількості карт ознак дорівнювало один до двох [19]. Приклад організації комунікацій наведено на рисунку 2.3.

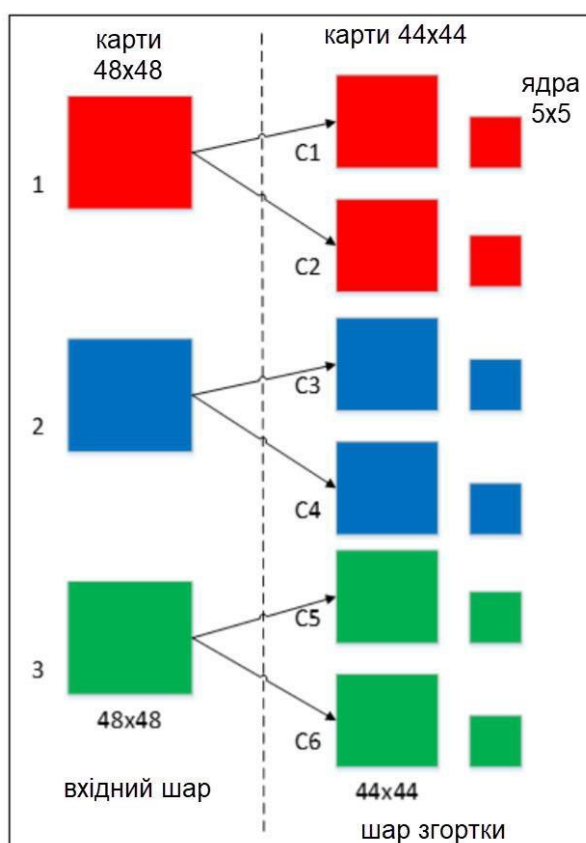


Рисунок 2.3 - Зв'язок карт ознак

Графіки згорткових шарів мають однакові розміри і розраховуються за такою формулою:

$$(w, h) = (mW - kW + 1, mH - kH + 1) \quad (2.2)$$

де (w, h) - обчислює розмір згорткового графіка;

mW - ширина попередньої карти;

mH - висота попередньої карти;

kW - ширина магнітопровода;

kH - висота сердечника.

Ядро - це фільтр або вікно, яке ковзає по всій області попередньої карти і знаходить певні характеристики об'єкта. Наприклад, якщо мережа тренується на великій кількості людей, одне з ядер може створювати максимальний сигнал в області рук або голови під час навчання, а інше ядро може виявляти інші сигнали. Розмір серцевини зазвичай береться в межах від 3×3 до 7×7 . Якщо розмір ядра малий, то воно не зможе розрізнити жодних сигналів, якщо занадто великий, то кількість зв'язків між нейронами збільшується. Крім того, при виборі розміру ядра розмір графа згорткового шару є рівномірним. Це дозволяє не втрачати інформацію під час зменшення розмірності шару підвибірки.

Ядро також є системою лусочок або синапсів, що є однією з головних особливостей згорткових нейронних мереж. У звичайній багатошаровій мережі між нейронами існує велика кількість зв'язків, які називаються синапсами, що значно уповільнює процес розпізнавання. Навпаки, у згортковій мережі загальні ваги зменшують кількість з'єднань, щоб однакові характеристики можна було знайти по всій області зображення.

Спочатку значення кожного згорткового шарового графіка дорівнює 0. Значення ваги ядра випадковим чином встановлюється в діапазоні від -0,5 до 0,5. Ядро ковзає по попередній карті та виконує операцію згортки, описану такою формулою:

$$(f * g)[m, n] = \sum_{k,1} f[m - k, n - 1] * g[k, 1] \quad (2.3)$$

де f - вихідна матриця зображення;

g - ядро згортки.

Операцію можна описати так - ми пропускаємо все зображення f через вікно розміром ядра g з певним розміром кроку (в даному випадку 1). На кожному кроці вміст вікна поелементно множиться на ядро g , а результати підсумовуються та записуються в матрицю результатів. Наочно цей процес показано на рисунку 2.4.

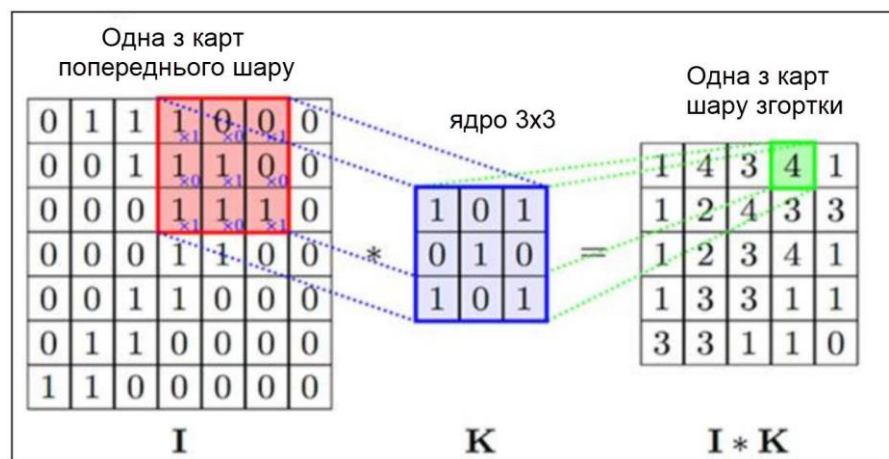


Рисунок 2.4 - Операція згортки з картою ознак

Зрештою, згортковий шар можна описати такою формулою:

$$x^l = f(x^{l-1} * k^l + b^l)x^l \quad (2.4)$$

де x^l - вихід шару l ;

$f()$ - функція активації;

b^l - l коефіцієнт зсуву шару;

$*$ - Операція згортки вхідних даних x із ядром k .

Варто відзначити, що завдяки краєвим ефектам розмір матриці зменшується і формула спрощується до наступного вигляду:

$$x_j^l = f(\sum_i x_i^{l-1} * k_j^l + b_j^l) \quad (2.5)$$

де x_j^l - карта ознак j (вихід шару l);

$f()$ - функція активації;

b_j^l - коефіцієнт зсуву l -го шару карти ознак j ;

kjl - ядро згорткового графа j , шар l ;

* - Операція згортки введення з ядром k .

Скалярний результат кожної згортки потрапляє на функцію активації, яка є деякою нелінійною функцією. Він потрібен для того, щоб додати мережі нелінійність.

Функція ReLU (Rectified Linear Unit) є однією з популярних функцій активації [20].

Він має форму такої формули:

$$f(x) = \max(0, x) \quad (2.6)$$

Графік показано на рисунку 2.5.

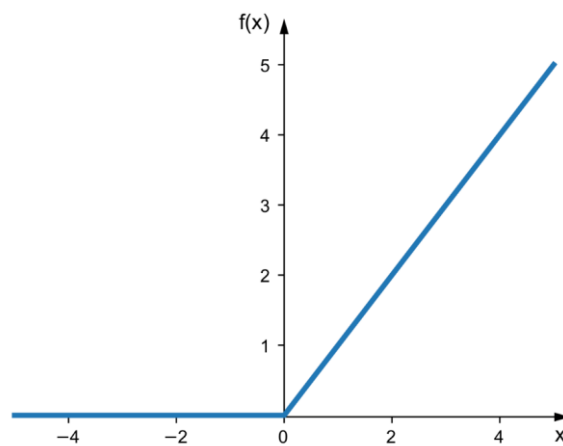


Рисунок 2.5 - Графік ReLU

Перевагами цієї функції є:

1. Реалізація ReLU з використанням порогового перетворення матриці активації до нуля, що не вимагає ресурсомістких обчислень. Крім того, ReLU не схильний до перенасичення.

2. Використання ReLU значно покращує швидкість спуску стохастичного градієнта порівняно з використанням сигмоподібного та гіперболічного тангенсу. По-перше, це пов'язано з лінійним характером функції та відсутністю насиченості.

Недоліком ReLU є те, що функція не завжди достатньо надійна. У процесі навчання вони можуть зірватися. Іноді, коли великі градієнти проходять через ReLU, це може призвести до оновлення вагових коефіцієнтів, через що певний нейрон більше не запускається. Якщо така подія станеться, градієнт через цей нейрон дорівнюватиме нулю. Іншими словами, цей нейрон буде відключений і незворотно відключений. Коли швидкість навчання висока, ви можете виявити, що велика кількість нейронів не активована. Для вирішення цієї проблеми необхідно підібрати оптимальні параметри швидкості навчання.

Функцію також було змінено, її графік показано на рисунку 2.6.

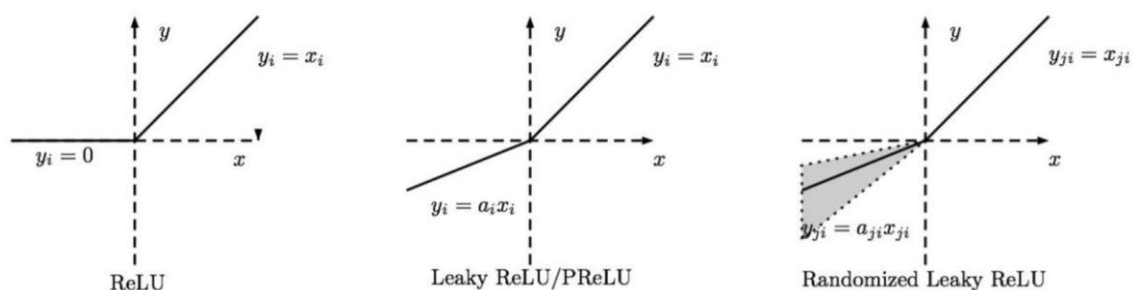


Рисунок 2.6 - Модифікації ReLU

Важливо відзначити, що ці модифікації виходять за рамки стандартного ReLU, причому параметричне ReLU (PReLU) і стохастичне ReLU (RReLU) вважаються найкращим вибором згідно з результатами досліджень [20].

Шар підвибірки (підвибірка, шар об'єднання)

Шар підвибірки має текстури, але їх кількість така ж, як у попереднього (згорнутого) шару. Призначення цього шару - зменшити розмірність карти попереднього шару. Якщо деякі особливості були виявлені під час попередньої операції згортання, то таке детальне зображення більше не потрібне для подальшої обробки та стискається в менш детальне зображення. Крім того, відфільтрування вже непотрібних деталей запобігає повторному навчанням.

Під час обробки ядра шару зі зниженою дискретизацією попереднього графіка шару ядра сканування не перетинаються, на відміну від згорткових шарів. Як правило, кожна карта має ядро 2×2 , що дозволяє зменшити попередню карту згорткового шару в 2 рази. Вся карта ознак розбита на комірки 2×2 елементи, з яких вибирається максимальне значення.

Зазвичай функція активації ReLU використовується на рівні субдискретизації. На малюнку 2.7 показано операцію підвибірки.

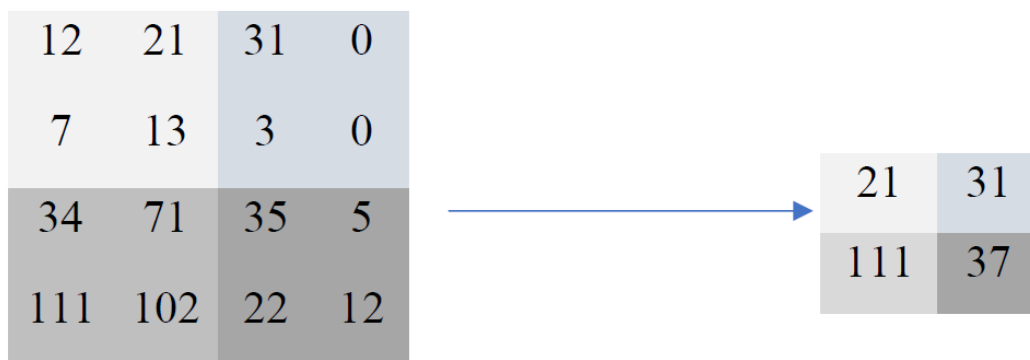


Рисунок 2.7 - Операція підвибірки, формування нового шару (2×2 Max-Pool)
Цю операцію можна описати такою формулою:

$$x^l = f(a^l * \text{subsample}(x^{l-1}) + b^l) \quad (2.7)$$

де x^l - вихід шару l ;

$f()$ - функція активації;

g^l, b^l - коефіцієнт зсуву шару l ;

$\text{subsample}()$ - операція вибірки.

Повністю зв'язаний шар

Повністю зв'язаний шар можна розглядати як звичайний шар багатошарового перцептрона (рис. 2.8). Цей рівень призначений для класифікації, він моделює складні нелінійні функції та оптимізує їх для покращення якості розпізнавання.

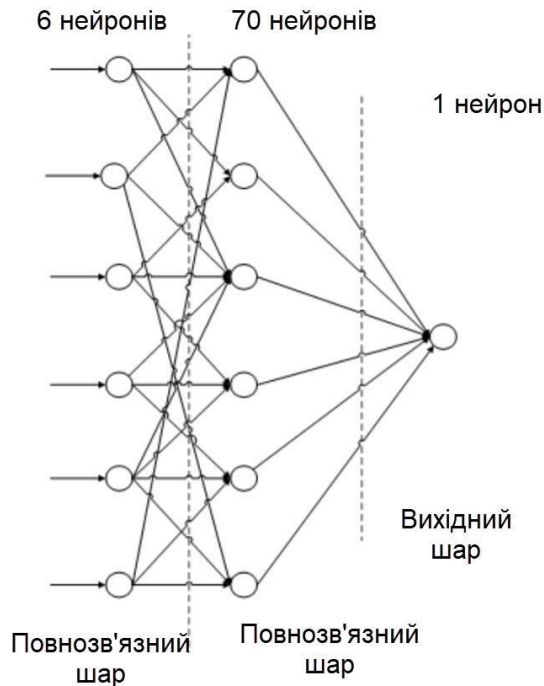


Рисунок 2.8 - Схема повнозв'язного шару

Кожен нейрон графіка попереднього шару підвибірки з'єднаний з нейроном прихованого шару. Таким чином, кількість нейронів у прихованому шарі дорівнює кількості карт у шарі підвибірки, але це не обов'язково стосується зв'язків. Наприклад, ви можете підключити лише частину нейронів будь-якого графіка в шарі зменшення дискретизації до першого нейрона прихованого шару, а решту до другого нейрона, або підключити всі нейрони першого графіка до нейронів 1 і 2 прихованих шарів. Розрахунок значень нейронів можна описати формулою:

$$x_j^l = f(\sum_i x_i^{l-1} * w_{i,j}^{l-1} + b_i^{l-1}) \quad (2.8)$$

де x_j^l - карта ознак j (вихід шару l);

$f()$ - функція активації;

b_j - 1 коефіцієнт зсуву шару;

$w_{i,j}^{l-1}$ - матриця вагових коефіцієнтів шару l .

2.2.2 Навчання згорткової мережі

На початковому етапі нейронна мережа не навчена (не налаштована). Взагалі під навчанням розуміється послідовне представлення зображень із навчального набору на вхід нейронної мережі з наступним порівнянням отриманих відповідей з бажаним виходом. У нашому випадку це 1 - зображення представляє людину, мінус 1 - зображення представляє фон або інший об'єкт (не людину), отримана різниця між очікуваною відповіддю та отриманим результатом є функцією помилки (додаткова помилка). Ця дельта помилки повинна потім поширюватися на всі підключені нейрони в мережі,

Помилка алгоритму зворотного поширення

Алгоритм зворотного поширення [21] зазвичай використовується для навчання нейронних мереж. Цей алгоритм є першим і на даний момент основним алгоритмом навчання багатошарових нейронних мереж.

Відповідно до цього алгоритму вага прихованого нейрона повинна змінюватися пропорційно помилці тих нейронів, до яких він підключений. Ось чому зворотне поширення цих помилок через мережу дозволяє правильно регулювати вагові коефіцієнти з'єднання між усіма рівнями.

У цьому випадку значення функції помилки зменшується і мережа навчається.

Значення похибки розраховується наступним чином:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - y_{pj})^2 \quad (2.9)$$

де E_p - значення функції помилки зображення p ;

t_{pj} - очікуваний результат нейрона j для зображення p ;

y_{pj} - вихід активації нейрона j для зображення p .

Неактивний стан кожного нейрона j зображення p записується як

$$S_{pj} = \sum_i w_{ij} y_{pi} \quad (2.10)$$

де S_{pj} - зважена сума виходів пов'язаних нейронів попереднього шару на вагу зв'язку, по-іншому ще позначається як неактивований стан нейрона для образу p ;

w_{ij} - вага зв'язку між нейронами i і j ;

u_{pj} - стан активації нейрона j у верхньому шарі зображення p .

Вихід кожного нейрона j є значенням функції активації f_j , яка переводить нейрон в активований стан. Будь-яка безперервно диференційована монотонна функція може бути використана як функція активації. Стан активації нейрона визначається такою формулою:

$$y_{pj} = f_j(S_{pj}) \quad (2.11)$$

де y_{pj} - стан активації нейрона j зображення p ;

f_j - функція активації;

S_{pj} - неактивний стан нейрона j зображення p .

Для мінімізації похибки найчастіше використовують градієнтний спуск і його модифікації. Суть цього методу зводиться до знаходження мінімального або максимального значення функції за рахунок руху по вектору градієнта.

Загальноприйнятим позначенням помилки нейрона є символ δ , який є:

$$\delta = \frac{\partial E}{\partial y_j} \quad (2.12)$$

Для вихідного рівня похибка обчислюється за формулою похідної середньоквадратичної похибки:

$$E_p^f = \sum_j (t_{pj} - y_{pj}) \quad (2.13)$$

Для розрахунку похибки прихованого шару використовується алгоритм зворотного розповсюдження, який полягає в обчисленні похибки прихованого

шару послідовно за значеннями похибок вихідного шару, тобто значення похибок проходять по мережі у зворотному напрямку. напрямом від виходу до входу. Розрахунок похибки прихованого шару:

$$\delta_i = \frac{\partial y_j}{\partial s_j} * \sum_j \partial_j * w_{ij} \quad (2.14)$$

де $\frac{\partial y_j}{\partial s_j}$ - похідне значення функції активації за незалежною змінною нейрона

j ;

δ_i - Помилка i -го нейрона прихованого шару;

∂_j - Помилка нейрона j у наступному шарі;

w_{ij} - вага з'єднання між нейроном i поточного (прихованого) шару та нейроном j оригінального або прихованого шару.

Алгоритм вводиться крок за кроком наступним чином:

- пряме поширення сигналів по мережі, розрахунок станів нейронів;
- розрахувати значення похибки δ вихідного шару;
- зворотне поширення: ми обчислюємо δ для всіх прихованих шарів послідовно від початку до кінця відповідно до формули;
- оновити ваги мережі на попередньо розраховану похибку δ .

Помилки обчислюються по-різному для шарів підвибірки та згорткових шарів.

Існує кілька варіантів обчислення похибок на шарі підвибірки. У першому випадку, коли рівень зменшення дискретизації знаходиться перед повністю зв'язаним шаром, він має той самий тип нейронів і з'єднань, що й повністю зв'язаний рівень, тому обчислення дельта-помилки не відрізняється від обчислення дельта для повністю зв'язаного рівня. підключений прихований шар. У другому випадку, коли рівень зменшення дискретизації знаходиться перед згортковим шаром, обчислення δ виконується шляхом зворотної згортки.

Щоб обчислити похибку на згортковому шарі, оскільки дельта шару зниження дискретизації відома, необхідно обчислити дельту поточного шару.

Насправді дельта-помилка не обчислюється, а тиражується. При прямому розповсюдженні сигналу нейрони шару понижуючої дискретизації утворюються за рахунок неперекриваючої кількості шарів, сканованих уздовж згорткового шару. При цьому вибирається нейрон із максимальним значенням. При зворотному поширенні ми повертає приріст помилки до значення Найбільший нейрон був обраний раніше, тоді як інші нейрони отримують нульові прирости помилки.

2.3 Удосконалення методу розв'язання задачі розпізнавання пози людини

Визначити позу людини на зображенні можна кількома способами. У першій роботі під поставою розуміли положення багатьох частин тіла людини, таких як литки, передпліччя, тулуб, голова тощо. Положення кожної частини тіла описується прямокутником, сторони якого не можуть бути паралельні осі координат. Вирішити завдання в цьому випадку складно, оскільки розміри частин тіла на зображенні можуть істотно відрізнитися в залежності від пози людини. Тому в роботі [22] автори визначають позу як положення суглобів на зображенні, які з'єднують частини тіла людини. В даний час це твердження використовується для визначення пози.

Завдання визначення пози людини полягає в тому, щоб знайти положення набору фіксованих точок на зображенні людини. Деякі з цих точок відповідають суглобам людського тіла і утворюють його віртуальний скелет. Оцінка пози відрізняється від завдання виявлення прогнозування структурованого результату, тобто положення різних суглобів на зображенні залежать одне від одного. Ця залежність визначається фізичними розмірами тіла людини. Існує два основних методи визначення пози людини на зображенні: модель з використанням набору деформованих частин і регресія положення суглоба.

2.3.1 Модель набору деформованих деталей

Перший розглянутий метод поєднує етап позиціонування окремих суглобів скелета з вибором найбільш ймовірної конфігурації пози людини в загальній задачі

мінімізації. Унікальною особливістю цього методу є можливість оцінити правдоподібність будь-якої пози на заданому зображенні. Іншими словами, модель із колекції деформованих частин задає розподіл ймовірностей людських поз на зображенні.

Моделі з набору деформованих частин є розширенням стандартних методів позиціонування для кращого керування об'єктами, що складаються з кількох частин. Особливістю цього підходу є те, що можуть бути враховані допустимі варіації взаємного розташування частин об'єкта відносно один одного. Спочатку цей метод використовувався для виявлення об'єктів у сцені, але пізніше його використовували для визначення пози людей на зображеннях.

Об'єкт моделюється мережею Маркова з вершинами, що відповідають необхідним стикам і ребрам, встановлюючи обмеження на їх взаємне положення. У [22] автори обмежуються розглядом лише контрпотенціалів, заданих на ребрах. Щоб зробити вихід у графічну модель точним та ефективним, автори обмежилися розглядом моделей у формі дерева.

Загалом, модель, що складається з набору частин, визначає поставу людини □ як мінімум енергетичної функції, описаної двома потенціалами (факторами):

$$E(P) = \sum_{i \in V} \varphi_i(p_i, s) + \sum_{(i,j) \in E} \psi_{(i,j)}^g(p_i, p_j, s) \quad (2.15)$$

де φ_i - унарний потенціал у поєднанні з p_i ,

$\psi_{(i,j)}^g$ - Встановити опозиційний потенціал суглобів p_i та p_j на зображенні,

s - дискретний параметр розміру тіла людини.

Парний потенціал φ_i , визначений у [23], не залежить від вхідного зображення. Завдяки використанню глобального прихованого параметра розміру тіла s можна уникнути ситуації, коли деякі частини непропорційно більші за інші у виявлених позах.

Одним із недоліків набору моделей деформованих деталей, виділених у [23], є деревоподібна структура залежностей між з'єднаннями. Наприклад, положення

колін безпосередньо не пов'язане, а лише через положення суглобів тулуба. Це дає змогу алгоритму знайти суглоби двох людських ніг на зображенні однієї ноги. Для вирішення цієї проблеми в роботі [4] пропонується розширити модель тіла людини набором поселетів (англійська версія poselet), що обмежує взаємне розташування певних підмножин суглобів тіла людини. Хоча отримана графічна модель більше не є деревом, алгоритм виведення все ще працює, оскільки він дозволяє повторювати невеликий набір станів польоту.

Базова модель [23] передбачає, що на зображенні видно всі суглоби тіла людини. Це стає серйозною проблемою в ситуаціях, коли тіло людини частково видно на зображенні через накладання об'єктів, а особа частково виходить за межі діапазону зображення. У роботі [24] автори визначають, чи є суглоб накладеним зображенням іншої людини.

Ця робота [25] демонструє важливість моделювання зовнішності людини для більш точної оцінки її постави. Автори розширюють модель пози людини кольоровими гістограмами зображення та пропонують спільний метод оцінки параметрів, який може підвищити точність визначення пози.

У роботі [26] автори використовують евристичні ознаки для опису зображень. Параметри графічної моделі вивчаються за допомогою структурного методу опорних векторів [27]. У статті [28] автори показують, що вихід моделі набору деформуючих деталей можна представити у вигляді згорткової нейронної мережі. Це дає можливість тренувати особливості зображення разом із параметрами графічної моделі.

У [27] автори показали, що модель із набору деформуючих частин може не тільки знайти позу, яка мінімізує $E(P)$, але й інші мінімуми функцій, які відрізняються від принаймні одного положення суглоба раніше.

Велика кількість таких мінімумів описує велику кількість гіпотез, які використовуються для визначення оптимальних поз людини на зображеннях. Коли кількість необхідних гіпотез значно менше кількості можливих положень суглоба на зображенні, обчислювальна складність побудови таких гіпотез не більше ніж у два рази перевищує складність побудови оптимальної гіпотези для пози людини.

Можливість встановлення гіпотез щодо поз людини на зображеннях і розширення функції мінімізації дає змогу застосовувати її для визначення поз людини у відео.

2.3.2 Повернення положення суглоба

Інший спосіб визначити позу людини на зображенні - це виконати спільну регресію позиції із зображення. На відміну від моделі набору деформуючих частин, вона не дозволяє оцінити якість довільних поз на зображенні, але може враховувати положення групи суглобів.

У роботі [28] автори побудували відображення вхідного зображення на кожен спільну координату. Запропонований ними алгоритм являє собою каскад двох нейронних мереж, який постійно уточнює положення кожного суглоба на зображенні. Автори показують, що перша регресія вказує на приблизне розташування суглобів по всьому тілу, а друга регресія вказує на розташування окремих суглобів. У роботі [28] автори показали, що прогнозування теплових карт позицій суглобів може досягти кращих результатів у рамках того ж підходу.

У [29] автори розширили попередній підхід. Щоб уточнити розташування кожного суглоба, вони також пропонують використовувати теплові карти інших місць розташування суглобів. Цей метод найбільш близький до моделі з набору деформованих частин, але дозволяє неявно враховувати положення всіх з'єднань на зображенні при їх уточненні. Найскладніша ситуація для цього методу, коли на зображенні кілька людей, і зображення цих людей частково перекриваються.

2.3.3 Визначення пози людини на відео

Моделі з набору деформованих деталей дозволяють розширити діапазон рам. З цією метою вводиться модель руху, що описує зміни пози між кадрами. Найпростіший його варіант - призначити кожному суглобу незалежні режими руху:

$$E(\{P_t\}_t^T) = \sum_{t=1}^T E(P) + \sum_{i \in V} \sum_{t=1}^{T-1} \psi_i^8(p_i^{t+1}, p_i^t, s^t) \quad (2.16)$$

де $E(P)$ - модель пози людини на зображенні.

У [27] пропонується проста модель зміни пози, яка передбачає незначні зміни пози між кадрами. В якості потенційних пар були обрані квадратичні функції зміни положення суглобів між кадрами.

Пошук мінімуму функціоналу є складним завданням, оскільки відповідна графічна модель містить цикли. Через високу обчислювальну складність зробити точні висновки неможливо. Тому автори [27] використовували побудову оптимальних гіпотез поз людини на зображеннях, щоб зменшити кількість прийнятних поз і зменшити проблему визначення оптимального стану в ланцюзі Маркова.

У цій статті ми пропонуємо узагальнення функції мінімізації, яка враховує швидкість руху кожного суглоба скелета людини. Також запропоновано алгоритм знаходження локальних мінімумів через набір положень суглобів та їх швидкісних параметрів.

Поза людини в момент часу t є послідовністю середніх точок зображення $P_t \in \{P_i = 1, \dots, t | P_i t \in R^2\}$.

У даній роботі алгоритми визначення пози людини розглядаються як наступний етап обробки відеоданих після відстеження об'єкта. Отже, формулювання завдання на визначення поз у відеоряді має такий вигляд:

вихід:

- Відеоряд $I = \{I_t\}, t = 1, \dots, N$

- Траєкторія руху тіла людини $T = \{b_t\}, t = 1, \dots, N$

Вихідні дані: Поза людини в кожному кадрі $P = \{P_t\}, t = 1, \dots, N$.

Відома траєкторія руху людини обмежує область зображення, з якої можна визначити його позу.

Розпізнавання поз людини у відеороликах розглядається як задача мінімізації енергетичної функції $E(P, \beta | I, T)$, де β - прихований параметр моделі. Параметри можуть включати або приховані параметри пози людини в кадрі (такі як параметри розміру), або глобальні параметри моделі людини (модель кольору). Енергетична

функція $E(P, t|I, T)$ визначається як $p \sim (P, t|I, T) = \exp(-E(P, |I, T))$. Відтепер, для спрощення пояснення, будемо припускати залежність енергетичної функції від початкового відеокадру та положення тіла людини в кожному кадрі, тобто $E(P, t) = E(P, t|I, T)$.

Модель спостережуваних даних є узагальненням моделі пози людини у випадку відеопослідовності. З цією метою базова модель розширена припущеннями про залежність пози людини від різних кадрів. Стандартний метод узагальнення базової моделі відповідає такій енергетичній функції:

$$E(P, \theta) = \sum_{t=1}^T E_1(p^{t+1}, p^t, \theta) \quad (2.17)$$

де $E_1(p^{t+1})$ - модель пози людини в кадрі,

$E_1(p^{t+1}, p^t)$ - модель зміни пози між кадрами.

Таку модель можна розглядати як ланцюг Маркова першого порядку, де стан у кожен момент є багатовимірним значенням, що описує позу людини. Він складається з двох частин:

- Базова модель пози людини на зображенні $E_1(p^t, \theta)$

- Модель руху суглоба $E_T(p^{t+1}, p^t, \theta)$

Найпростіший спосіб визначити модель зміни пози - це припустити незалежний рух суглоба:

$$E_T(p^{t+1}, p^t, \theta) = \sum_{i=1}^K \psi_i^t(p_i^{t+1}, p_i^t, \theta) \quad (2.18)$$

Оскільки характер руху різних суглобів подібний, достатньо розглянути лише один із них. Для спрощення специфікації в цьому підрозділі індекси розглянутих з'єднань опущені. Наприклад, $P(t)$ використовується для вказівки стану суглоба, який розглядається в кадрі $I(t)$. Це розширення моделей поз людини від зображень до відеорядів також використовувалося в попередніх роботах.

Наприклад, робота [26] пропонує модель руху, яка передбачає незначні зміни пози людини між кадрами:

$$\psi^t(p^{t+1}, p^t, \theta) = \frac{1}{2s^{t2}} (p^{t+1} - p^t)^{T-1} (\Sigma_p^p)^{-1} (p^{t+1} - p^t) \quad (2.19)$$

Таким чином, оптимальне значення цієї моделі руху досягається, коли поза людини незмінна на відео. Зміна пози під час руху виявилася «дозволенним шумом».

У цій роботі ми розширюємо цю модель руху. Для опису руху суглобів людини використовуються лінійні динамічні системи. З цією метою прихований стан моделі розширюється характеристиками руху кожного суглоба. У роботі розглядається лінійна модель руху суглоба, тобто стан кожного суглоба описується його положенням $P(t)$ на рамі та миттєвою швидкістю руху $v(t) \in R^2$. За аналогією з людськими позами, швидкості всіх суглобів у відеоряді представлені V . Якщо ми використовуємо $h(t) = [P(t), v(t)]$ для представлення стану розглянутого суглоба тіла людини в кадрі t , то запропонована модель руху набуває наступного вигляду:

$$\sum_{t=1}^{T-1} \psi(p^{t+1}, p^t, \theta) = \sum_{i=1}^K \psi_i^0 \quad (2.20)$$

$$\sum_{t=1}^{T-1} \psi(p^{t+1}, p^t, \theta) = \sum_{i=1}^K (\psi_i^0 (v_i^1 + \sum_{t=1}^{T-1} \psi_i^t (h_i^{t+1}, h_i^t, \theta)) + \sum_{t=1}^{T-1} \eta^t (s^{t+1}, s^t)) \quad (2.21)$$

Тому під час аналізу було розроблено серію алгоритмів для попередньої обробки вхідних даних для подання в нейронну мережу. Детально описано алгоритм, а також принципи роботи шарів згорткової нейронної мережі, що використовуються для обробки зображень і послідовностей зображень: згорткових шарів і шарів субдискретизації. Наведено спосіб міжрівневої передачі даних. Наведено опис функцій і методів активації, які будуть використані при розробці власної архітектури системи для розпізнавання поз людини на зображеннях і відео. У цьому розділі пропонується алгоритм оцінки пози людини у відеоряді, який враховує положення кожного суглоба тіла людини в кадрі.

3 ОСОБЛИВОСТІ ПОБУДОВИ МОДЕЛІ ДЛЯ НЕЙРОННОЇ МЕРЕЖІ ДЛЯ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ

3.1 Обґрунтування базової моделі нейромережі

Під час процесу проектування системи було проведено кілька експериментів, щоб вибрати найбільш прийнятну модель, класифікатор і найбільш прийнятну функцію втрат.

Для полегшення розробки також був розроблений модуль для зберігання та читання даних, а також модуль, що дозволяє задавати додаткові параметри нейронної мережі, що навчається (кількість епох, ймовірність випадання, кількість шарів тощо).

Набір даних, наданий Kaggle, складається з понад 6000 фотографій, зроблених камерою LeapMotion і використаних для навчання.

Під час вибору найбільш підходящої моделі нейронної мережі використовувалися кращі практики TRVZ (Теорія творчого вирішення проблем).

З початку 20 століття публікуються списки різноманітних винайдених технологій. Автори публікацій доволіно включають ті технології, які здаються їм найкращими, не враховуючи природу цих технологій.

Під технічними (системними) протиріччями в TRVZ розуміються ситуації, коли спроби поліпшити одну характеристику технічної системи призводять до погіршення іншої характеристики [20]. Аналіз великої кількості патентної інформації показує, що лише 40 найпотужніших технологій можуть забезпечити ефективні рішення для усунення приблизно 15 000 технічних протиріч. У задачі розпізнавання жестів було виявлено кілька технічних протиріч і запропоновано рішення за допомогою TRVZ [21].

Тому виникає проблема: необхідно створити модель нейронної мережі для розпізнавання жестів людини. Найважливішими проблемами, які виникли під час розробки, були наявність фону, недостатня кількість фотографій (зразків), проблема розпізнавання жестів лише однієї людини.

Для вирішення цих проблем із принципів вирішення технічних протиріч були обрані наступні пункти: принцип асиметрії, принцип зміщення, принцип переходу в інший вимір, принцип єдності.

Принцип асиметрії - Цей принцип говорить, що для вирішення деяких протиріч необхідно:

а) форма об'єкта змінюється від симетричної до асиметричної; б) якщо об'єкт асиметричний, ступінь асиметрії збільшується.

У нейронних мережах цей підхід використовується повсюдно при створенні навчальних наборів даних. Такий підхід допоможе нам вирішити технічне протиріччя - розпізнавання жестів може виконувати тільки один оператор. Основна проблема (технічне протиріччя) полягає в тому, що жести різних людей (операторів) повинні розпізнаватися, але використання великих наборів даних не є доцільним.

Тому, щоб вирішити цю проблему, необхідно додати трансформації, які спотворять вихідне зображення. Для цього ми скористаємося наявним інструментом у pytorch під назвою «transform».

```
self.transformations = transforms.Compose([
    transforms.Resize(255),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Рисунок 3.1 - Трансформації зображення

Жест - складний образ. Зображення, що містять жести людини, і сама людина займають багато місця в пам'яті комп'ютера, сповільнюючи роботу нейронних мереж.

Ще одна технічна суперечність полягає в тому, що розпізнавання жестів вимагає зчитування великої кількості даних із зображення. В той же час велика кількість даних, представлених на зображенні, спричиняє повільне розпізнавання жестів.

Для вирішення цієї проблеми звертає увагу наступний принцип - принцип видалення: відокремити від об'єкта «непотрібні» частини («непотрібні» властивості) або, навпаки, виділити лише потрібні частини (потрібні властивості).

Цей принцип привернув увагу розробників нейронних мереж і навіть отримав власну назву - підвибірка. Операція субвибірки (англ. Subsampling, англ. Pooling, також перекладається як

«Операція підвибірки» або операція об'єднання), яка виконує зменшення розмірності на згенерованій карті ознак. У цій мережевій архітектурі інформація про наявність потрібної ознаки вважається важливішою, ніж точне знання її координат, тому найбільший вибирається з кількох сусідніх нейронів карти ознак і використовується як нейрон. Стислі карти функцій меншого розміру. Завдяки цій операції, крім прискорення подальших обчислень, мережа стає більш інваріантною до масштабу вхідного зображення.

У Pytorch є інструменти, необхідні для реалізації (наприклад, MaxPool2d) поділу карти функцій на комірки розміром 2x2 елементи та вибору з них максимального значення. Формально шар можна описати наступним чином:

$$x^l = f(a^l \text{subsample}(x^{l-1}) + b^l) \quad (3.1)$$

де x^l - вихід l -го шару;

$f()$ - функція активації;

a, b - коефіцієнти;

$\text{subsample}()$ - операція дискретизації локального максимуму.

Використання цього шару покращує розпізнавання зразків, які змінюють масштаб (зменшений або збільшений). Крім того, цей шар зберігає пам'ять і прискорює навчання.

Крім того, вибраний набір даних частково вирішує проблему «непотрібних частин», викликану тим, що фотографії в наборі даних не мають фону.

Цей принцип також застосовується до згорткових шарів, де ми вибираємо ознаки для розпізнавання зображень. Pytorch надає можливість задовольнити цю вимогу за допомогою методів Conv2d. Для зручності експериментів був створений метод conv3x3.

```
def conv3x3(in_planes, out_planes, stride=1, groups=1, dilation=1, padding = 1):
    """3x3 convolution with padding"""
    return nn.Conv2d(in_planes, out_planes, kernel_size=3, stride=stride, padding=padding, groups=groups, bias=False, dilation=dilation)
```

Рисунок 3.2 - Шар згортки

Для наших власних експериментів були створені додаткові методи з використанням наших власних ядер, щоб задовольнити вимоги принципу асиметрії - з використанням викривлення Гауса, ядер пошуку контурів, підвищення різкості зображення та наших власних ядер. Ось приклад створення ядра з спотворенням Гауса:

```
def get_gaussian_kernel(kernel_size=3, sigma=2, channels=3):
    x_coord = torch.arange(kernel_size)
    x_grid = x_coord.repeat(kernel_size).view(kernel_size, kernel_size)
    y_grid = x_grid.t()
    xy_grid = torch.stack([x_grid, y_grid], dim=-1).float()
    mean = (kernel_size - 1)/2.
    variance = sigma**2.
    gaussian_kernel = (1./(2.*pi*variance)) *\
        torch.exp(
            -torch.sum((xy_grid - mean)**2., dim=-1) /\
            (2*variance)
        )
    gaussian_kernel = gaussian_kernel / torch.sum(gaussian_kernel)
    gaussian_kernel = gaussian_kernel.view(1, 1, kernel_size, kernel_size)
    gaussian_kernel = gaussian_kernel.repeat(channels, 1, 1, 1)
    gaussian_filter = nn.Conv2d(in_channels=channels, out_channels=channels,
                               kernel_size=kernel_size, groups=channels, bias=False)
    gaussian_filter.weight.data = gaussian_kernel
    gaussian_filter.weight.requires_grad = False
```

Рисунок 3.3 - Ядро з гаусовим викривленням

Такий же метод був створений для інших ядер:

```
outline_kernel_weights = torch.Tensor([[[-1, -1, -1], [-1, 8, -1], [-1, -1, -1]]].unsqueeze(0).unsqueeze(0))
sharpen_kernel_weights = torch.Tensor([[0, -1, 0], [-1, 5, -1], [0, -1, 0]]).unsqueeze(0).unsqueeze(0)
kustom_kernel_weights = torch.Tensor([[[-1, 0, 1], [0, 0, 0], [1, 0, -1]]].unsqueeze(0).unsqueeze(0))
```

Рисунок 3.4 - Створення власних ядер з викривленнями

Такий же метод був створений для інших ядер. Одним з найцікавіших принципів побудови моделей нейронної мережі є принцип переходу в інший вимір:

а) Якщо об'єкт набуває здатності переміщатися в двох вимірах (на площині), то усуваються труднощі, пов'язані з переміщенням (або розміщенням) об'єкта по прямій. Тому при переміщенні об'єктів у тривимірний простір усуваються завдання, пов'язані з переміщенням (або розміщенням) на площині.

б) Використовуйте багатошаровий макет об'єкта замість одношарового.

Багато дослідників у всьому світі працюють над цим завданням (проекування 2D-зображень на 3D-зображення). Однак ця проблема залишається невирішеною. Люди вирішують такі завдання за допомогою просторового мислення, одного з важливих компонентів інтелекту. З його допомогою ми можемо орієнтуватися в просторі, розв'язувати геометричні задачі, зображувати об'єкти в тривимірному вигляді. Іншими словами, просторове мислення - це здатність людини уявити об'єкт у всіх його деталях і уявленнях і будь-яким чином перетворити цей об'єкт. Нейробіологічні роботи Джона О'Кіфа, Мей-Брітт Мозер і Едварда Мозера допомогли описати вроджені системи просторової орієнтації. За свої дослідження вчені отримали Нобелівську премію з фізіології та медицини 2014 року за «відкриття клітин, відповідальних за систему позиціонування мозку» - половину премії, присуджену О'Кіфу, а половину - Мозерам.

Однак їх робота дає лише детальний опис того, як працює нервова система під час вирішення просторових задач. Навіть сьогодні просторове мислення людини залишається загадкою для вчених усього світу.

Оскільки нейронні мережі в системах штучного інтелекту побудовані за тими ж принципами, що і нейронні мережі в мозку, поки не варто чекати відкриттів у цьому напрямку в області штучного інтелекту. Проте запропонований Х. Альтшуллером принцип вирішення технічних протиріч буде застосований у штучному інтелекті через десятиліття чи навіть століття.

Останній принцип, який привертає увагу, це принцип єдності:

а) З'єднувати об'єкти одного типу або об'єкти, які використовуються для суміжних операцій; б) Своєчасно об'єднувати подібні або суміжні операції.

У нейронних мережах цей принцип також застосовується до так званих повнозв'язаних шарів - прихованих шарів штучних нейронних мереж перцептронного типу.

Основне призначення цього шару - перетворити сигнали, отримані попередніми шарами мережі, в одновимірну форму та виділити характеристики одновимірного шару. Цей рівень також можна використовувати як кінцевий (вихідний) рівень нейронної мережі, де результатом є ймовірність того, що вхідне зображення належить до певного класу. У наших власних експериментах було створено декілька варіацій повністю з'єднаних шарів.

Нижче представлена архітектура створеної моделі.

Перша модель. Ця модель складається з одного шару. При побудові моделі використовувалися лише об'єднуючі принципи.

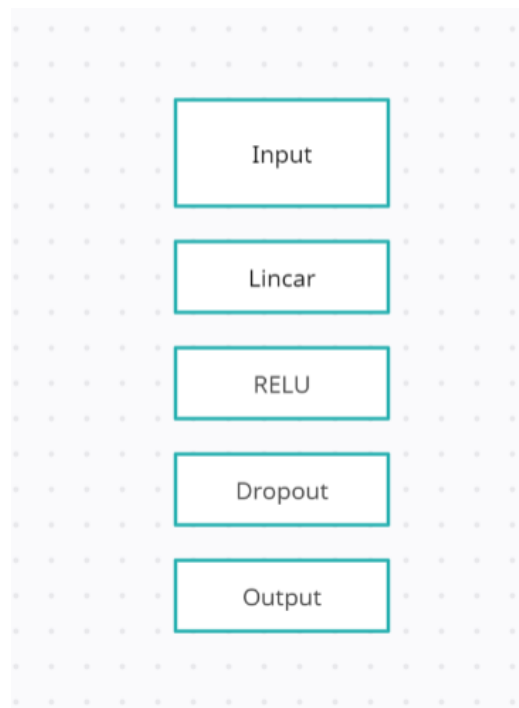


Рисунок 3.5 - Нейронна мережа що складається з одного шару

Результати роботи наведено на рисунку 3.6

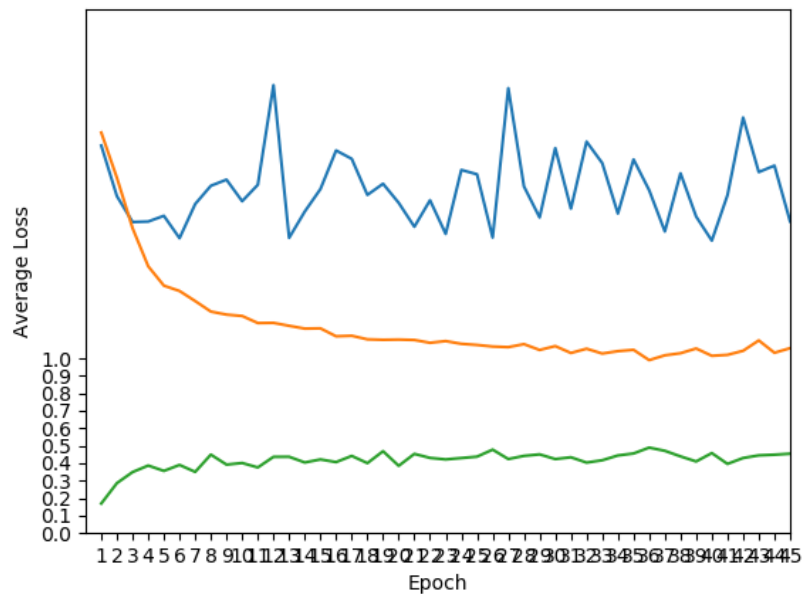


Рисунок 3.6 - Результати роботи неройнної мережі що складається з єдиного шару

Як бачимо, з більш ніж 45 епох модель показала найкращі результати на 18 епосі - досягнувши точності 46,848958%, і припинила навчання.

ТРВЗ вважає, що якщо можливості розвитку, тобто можливості реорганізації всередині системи, вичерпані, то система змінюється для включення в суперсистему як компонента суперсистеми або для виконання функцій суперсистеми, після чого, Подальший розвиток системи відбувається за надсистемним складом і рівнями.

Відповідно до ТРВЗ, перехід до суперсистеми через уніфікацію йде по шляху «єдине-подвійне-багато» і починається з об'єднання двох (або більше) вихідних систем. Подвійні системи отримують шляхом поєднання двох ідентичних (або майже ідентичних) єдиних систем з певною метою. Поєднати - означає встановити якийсь зв'язок. Мультисистеми виходять шляхом об'єднання кількох одиночних систем. Через такі комбінації кожна окрема система зберігає свої властивості, а сукупний зв'язок із властивостями окремих систем допомагає новій системі (по суті, суперсистемі) виконувати нові функції, які не можуть бути досягнуті окремими системами. Іншими словами, якщо об'єднати кілька єдиних систем,

певні потреби суспільства (функції системної ієрархії) можуть бути задоволені ефективніше, ніж раніше [22].

Ви можете комбінувати:

- гомогенна (одна функція) система,
- системи зі змінними (дещо відмінними) характеристиками,
- гетерогенні (з різними функціями) системи,
- інверсна (виконує протилежну функцію) система.

Об'єднання і злиття систем у всіх конкретних випадках відбувається на одному етапі.

Перетворення «непарний-парний-множинне» може бути виконане на будь-якому етапі розробки та застосовне до всіх рівнів ієрархії.

Створимо багатоваріантну систему для вирішення задачі розпізнавання жестів.

Друга модель складається з чотирьох шарів. Архітектура моделі показана на рисунку 3.7.



Рисунок 3.7 - Архітектура нейронної мережі що складається з чотирьох шарів

Як бачите, модель дуже проста і складається з чотирьох шарів. Він використовує функцію RELU як активацію.

Результати навчання цієї моделі показані на рисунку 3.8.

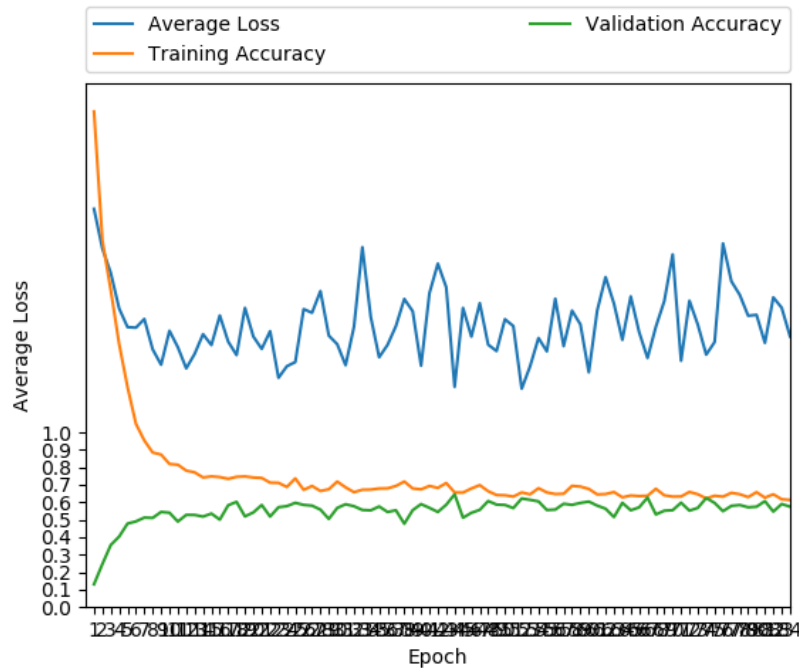


Рисунок 3.8 - Результати роботи моделі що складається з чотирьох шарів

Найкращу продуктивність цієї моделі показує Epoch 66 з точністю 62,656250%. Кілька систем показали значно кращі результати.

Наступним етапом створення системи є використання принципів асиметрії та зміщення.

Для цього набір даних трансформується, а вся система ускладнюється шляхом додавання згорткових шарів, шарів нормалізації та об'єднання. Експерименти також показують, що модель працює краще, якщо її ядро згорнуто в згортковий шар. Найкращі результати показало ядро під назвою Gaussian kernel для розмивання зображень.

Архітектура системи показана на малюнку нижче (3.9).

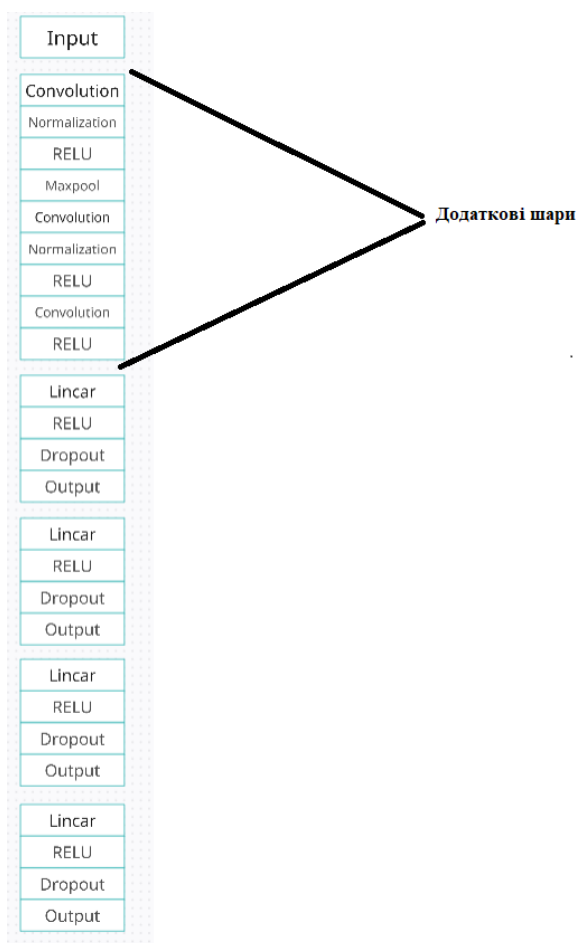


Рисунок 3.9 - Архітектура моделі з додатковими шарами

Результати роботи цієї моделі показані на малюнку 3.10.

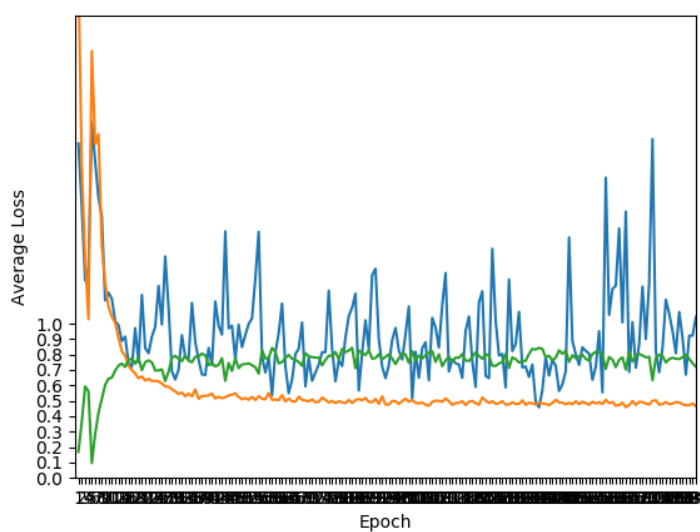


Рисунок 3.10 - Результати роботи моделі

Найкращий показник показує 76 періодів - 92,375000 %.

Остання модель показала найкращі результати. Тому було вирішено використовувати цю модель для подальшого розвитку.

3.2 Опис інструментів розробки

Головна мета - створення мультиплатформних рішень з використанням відкритих технологій.

Система розроблена з використанням новітніх технологій і методів, які сьогодні використовуються в ІТ-індустрії. Ми дуже серйозно ставимося до вказівок провідних світових компаній, таких як Microsoft, Google, Facebook та інших.

Розроблена система складається з трьох модулів: навчального модуля, модуля виведення інформації про результати навчання та модуля музичного плеєра. Тому ми розглянемо характеристики інструментів, які використовуються при розробці цих модулів.

Використовуйте мову програмування Python, бібліотеку комп'ютерного зору openCV і бібліотеку нейронної мережі pytorch для розробки навчальних модулів. Щоб пришвидшити роботу програми, було вирішено вручну зібрати бібліотеки openCV за допомогою інструменту CMake - це дало можливість виконувати обчислення на графічному процесорі, завдяки чому програма працювала в рази швидше [24].

В даний час з бурхливим розвитком персональних комп'ютерних пристроїв вимоги до мов програмування поступово змінюються. Оскільки персональні комп'ютери ставали все більш потужними і почали забезпечувати достатню швидкість для виконання інтерпретованих програм, інтерпретовані мови почали відігравати все більш важливу роль. Єдиною істотною перевагою скомпільованих мов програмування є високошвидкісний код, який вони створюють. Коли швидкість виконання програми не є критичною величиною, інтерпретована мова є

хорошим вибором, оскільки це простіший і гнучкіший інструмент програмування, який дозволяє швидко отримувати результати.

У зв'язку з цим Python, створений Гвідо ван Россумом на початку 1990-х років, зараз досить поширений.

Переваги цієї мови:

1. Python реалізовано майже на всіх платформах і операційних системах. Першою такою мовою була C, але типи даних у мові програмування C могли займати різні обсяги пам'яті на різних машинах, що було певною перешкодою під час написання незалежних від платформи програм. Python немає таких недоліків.

2. Розширюваність мови. Їй приділено багато уваги, і, як пише сам автор, мова розглядається саме як розширювана. Це означає, що кожен зацікавлений програміст має можливість вдосконалити мову. Інтерпретатор Python написаний на C, і вихідний код можна використовувати для будь-якої операції. При бажанні мови C і C++ можна вставляти в програми, написані на Python, і використовувати як вбудовані оболонки. Крім того, отримайте «розширений» інтерпретатор з новими можливостями, написавши доповнення до Python на C і скомпільовавши програму.

3. Пропонує велику кількість модулів, які можна підключити до програми для надання додаткових функцій. Такі модулі написані на мовах C і Python і можуть бути розроблені будь-яким достатньо кваліфікованим програмістом. Як приклад можна послатися на такий модуль:

Числовий NumPy - розширені математичні функції, такі як операції над цілими векторами та матрицями;

Tkinter - створюйте програми за допомогою графічного інтерфейсу користувача (GUI) на основі широко розповсюдженого інтерфейсу Tk на Включено до таких поширених операційних систем Microsoft Windows 95 OSR 2, 98 і Windows NT 4.0.

Недоліки мови:

Єдиним недоліком є те, що програми Python виконуються відносно повільно, оскільки мова інтерпретується.

Мова програмування Python дуже популярна в області розробки нейронних мереж завдяки своїй простоті і зручності. Тому спеціально для мови Python було розроблено декілька бібліотек - pytorch, keras, tensorflow. Ці бібліотеки полегшують роботу з нейронними мережами, а також надають можливості для експериментів.

PyTorch - це бібліотека Python, яка забезпечує тензорні обчислення за допомогою GPU, схожі на NumPy. Крім того, PyTorch надає багатий API для вирішення прикладних проблем, пов'язаних із нейронними мережами [23].

PyTorch відрізняється від інших фреймворків машинного навчання тим, що він не використовує статичні обчислювальні графіки (визначені заздалегідь, негайно та з часом, як TensorFlow, Caffe2 або MXNet). Навпаки, обчислювальні графіки в PyTorch є динамічними та динамічно визначеними.

OpenCV (Бібліотека комп'ютерного зору з відкритим кодом) - це бібліотека функцій і алгоритмів з відкритим кодом для комп'ютерного зору, обробки зображень і загальних чисельних алгоритмів. Бібліотека надає інструменти для обробки й аналізу вмісту зображень, включаючи ідентифікацію об'єктів на фотографіях (таких як обличчя та люди, текст тощо), відстеження руху об'єктів, трансформацію зображень, застосування методів машинного навчання та ідентифікацію типових зображень. Різні графічні елементи.

Ця бібліотека була розроблена Intel і наразі підтримується Willow Garage та Itseez. Вихідний код бібліотеки написаний мовою C++ і поширюється за ліцензією BSD. Підготовлені прив'язки для різних мов програмування, таких як Python, Java, Ruby, Matlab, Lua тощо. Безкоштовно для академічних і комерційних цілей.

CMake використовується для створення бібліотеки openCV.

CMake - це кросплатформний генератор сценаріїв збірки з відкритим кодом. CMake не обробляє компіляцію безпосередньо, а лише генерує файли керування компіляцією з файлу CMakeLists.txt [24]:

- Makefile в системах Unix, використовується для збирання за допомогою make;
- файли проекту/робочої області (.dsp/.dsw) у Windows для складання за допомогою Visual C++;

- проекти XCode на Mac OS X.

Переваги CMake включають: просту мову сценаріїв, спосіб розширення функціональності за допомогою модулів, мінімальну кількість залежностей (не прив'язаних до M4, Perl або Python), підтримку кешування, наявність інструментів крос-компіляції, підтримку генерації файлів збірки для різноманітних системного асемблера та компілятора, утиліти `ctest` і `ctest` можна використовувати для визначення сценаріїв тестування та складання пакунків, а утиліту `cmake-gui` можна використовувати для інтерактивного налаштування параметрів складання.

CMake використовується як альтернатива Autotools такими проектами, як KDE, LLVM/Clang, MySQL, MariaDB, ReactOS і Blender. сирий

Код CMake написаний мовою C++ і поширюється за ліцензією BSD.

Розробити модуль виведення навчальної інформації з використанням мови програмування Python.

Використовуйте мову програмування Python, бібліотеку `vlc`, бібліотеку `openCV` і модуль розпізнавання мовлення для розробки модуля музичного плеєра. Для зберігання даних використовується інструмент `git`.

VLC Media Player (VideoLAN Client англійською) - це безкоштовний портативний мультиплатформенний медіаплеєр із відкритим кодом, розроблений у рамках проекту VideoLAN. Цей медіаплеєр доступний для багатьох настільних операційних систем, таких як Windows, Linux, Mac OS. VLC також працює на мобільних операційних системах, таких як Tizen, IOS, Android тощо, де основним методом розповсюдження є цифровий (Apple App Store, Android Google Play і Windows Store). [25]

VLC підтримує багато типів аудіо- та відеоформатів стиснених файлів, включаючи DVD-Video, Video CD і протоколи потокової передачі. Він здатний передавати та декодувати мультимедійні файли через комп'ютерні мережі. За замовчуванням VLC включає велику кількість безкоштовних бібліотек декодування та кодування, усуваючи необхідність пошуку та калібрування власних плагінів. Хоча програвач VLC переважно використовує власні мультиплектори та демультіплектори, він також використовує сторонні бібліотеки з інших проектів,

наприклад FFmpeg. VLC є першим програвачем, який підтримує зашифровані DVD-диски на платформах Linux і Mac OS, використовуючи бібліотеку дешифрування DVD libdvdcss.

При проектуванні системи досліджувалися та аналізувалися тематика та вимоги замовника. Після ретельного аналізу було вирішено розробити програмний продукт, реалізований за допомогою бібліотеки OpenCV, завантаженої для GPU за допомогою інструменту CMake, мови програмування Python, бібліотеки pytorch і бібліотеки vlc для обробки аудіо- та медіаінформації.

Python - це мова програмування загального призначення, призначена в першу чергу для підвищення продуктивності програмістів. На Python можна писати майже будь-що (веб/десктопні програми, ігри, сценарії автоматизації, складні обчислювальні системи, системи керування життєзабезпеченням тощо) без будь-яких помітних проблем. Завдяки простоті коду подальше обслуговування програм, написаних на Python, стає легшим і приємнішим, що з точки зору бізнесу означає зниження витрат і підвищення продуктивності співробітників. [27]

Для створення моделі нейронної мережі була обрана бібліотека pyTorch - це науковий обчислювальний пакет на основі Python, який використовує потужність графічних процесорів. Це також одна з найкращих дослідницьких платформ глибокого навчання, розроблена для забезпечення максимальної гнучкості та швидкості. Він відомий тим, що надає дві найсучасніші функції, а саме потужну підтримку прискорення GPU та тензорних обчислень для створення глибоких нейронних мереж. [28]

Однією з ключових причин успіху PyTorch є те, що він використовує всі переваги мови Python і надає можливість створювати власні моделі нейронних мереж.

Деякі основні переваги PyTorch:

1. Простий інтерфейс: він забезпечує простий у використанні API, тому ним дуже легко керувати та він працює як Python.

2. Обчислювальні графіки: на додаток до цього, PyTorch також надає чудову структуру, яка забезпечує динамічні обчислювальні графіки, щоб ви могли змінювати їх під час виконання.

Для попередньої обробки було вибрано використання бібліотеки openCV, оскільки вона написана на C, добре оптимізована та може використовувати переваги багатоядерних процесорів. OpenCV, встановлений за допомогою стандартного пакета, споживає ЦП, що зрештою впливає на продуктивність. Ось чому я вирішив самостійно скомпілювати цю бібліотеку для GPU, використовуючи вихідний код і інструмент cmake.

Щоб забезпечити розпізнавання жестів людини, ми створили спеціальну модель і отримали дані з набору даних Kaggle. Цей набір даних було вибрано, оскільки він містить понад 6000 зображень 15 різних жестів рук, зроблених за допомогою камери LeapMotion.

Для обробки аудіоінформації була обрана бібліотека vlc, оскільки вона є проектом з відкритим кодом і має надзвичайно високий рівень підтримки розробників. Ще однією важливою особливістю є висока швидкість роботи, надійність і кількість вбудованих функцій. Бібліотеку можна використовувати в будь-якій сучасній операційній системі або контейнері Docker.

Разом ці технології дозволяють створювати якісні та надійні продукти, які не захищені патентними претензіями розробників, оскільки всі ці технології захищені ліцензіями, які виключають таку можливість і надають доступ до вихідного коду цих проектів Access.

Згортова нейронна мережа - це штучна архітектура нейронної мережі, яка імітує характеристики зорової кори головного мозку. Він складається з кількох багатовимірних шарів, призначених для ефективного розпізнавання складних зображень. Ця модель була вперше запропонована Янном Лекуном у 1998 році для розпізнавання рукописних символів. Унікальною особливістю нейромережевої моделі, запропонованої Лекуном, є введення в архітектуру багат шарового перцептрона згорткових шарів, де кожен нейрон пов'язаний лише з невеликою частиною нейронів попереднього шару. Такі функції дають змогу виділяти

оригінальні об'єкти на вихідному документі та на наступних рівнях мережі комбінувати вибрані об'єкти для отримання всіх більш складних елементів і таким чином виявляти складні об'єкти на зображеннях.

У той час, коли створювалися перші ZNM, комп'ютери мали відносно невелику обчислювальну потужність, а накопичених візуальних даних на той час було недостатньо для того, щоб такі математичні моделі могли бути широко розповсюджені. На даний момент штучний інтелект і машинне навчання є одними з основних напрямків розвитку більшості галузей, пов'язаних з інформаційними технологіями.

Однак основні принципи побудови таких мережевих архітектур мало змінилися з 1998 року.

Згорткові нейронні мережі можуть мати багато типів багатовимірних шарів (переважно двовимірні шари, наприклад, у мережах обробки зображень, і тривимірні шари, наприклад, шляхом додавання кількох кольорових каналів до зображення):

- вхідний шар: вхідне зображення містить кілька кольорових каналів.
- згортковий шар (Convolution): на відміну від перцептрона, усі нейрони цього шару з'єднані лише з деякими нейронами попереднього шару.
- шар підвибірки (pooling, subsampling): вибирає найважливіші характеристики попереднього шару та значно зменшує розмірність наступних шарів мережі.
- повністю підключений рівень: прихований шар нейронної мережі типу штучного перцептрона.

Крім того, після згорткових і повністю з'єднаних шарів може бути застосована функція активації нейронів, яка перетворює сигнал нейронів і формує вихідний сигнал.

Згорткові нейронні мережі демонструють найкращу продуктивність у розпізнаванні образів, тому їх було обрано для створення нашої власної системи.

Спосіб реалізації програмного продукту обрано та обґрунтовано на основі аналізу предметної області та вимог, висунутих у процесі проектування. Було

розроблено набір методів для створення моделей нейронної мережі, отримання результатів навчання, виділення та розпізнавання людських жестів, а також використання людських жестів для керування медіаплеєрами.

4. СИСТЕМИ РОЗПІЗНАВАННЯ ПОЗИ ЛЮДИНИ

4.1 Опис функціональності системи

Для вирішення проблеми розпізнавання жестів людини необхідно створити систему, що забезпечує можливість автоматичного розпізнавання жесту одного або кількох людей у відеокліпі. Користувачі повинні мати можливість вказати шляхи доступу до даних для ідентифікації. Розроблена система повинна відповідати таким функціональним вимогам:

- система повинна приймати потік даних користувача (зображення, відеокліпи, відеопотоки);
- система повинна виділяти рухомі об'єкти на відео та виявляти людей;
- система повинна відокремлювати послідовність кадрів від відеокліпу;
- система має розпізнавати пози людей у відеороликах у реальному часі.

Оскільки існують готові бібліотеки та надбудови для створення нейронних мереж, система повинна відповідати наступним нефункціональним вимогам:

- система повинна бути реалізована мовою Python;
- система повинна використовувати фреймворк для створення нейронних мереж TensorFlow;
- система має бути уніфікованою і не вимагати від користувача додаткових дій для роботи, крім запуску та налаштування вхідних даних.

4.2 Варіанти використання системи

Програми розроблені з використанням мови графічного опису UML для моделювання об'єктів. Встановлюється модель взаємодії між учасником «користувачем» і системою. Діаграма варіантів використання показана на рисунку 4.1.

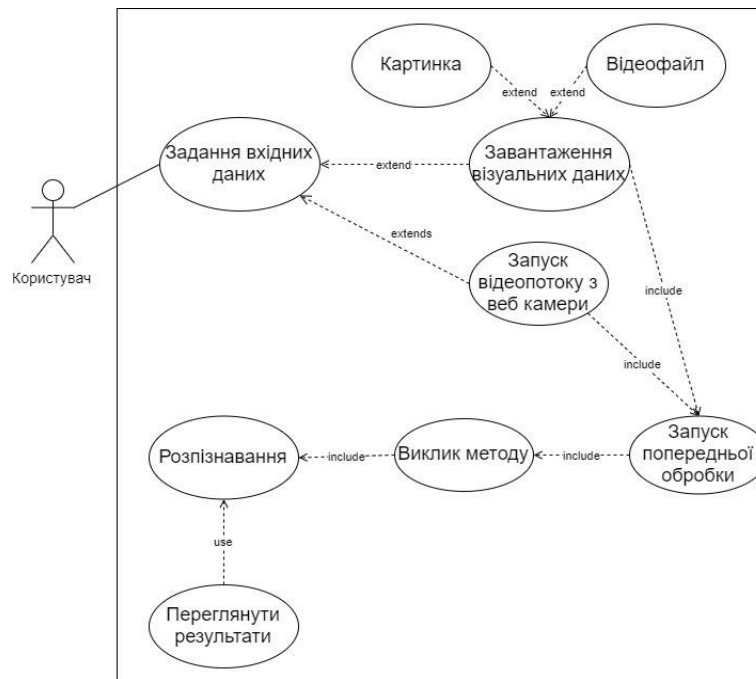


Рисунок 4.1 - Модель взаємодії актора «Користувач» з системою Актори, які взаємодіють з системою

Користувач є єдиним учасником, який використовує систему для розпізнавання людських жестів у реальному часі.

Короткий опис варіантів використання Користувачі можуть:

- встановлення шляху введення даних системи-відеокліпів і параметрів програми;
- запуск попередньої обробки відеопотоків;
- ініціювати розпізнавання пози людини за даними, наданими особою;
- перегляд результатів програми в реальному часі.

4.3 Етапи роботи системи розпізнавання пози людини

Щоб розпізнавати жести людини на зображеннях і відео, необхідно підготувати вхідні дані та застосувати алгоритми машинного навчання для розпізнавання. Додаток А містить детальний опис системи.

Вхідне зображення або відеопотік. Для подальшої обробки відеопотік потрібно розбити на окремі кадри. Отже, першим етапом є дискретизація вхідного відеопотоку. На другому етапі виділяються точки розташування стиків на знімку. На цьому етапі використовуються згорткові нейронні мережі та архітектури ResNet. На цьому етапі ми використовуємо алгоритм Part Strength Field для отримання масиву точок з'єднань і з'єднань пар найближчих з'єднань.

На третьому етапі знайдені пари частин тіла за допомогою лінійної регресії комбінуються в індивідуальні пози.

Розглянемо обробку вхідних даних:

- вхідні елементи зображення - зображення, відеофайли, відеопотоки;
- коефіцієнт масштабування зображення - число від 0,2 до 1. Значення за замовчуванням 0,5. Зображення потрібно збільшити або зменшити, перш ніж надсилати його в Інтернет. Ми встановлюємо цей параметр нижче, щоб зменшити зображення та підвищити швидкість під час подачі через мережу, але за рахунок точності;

- горизонтальний дисплей. Значення за замовчуванням - false. Цей параметр необхідний, якщо позу потрібно перевернути/віддзеркалити горизонтально. Для відео, де відео за замовчуванням перевертається горизонтально (наприклад, веб-камера), ми встановлюємо значення true, щоб поза поверталася в правильній орієнтації;

- крок виходу. Значення за замовчуванням - 16. У внутрішньому представленні цей параметр впливає на висоту і ширину шарів в нейронній мережі. На високому рівні це впливає на точність і швидкість оцінки пози. Менші значення розміру вихідного кроку призводять до вищої точності, але нижчої швидкості, вищі значення призводять до вищої швидкості, але зниженої точності.

Поріг оцінки впевненості пози - від 0 до 1. Значення за замовчуванням 0,5. На верхньому рівні цей параметр контролює мінімальну впевненість у повернутих позах.

На виході алгоритму отримуємо:

- поза з рейтингом впевненості та масивом із 17 ключових точок;
- розташування та рейтинг надійності кожної ключової точки. Так само всі розташування ключових точок мають координати x і y у просторі введення та можуть відображатися безпосередньо на зображенні.

Компоненти, що входять до складу системи.

Main - головний клас, що забезпечує роботу системи. Він має атрибут parameters, який зберігається в деякій структурі Parameters і містить задані користувачем параметри. Метод startPretreatment() запускає сценарій попередньої обробки даних шляхом виклику класу Pretreatment. Метод startNN() запускає нейронну мережу, викликаючи клас NeuralNetwork.

Попередня обробка - класи, відповідальні за попередню обробку даних. Він має атрибут параметра, що зберігається в деякій структурі параметрів, яка містить параметри, передані з основного класу. Метод cutVideo() обрізає відео до заданої довжини, frame() ділить відео на кадри, morph запускає модуль Morph, а scale() запускає клас Scaling.

NeuralNetwork - клас, відповідальний за роботу нейронної мережі. Він містить такі властивості: model - процедурна модель нейронної мережі, trainX - відеоряд навчальних зразків, trainY - категорія навчальних відео, testX - контрольний зразок, testY - "відповідь" на контрольний зразок. Метод readData() дозволяє обчислювати вхідні дані, buildModel(), trainModel() створювати та навчати модель відповідно. Метод testModel() виконує перевірку контрольного зразка та виводить результати.

Клас Scaling відповідає за підготовку структури масштабування. Метод findBorders() шукає крайні точки відображеного скелета та записує їх у цілочисельні атрибути вгору, вниз, вправо та вліво. Потім метод cut() обрізає зображення до цих кінцевих точок.

Компонент `MainProgram` - основний програмний модуль, який керує всіма роботами системи та ініціює роботу інших компонентів, передаючи їм вхідні параметри та запускаючи їх.

`NeuralNetwok` - компонент, який використовує штучну нейронну мережу для розпізнавання жестів людини з відеороликів, що записують рух, і інкапсулює роботу мережі.

`PreprocessingScript` - компонент, який забезпечує попередню обробку вхідних даних системи. Параметри попередньої обробки, включаючи розташування вхідних даних, надходять як вхідні дані від компонента `MainProgram`.

Попередня обробка вхідних даних

Найбільш неформальним етапом використання нейронних мереж для вирішення прикладних завдань є попередня обробка даних. Іноді попередньої обробки можна уникнути або звести до мінімуму, але вона відіграє важливу роль під час обробки зображень. Враховуючи вимогу високої швидкості розпізнавання, алгоритм попередньої обробки не повинен вимагати великих обчислювальних витрат.

Використовувати відеокліпи як вхідні дані для системи. Попередня обробка цього відеоролика складається з кількох етапів:

Фаза перша: рамка. Оскільки нейронні мережі обробляють послідовності зображень, відеокліпи повинні бути розділені на кадри.

Фаза 2: Розширення. Для оптимізації роботи нейронної мережі необхідно зменшити кількість вхідних ознак, тобто вхідні дані необхідно масштабувати до меншого розміру. Оскільки людські фігури в різних відеороликах можуть бути на абсолютно різній відстані від камери, масштаб повинен бути таким, щоб зображення завжди займало весь розмір кадру, а відстань від камери не впливала на результати розпізнавання. Для цього спочатку обріжте зображення з усіх 4 сторін, щоб видалити порожній фон, а потім розтягніть або стисніть до потрібного розміру.

Для попередньої обробки вхідних даних реалізовано утиліту командної оболонки `bash`, яка виконує етапи попередньої обробки послідовно. На вхід утиліти

подається вхідне відео та параметри, що дозволяють регулювати розмір і масштаб відеоролика. На виході ми отримуємо масштабовані кадри, готові до подачі в нейронну мережу. У роботі використовувалися наступні параметри: тривалість сегмента 1 секунда, розмір вхідного зображення 500x500.

4.4 Архітектура згорткової нейронної мережі для вирішення задачі розпізнавання пози людини

Згорткова нейронна мережа ResNet (Residual Network) була обрана для вирішення проблеми розпізнавання пози людини на зображеннях і відеопотоках. Архітектура цієї мережі реалізує ідею передачі вихідних і вхідних значень двох послідовно розташованих згорткових шарів на наступні шари (рис. 4.2) [31].

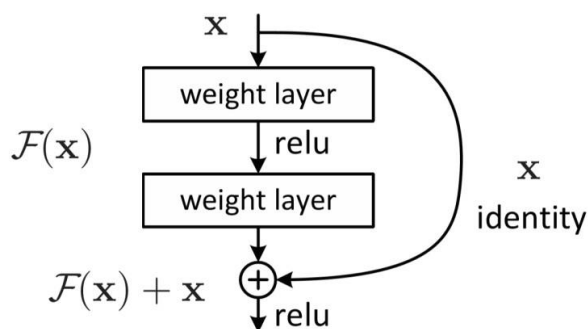


Рисунок 4.2 - Будівельний блок ResNet[31]

Це перший тип нейронної мережі, в якій кількість шарів може бути дуже великою без ризику деградації мережі. Так, деякі мережі можуть досягати 1000 шарів без збільшення помилки класифікації. Таким чином, ця архітектура дозволяє впоратися з так званою деградацією нейронної мережі, тобто на певному етапі збільшення кількості шарів вже не дає позитивних результатів, а лише збільшує кількість помилок (це погіршення якості робота не пов'язана з перепідготовкою).

Існують численні варіації архітектури ResNet: ResNet-50, ResNet-101, ResNet-152 тощо. Цифра в назві вказує на кількість шарів нейронної мережі. Для вирішення спеціалізованої задачі розпізнавання жестів людини була обрана архітектура

ResNet-50. Цей вибір був зроблений на основі того, що цей тип нейронної мережі вимагає найменших обчислювальних витрат для навчання.

Ми пропонуємо розширити концепцію поля в оцінці пози за межі скалярних і векторних полів до складених полів. Ми пропонуємо нову архітектуру нейронної мережі з двома основними мережами. Для кожної частини тіла або суглоба головна мережа передбачає оцінку достовірності, точне розташування та розмір цього суглоба, що ми називаємо картою інтенсивності частин (PIM). Інша велика мережа включає асоціації між частинами, які називаються полями асоціації частин (PAF), які мають нову об'єднану структуру. Система здатна зберігати детальну інформацію на картах із низькою роздільною здатністю. Точна регресія позицій суглобів має вирішальне значення, і в цій роботі ми використовуємо втрату L1 на основі Лапласа [3]. Експерименти показують, що розроблений метод перевершує методи «знизу вгору», а також методи «зверху вниз», встановлені для зображень із низькою роздільною здатністю.

На рисунку 4.3 показана архітектура розробленої системи. Це загальна базова мережа ResNet із двома основними мережами: одна головна мережа передбачає надійність, точне положення та розмір з'єднань, яку ми називаємо Part Intensity Map (MIF), а інша основна мережа передбачає кореляцію між частинами, яка називається Part Поле асоціації (PACH або PAF англійською).

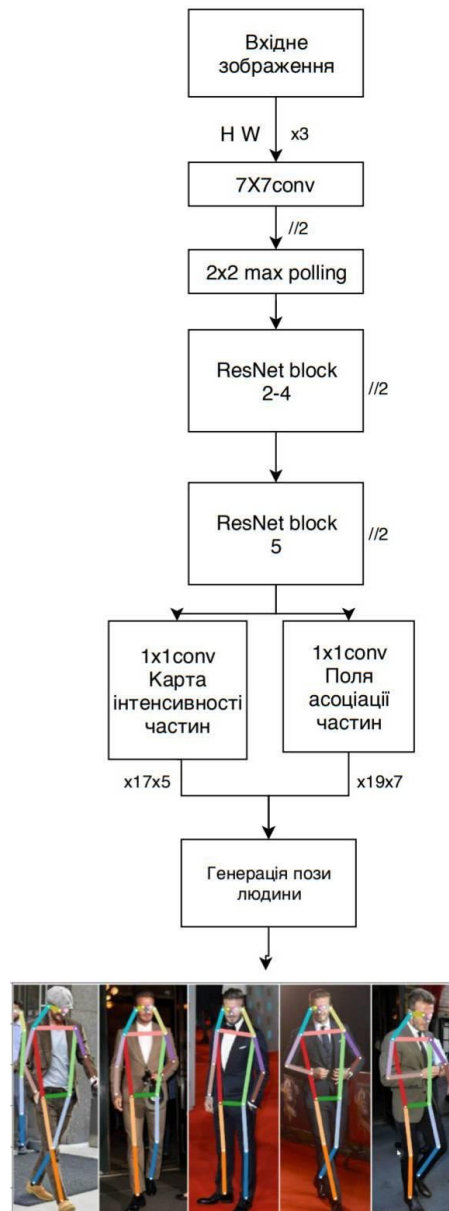


Рисунок 4.3 - Архітектура моделі розпізнавання пози людини

На вхідній стороні ми отримуємо зображення з певними параметрами довжини та ширини та 3 кольоровими каналами. Кодер на основі нейронної мережі створює карти SIC і AFC каналів 17×5 і 19×7 , а декодер - це програма, яка перетворює поля SIC і AFC в оцінки пози, що містять 17 суглобів. Кожне з'єднання представлено координатами x і y та оцінкою достовірності.

Алгоритми виявлення кількох поз здатні ідентифікувати кілька поз на зображенні. Він повільніший і складніший за алгоритми, які визначають окремі пози, але має перевагу: якщо на зображенні кілька людей, виявлені для них ключові

точки з меншою ймовірністю будуть пов'язані з неправильною позою. Таким чином, цей алгоритм може бути більш придатним, навіть якщо потрібно визначити позу лише однієї людини.

Крім того, перевага цього алгоритму полягає в тому, що кількість людей на вхідному зображенні не впливає на продуктивність. Час обчислення однаковий незалежно від того, чи виявлено 15 чи 5 осіб.

Діаграма міцності деталей

Карти інтенсивності частин виявляють і точно визначають частини тіла. У цій роботі [32] представлено синтез карт надійності через регресію для визначення критичних точок. У цій статті ми підсумовуємо цю техніку мовою складених полів і додаємо шкалу σ як новий компонент для формування нових карт часткової інтенсивності.

КІС має складну структуру. Вони складаються зі скалярного компонента, який представляє рівень достовірності, векторного компонента, який представляє найближчу частину тіла певного типу, та іншого скалярного компонента, який представляє розмір суглоба. Більш формально, у кожній позиції виходу (i, j) , впевненість прогнозу CSI c , вектор (x, y) з інверсією b і масштабом σ , і може бути записаний як:

$$p^{ij} = \{p_c^{ij}, p_x^{ij}, p_y^{ij}, p_b^{ij}, p_\sigma^{ij}\} \quad (4.1)$$

Коли мережа обробляє зображення, вона повертає теплову карту разом із векторами зміщення, які можна декодувати, щоб знайти високодостовірні області ключових точок поз на зображенні. Графік довіри MIF дуже грубий. На малюнку 4.4(a) показано карту довіри для лівого плеча на прикладі зображення. Щоб покращити локалізацію цієї карти довіри, ми об'єднуємо її з векторною частиною MIF, показаної на малюнку 4.4(b), у карту довіри високої роздільної здатності.

Карту довіри високої роздільної здатності $f(x, y)$ генерують за допомогою згорткового ненормалізованого ядра Гауса N ширини p_σ на цілі регресії з карти інтенсивності сегмента, зваженої за довірою:

$$f(x, y) = \sum_{ij} p_c^{ij} N(x, y | p_x^{ij}, p_y^{ij}, p_\sigma^{ij}) \quad (4.2)$$

Це рівняння підкреслює вільний від мережі характер позиціонування. Просторове вимірювання σ суглоба вивчається як частина поля. На малюнку 4.4(с) показано приклад. Отриманий локальний графік суглобів використовується для створення поз і оцінки позицій новостворених суглобів.

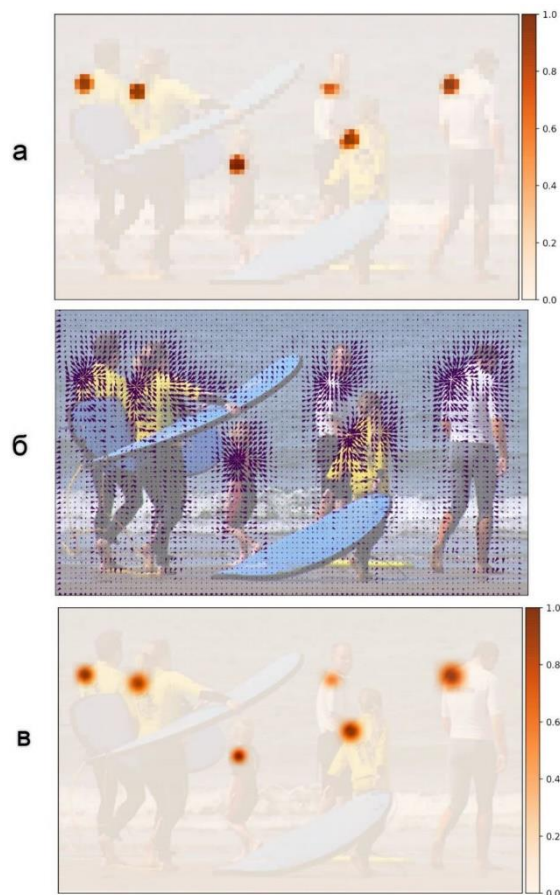


Рисунок 4.4. - Візуалізація компонентів КІС в лівому плечі. Це один із 17 компонентів КІС. Карту достовірності показано на (а), а векторне поле показано на (б). Поєднання карти вірогідності, векторних компонентів і масштабу показано на (с).

Частина пов'язаних полів

Об'єднання суглобів у кілька поз може бути проблемою в людних сценах, де люди частково накладаються один на одного. У цьому випадку ефективним є двоетапний низхідний процес: спочатку виявляють обмежувальні особини, а потім знаходять тип з'єднання для кожного граничного поля.

Ми рекомендуємо використовувати поля асоціації частин (PAF), щоб зв'язати спільні розташування частин у пози. Принципова схема PAF показана на рисунку 4.5.

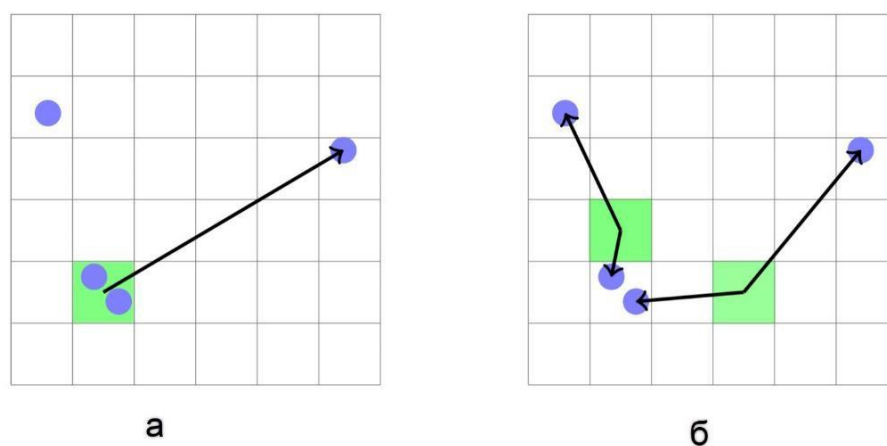


Рисунок 4.5 - Візуалізація різниці між змінами середнього діапазону PersonLab(a) і PAF(b) на сітці функціональної карти.

Сині кола позначають точки з'єднання, а надійність - зеленим. Зсув середнього діапазону (а) починається від центру клітинки карти. Пов'язані поля в частині (б) мають точність із плаваючою комою.

На виході кожного такого розташування є впевненість передбачення PAF, два вектори, що з'єднують дві частини цієї асоціації, і дві ширини b точки просторової регресії. ПАУ виражається як:

$$a^{ij} = \{a_c^{ij}, a_{x1}^{ij}, a_{y1}^{ij}, a_{b1}^{ij}, a_{x2}^{ij}, a_{y2}^{ij}, a_{b2}^{ij}\} \quad (4.3)$$

Візуалізація зв'язку між лівим плечем і лівим стегном показана на малюнку

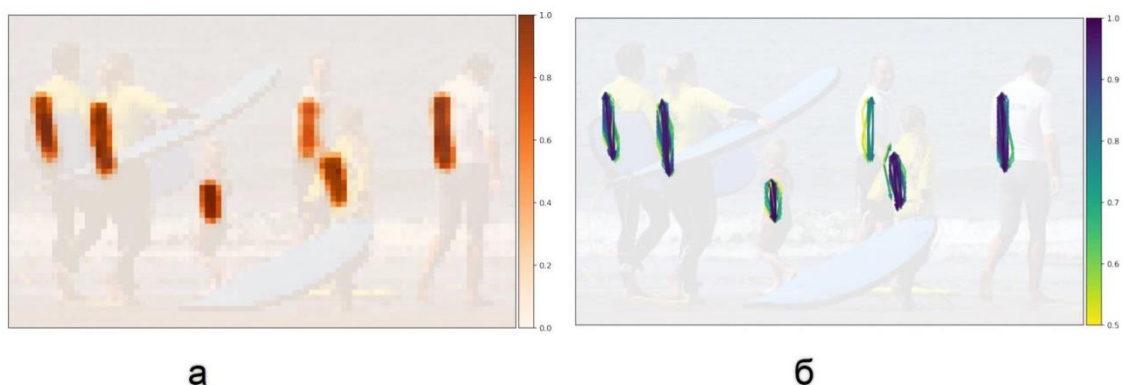


Рисунок 4.6 - Візуалізація компонентів ЛАГ, що відносять ліве плече до лівого стегна.

Це одне з 19 полів РАСН. Кожне розташування на карті зображення є початком двох векторів, які вказують на плечі та стегна для кореляції. Частина (а) показує початкову впевненість асоціації a_s , а частина (б) показує компоненти вектора для $a_s > 0,5$.

Обидві кінцеві точки локалізовані за допомогою регресії, на яку не впливає дискретизація, оскільки вони побудовані з використанням підходу на основі сітки. Це допомагає точно визначити загальне розташування людей, які знаходяться поруч, і розділити їх на чіткі нотатки для подальшого аналізу.

У наборі даних СОСО клас особи має 19 з'єднань, і кожне з'єднання поєднує два типи з'єднань. Наприклад, існує асоціація правого коліна з правою щиколоткою. Алгоритм побудови компонентів РАФ у певних місцях на функціональній карті складається з двох етапів. По-перше, існує найближче з'єднання будь-якого типу, яке визначає одну з компонент вектора.

Далі визначте інший векторний компонент, який представляє асоціацію.

Другий суглоб не обов'язково найближчий, він може бути досить далеким.

Під час навчання живий компонент повинен вказувати на частину, яку потрібно зв'язати. Подібно до того, як x -компонент векторного поля завжди має вказувати на ту саму ціль, що й y -компонент, компоненти поля РҒ також повинні вказувати на те саме часткове об'єднання.

Втрата адаптивної регресії

Алгоритми оцінки пози людини повинні враховувати різні масштаби, які людські пози можуть мати на зображеннях. У той час як похибка спільної локалізації може бути малою для великої, та сама абсолютна похибка може бути основною похибкою для параметрів малої. Ми використовуємо втрату типу L1 для навчання та навчання результату регресії. Ми покращуємо здатність мережі до локалізації, вводячи залежність від масштабу в втрати регресії за допомогою SmoothL1 або втрати Лапласа [33].

Втрата SmoothL1 дозволяє регулювати радіус r навколо початку координат, створюючи більш плавний градієнт. Для розміру граничної області та ключової точки σ_k грані A_i , $r(i,k,smooth)$ можна визначити за формулою $SQTR(\sigma_k, r(i,k,smooth))$.

Лапласова втрата - це ще одна втрата типу L1, яка стає розрідженою через поширення гіпотези b :

$$L = \frac{|x-\mu|}{b+\log(2b)} \quad (4.4)$$

Це не залежить від жодних оцінок A_i та σ_k , які ми будемо використовувати для всіх компонент вектора.

жадібне декодування

Декодування - це процес перетворення вихідної карти функції нейронної мережі в набір із 17 координат, які оцінюють поставу людини. Нова поза пов'язана з вектором \mathbb{F} з найвищим значенням у довільній карті високої роздільної здатності $f(x, y)$. Далі додайте з'єднання з іншими з'єднаннями за допомогою поля RASN. Після встановлення з'єднання з новим з'єднанням рішення вважається остаточним.

Кілька асоціацій РАН можуть утворювати зв'язок між поточним суглобом і наступним суглобом. Враховуючи положення початкового суглоба $\sim x, s$ асоціації РАН а розраховуються таким чином:

$$s(a, \vec{x}) = a_c \exp\left(-\frac{\|\vec{x} - \vec{a}_1\|_2}{b_1}\right) f_2(a_{x2}, a_{y2}) \quad (4.5)$$

Враховано достовірність цього зв'язку a_c , відстань до положення першого вектора, відкаліброваного за лапласівським розподілом ймовірностей, і високу роздільну здатність частини другого вектора f_2 у цільовому положенні.

Щоб підтвердити запропоноване положення нових з'єднань, ми виконуємо процес зворотного зіставлення. Повторюйте цей процес, поки не досягнете повної пози. Ми застосовуємо немаксимальне стиснення на рівні ключових точок. Радіус стиснення є динамічним і базується на припущеному розмірі компонента IF. Ми не уточнюємо жодних полів під час навчання чи тестування.

Бібліотека, яка використовується для навчання: OpenCV

Розглянемо бібліотеку з відкритим кодом OpenCV [34-36]. Ця бібліотека в основному призначена для пошуку зображень у режимі реального часу. Широке використання інструменту пояснюється його відкритістю, крос-платформною сумісністю, великою кількістю літератури та навчальних посібників. Він написаний на C/C++ і його вихідний код відкритий. Бібліотека містить близько 1000 функцій і алгоритмів. З 1998 року Intel, Itseez та інші компанії розробляють його за активної участі спільноти. Бібліотеку завантажили понад 6 мільйонів разів, що свідчить про її популярність.

Бібліотека поширюється за ліцензією BSD [37]. Тобто його можна використовувати як у відкритих, так і в комерційних проектах. Вам не потрібно повністю переносити бібліотеку в свій проект, ви можете просто використовувати відповідні частини коду. Для безкоштовного використання ви повинні відповідати вимогам ліцензії - копія ліцензії OpenCV повинна бути вказана в супровідних матеріалах.

По суті, OpenCV - це бібліотека алгоритмів і функцій для обробки зображень і відео. Ось приклади доступних модулів:

- `opencv_video` - аналізує та відстежує об'єкти у відео, а також видаляє фон;

- opencv_gpu - модуль для підключення обчислень на GPU завдяки CUDA (Nvidia);
- opencv_imgproc - обробка зображень (фільтри, конвертації);
- opencv_ml - інструменти машинного навчання (SVM, дерева рішень).

З метою економії часу на навчання нейронної мережі було вирішено використовувати доступні навчальні зразки. COCO [38] (Common Objects in Context) [35-38] виводить 18 ключових точок (19-та точка є фоном).

Кожна така точка має свої координати на кожному кадрі зображення. З цього ви можете побудувати скелет для кожного кадру та реалізувати власні функції постобробки.

4.5 Експерименти та оцінка роботи системи

Камери в безпілотних автомобілях мають широке поле зору і повинні знаходити всіх пішоходів у цьому полі зору. Ми хочемо змодельовати розподіл пікселів по висоті пішоходів за допомогою загальнодоступних наборів даних і протоколів оцінки пози людини.

Крім того, щоб продемонструвати широку застосовність нашого методу, ми також вивчаємо оцінку пози в задачі повторної ідентифікації людини (Re-Id). Тобто отримати зображення людини та ідентифікувати цю особу на інших зображеннях.

Ми кількісно оцінюємо запропонований нами метод розпізнавання людини на зображеннях із низькою роздільною здатністю за допомогою набору даних COCO. Базуючись на оригінальному наборі даних COCO, ми обмежуємо максимальну довжину сторони зображення 321 пікселем, щоб імітувати кадрування зображень із камери 4к. Ми отримуємо обмежувальний діапазон 66 ± 65 пікселів у висоту. індекс

COCO [39] містить розподіл зображень людей середнього розміру в AP(M) і AR(M) із початковими площами обмежувальної рамки між $(32 \text{ px})^2$ і $(96 \text{ px})^2$. При зміні розміру для зображення з низькою роздільною здатністю це відповідає обмежувальній рамці висотою 44 ± 19 пікселів.

Ми проводимо поглиблене дослідження ефективності нашого методу на зображеннях, зроблених безпілотними автомобілями, і на випадкових сценах з багатьма людьми.

У контексті Re-Id ми досліджуємо популярний і загальнодоступний набір даних Market-1501. Він складається з пішохідного кадрування розміром 64×128 пікселів. Ми застосовуємо ту саму модель, навчену на даних COCO.

Завдання виявлення ключової точки COCO оцінюється як завдання виявлення об'єкта, а основними показниками є варіанти середньої точності (AP) і середнього запам'ятовування (AR). COCO припускає фіксоване співвідношення розміру ключової точки до обмежувальної площі поля для кожного типу ключової точки для визначення подібності ключових точок об'єкта (OKS). Для кожного зображення оцінювач пози повинен передбачити розташування 17 ключових точок для кожної пози та підрахувати кожну позу. Для оцінювання відбираються лише 20 із найкращими загальними балами.

Для еталонних значень Mask R-CNN [40] і OpenPose ми використовуємо реалізації на основі робіт [3, 7], модифіковані для забезпечення максимальної довжини зображення. Маски R-CNN модифіковано для використання з низькою роздільною здатністю зображення. Результати нашого методу отримані на основі ResNet50 [41-42].

Зокрема, ми використовуємо 64115 зображень із навчального набору COCO 2017 року, які анотовані для навчання. Наша перевірка виконується на наборі перевірки COCO 2017 (містить 5000 зображень). Базовою мережею є модифікована мережа ResNet50/101/152. Основна мережа - це один шар згортки 1×1 пікселів, що подвоює просторову роздільну здатність. Еталонна складова поля нормується S-подібною нелінійністю.

Додано лише деякі мінімальні дані. Щоб створити однакові групи, ми обрізаємо зображення у квадрат, де сторони квадрата становлять від 95% до 100% короткої сторони зображення, і місце вибрано випадково. Застосовано кадрування фотографій, щоб зберегти якомога більше даних навчання мережі. Усе зображення залишається необрізаним половину часу, а смуги додаються, щоб зробити його

квадратним. Подальша зміна розміру використовує бікубічну інтерполяцію. Навчальні зображення та нотатки довільно гортаються горизонтально.

Компоненти полів [43-44], які формують карту довіри, навчаються з використанням незалежних двійкових перехресних втрат ентропії. Ми використовуємо втрати L1 для масштабної складової поля CIR і лапласівські втрати для всіх векторних компонентів.

Під час навчання ми записуємо статистику запущених операцій пакетного забезпечення та їх попередньо перевірені значення. Ми використовуємо оптимізатор SGD зі швидкістю навчання 10⁻³, імпульсом 0,95, розміром партії 8 і без ваг. Ми використовуємо усереднення моделі, щоб отримати стабільні моделі для тестування. На кожному кроці оптимізації ми оновлюємо експоненціально зважені версії параметрів моделі. Наша константа розпаду дорівнює 10⁻³. Час навчання ResNet101 на двох GTX1080Ti становить близько 95 годин протягом 75 епох.

Ми порівнюємо наш запропонований метод із відтворюваним сучасним методом OpenPose та методом Mask R-CNN [45]. Хоча мета - вийти за межі методів

Знизу вгору, ми все ще аналізуємо результати підходу зверху вниз, щоб оцінити ефективність нашого підходу. Оскільки це імітація маленьких людей на великих зображеннях, ми змінили існуючий метод, щоб запобігти масштабуванню маленьких зображень.

Таблиця 4.1 показує наші кількісні результати з використанням набору даних COCO. Ми перевершуємо метод OpenPose в усіх аспектах, навіть метод Mask R-CNN. Ці цифри зазвичай нижчі, ніж відповідні цифри для зображень із вищою роздільною здатністю. Два концептуально різні базові методи демонструють однакову ефективність, тоді як наш метод значно перевершує більш ніж на 18% у AP [46].

Таблиця 4.1 - Застосування методу оцінки пози для зображень низької роздільної здатності з довгою стороною, що дорівнює 321 пікселів

	AP	$AP^{0.50}$	$AP^{0.75}$	AP^M	AP^L	AR	$AR^{0.50}$	$AR^{0.75}$	AR^M	AP^L
Mask R-CNN	42.6	67.1	41.5	27.2	58.9	48.1	75.0	49.9	36.7	66.5
OpenPose	38.6	61.5	36.1	24.1	55.7	44.0	64.8	45.7	25.4	67.5
Наш метод	48.3	74.1	51.7	36.1	68.6	56.0	76.4	57.9	38.9	75.7

Наші кількісні результати моделюють розподіл людей у міських вуличних сценах за допомогою загальнодоступних анотованих наборів даних. Використовуючи наш підхід, можна досягти мінімальної індикації місця виявлення помилки. Рисунок 4.7 показує якісні результати для вуличної сцени. Ми також виявляємо пішоходів, які частково перекривають один одного. Наш метод може виявляти лише ключові жести, такі як «махання» машині. Ні Mask-RCNN, ні OpenPose точно не оцінюють жести на зображеннях. Ця відмінність має вирішальне значення для розробки безпечних самокерованих автомобілів.



Рисунок 4.7. Візуалізація нашого алгоритму (праворуч) проти OpenPose [3] (перша колонка) на наборі даних nuScenes

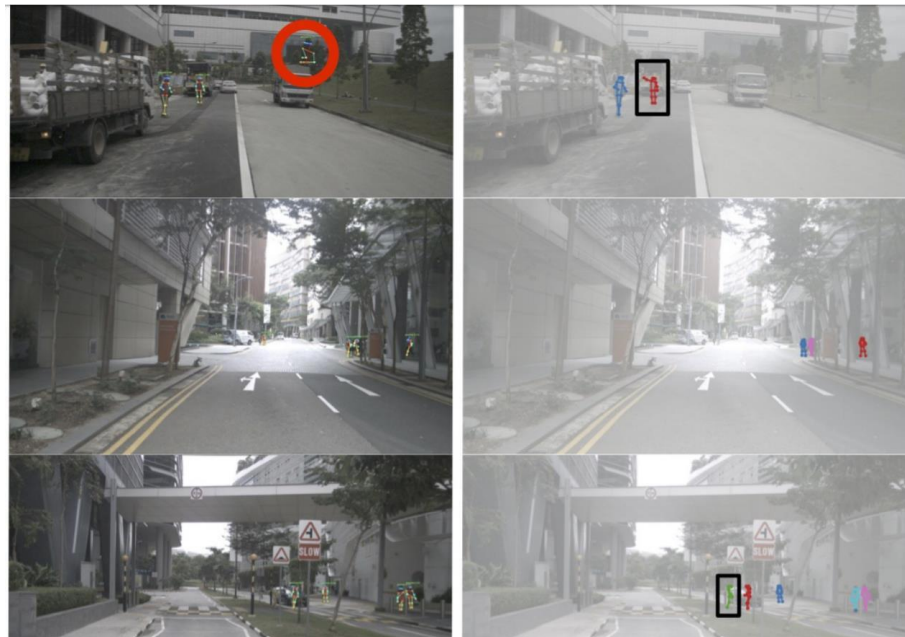


Рисунок 4.8 - Візуалізація нашого алгоритму (справа) проти Mask RCNN [4] (перший стовпець) на наборі даних nuScenes

Система виділяє прямокутниками всіх людей, не розпізнаних іншими методами, а кружечками - усі неправильно розпізнані жести. Зауважте, що наш метод правильно оцінює жест «махання» людини (перший ряд, перший прямокутник), тоді як інші методи цього не роблять.

Щоб кількісно оцінити ефективність набору даних Market-1501, ми створили спрощену метрику точності. Точність Mask R-CNN [47-48] становить 43%, а нашого методу - 96%. Оцінка базується на кількості правильно розпізнаних поз із 202 випадкових зображень. До трьох суглобів розміщуються в правильному положенні.

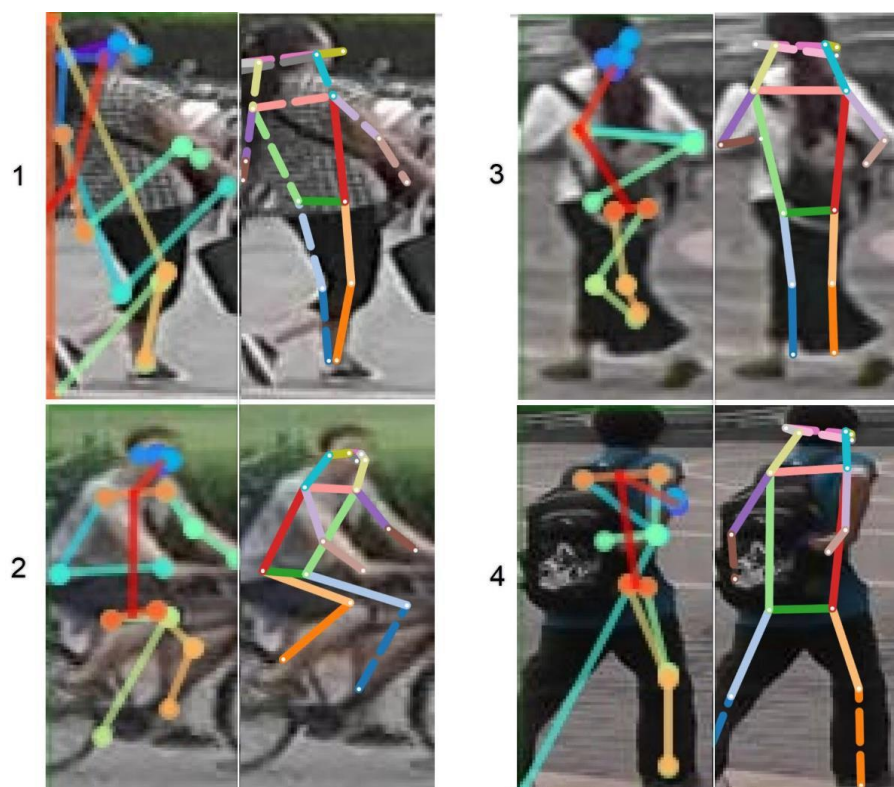


Рисунок 4.9 - Зображення ліворуч на кожному із 4 прикладів - вихід із Mask

R-

CNN.

Щоб покращити результати Mask R-CNN, ми встановили параметри так, щоб метод міг точно передбачити позу в межах обмеженого діапазону, який охоплює все зображення.

Правильне зображення є результатом роботи нашої системи, яка не обмежена однією людиною і може вибрати для виведення одну або кілька поз, що є більш складним завданням (рис. 4.9).

Таблиця 4.2 - Показники у відсотках, оцінені на датасеті COCO 2017, встановлені при оптимальній роздільній здатності

	AP	AP^M	AP^L
Mask R-CNN	62.1	58.1	70.6
OpenPose	60.9	56.8	67.1
PersonLab	64.4	62.3	72.1
Наш метод	66.5	62.7	72.9

Інші методи оптимізовані для отримання зображень з вищою роздільною здатністю. Для порівняння ми провели кількісне порівняння на наборі даних COCO 2017 з високою роздільною здатністю, наведеному в таблиці 4.2. З результатів можна побачити, що продуктивність цієї системи порівнянна з найкращими існуючими методами «знизу вгору» [49].

Таблиця 4.3 - Вивчення залежності від типу втрати L1

	AP	AP^M	AP^L
Vanilla L 1	36.5	22.9	55.6
SmthL1, $r = 0,2\sqrt{A_i}\sigma_k$	36.7	23.4	55.5
SmthL1, $r = 0,5\sqrt{A_i}\sigma_k$	36.2	22.8	55.2
SmthL1, $r = 1,0\sqrt{A_i}\sigma_k$	34.9	21.8	54.6
Laplace	41.8	29.3	57.4
Laplace (використовуючи b в декодері)	43.0	28.9	60.1

Показники в таблиці 4.3 представлені у відсотках. Усі моделі засновані на ResNet-50 і навчені на 10 епох.

Ми вивчаємо залежність від типу втрати $L1$, зведену в таблицю 4. Ми виявили, що можемо налаштувати продуктивність для менших або більших об'єктів, змінивши загальну шкалу r -smooth, тому ми проаналізували вплив цього параметра. Однак реальне покращення відбувається завдяки використанню лапласівських втрат. Додавання компонента масштабу σ до поля IF покращує AP нашої моделі ResNet101 з 65% до 66,1%.

Таблиця 4.4 - COCO val.set [50] Співвідношення між точністю розпізнавання різних шарів ResNet у наборі даних і часом виведення t одного зображення.

	AP [%]	t [мс]	t(dec) [мс]
ResNet50	61.8	237	191
ResNet101	66.1	251	186
ResNet152	67,9	276	185

Останній стовпець - це час декодування $t(\text{dec})$. Продуктивність різних списків ResNet показано в таблиці 4. На цій же основі ми випередили PersonLab на 10,0% у AP, але були на 29% швидшими. Дані для PersonLab на ResNet101 становлять 60,0% і 355 мс.

Удосконалений метод «знизу вгору» розроблено для двовимірної оцінки пози людини на зображеннях кількох осіб, який можна ефективно застосувати в галузі транспорту, особливо автономних транспортних засобів і соціальних роботів. Результати показують, що розроблена система перевершує попередні методи в режимі низької роздільної здатності та порівнюється з існуючими методами в режимі високої роздільної здатності.

5 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічна розробка має право на існування та впровадження, якщо вона відповідає вимогам часу, як в напрямку науково-технічного прогресу та і в плані економіки. Тому для науково-дослідної роботи необхідно оцінювати економічну ефективність результатів виконаної роботи.

Магістерська кваліфікаційна робота з розробки та дослідження «Інтелектуальна система автоматизованого детектування пози оператора» відноситься до науково-технічних робіт, які орієнтовані на виведення на ринок (або рішення про виведення науково-технічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науково-технічної розробки. Цей напрямок є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для наведеного випадку нами мають бути виконані такі етапи робіт:

- 1) проведено комерційний аудит науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- 2) розраховано витрати на здійснення науково-технічної розробки;
- 3) розрахована економічна ефективність науково-технічної розробки у випадку її впровадження і комерціалізації потенційним інвестором і проведено обґрунтування економічної доцільності комерціалізації потенційним інвестором.

5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інтелектуальна система автоматизованого детектування пози оператора» є

оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в табл. 5.1 [51].

Таблиця 5.1 - Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту на	Технічні та споживчі властивості продукту	Технічні та споживчі властивості продукту
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					

8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела	Потрібні незначні фінансові ресурси. Джерела фінансування	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовують
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці.

Таблиця 5.2 - Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	4	3	4

2. Ринкові переваги (наявність аналогів)	3	3	2
3. Ринкові переваги (ціна продукту)	3	2	3
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	2	2	2
6. Ринкові перспективи (розмір ринку)	2	2	2
7. Ринкові перспективи (конкуренція)	2	2	2
8. Практична здійсненність (наявність фахівців)	4	4	4
9. Практична здійсненність (наявність фінансів)	2	3	2
10. Практична здійсненність (необхідність нових матеріалів)	2	2	2
11. Практична здійсненність (термін реалізації)	3	4	4
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	33	34	34
Середньоарифметична сума балів $СБ_c$	33,7		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в табл. 5.3

Таблиця 5.3 - Науково-технічні рівні та комерційні потенціали розробки [52]

Середньоарифметична сума балів $СБ_c$, розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» становить 33,7 бала, що, відповідно до таблиці 4.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

5.2 Розрахунок узагальненого коефіцієнта якості розробки

Окрім комерційного аудиту розробки доцільно також розглянути технічний рівень якості розробки, розглянувши її основні технічні показники. Ці показники по-різному впливають на загальну якість проектної розробки.

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення розрахуємо за формулою [40]:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i, \quad (5.1)$$

де k - кількість найбільш важливих технічних показників, які впливають на якість нового технічного рішення;

α_i - коефіцієнт, який враховує питому вагу i -го технічного показника в загальній якості розробки. Коефіцієнт α_i визначається експертним шляхом і

при цьому має виконуватись умова $\sum_{i=1}^k \alpha_i = 1$;

β_i - відносне значення i -го технічного показника якості нової розробки.

Відносні значення β_i для різних випадків розраховуємо за такими формулами:

- для показників, зростання яких вказує на підвищення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ni}}{I_{ai}}, \quad (5.2)$$

де I_{ni} та I_{ai} - чисельні значення конкретного i -го технічного показника якості відповідно для нової розробки та аналога;

- для показників, зростання яких вказує на погіршення в лінійній залежності якості нової розробки:

$$\beta_i = \frac{I_{ai}}{I_{ni}}; \quad (5.3)$$

Використовуючи наведені залежності можемо проаналізувати та порівняти техніко-економічні характеристики аналогу та розробки на основі отриманих наявних та проектних показників, а результати порівняння зведемо до таблиці 5.4.

Таблиця 5.4 - Порівняння основних параметрів розробки та аналога.

Показники (параметри)	Одиниця вимірювання	Аналог	Проектований пристрій	Відношення параметрів нової розробки до аналога	Питома вага показника
Вимоги до оперативної пам'яті	GB	4	2	2	0,2
Рекомендоване місце на диску	GB	6	3	2	0,3
Точність розпізнавання	%	87	94	1,1	0,15
Вимоги до CPU	бал	7	8	0,87	0,25
Захищеність БД від зовнішнього впливу (до 10 балів)	бал	7	7	1	0,1

Узагальнений коефіцієнт якості (B_n) для нового технічного рішення складе:

$$B_n = \sum_{i=1}^k \alpha_i \cdot \beta_i = 2 \cdot 0,2 + 2 \cdot 0,3 + 1,1 \cdot 0,15 + 0,87 \cdot 0,25 + 1 \cdot 0,1 = 1,48.$$

Отже за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,48 рази.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інтелектуальна система автоматизованого детектування пози оператора», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.3.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам,

креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.4)$$

де k - кількість посад дослідників залучених до процесу досліджень;

M_{ni} - місячний посадовий оклад конкретного дослідника, грн;

t_i - число днів роботи конкретного дослідника, дн.;

T_p - середнє число робочих днів в місяці, $T_p=22$ дні.

$$Z_o = 17640,00 \cdot 34 / 22 = 27261,82 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.5 - Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник проекту	17640,00	801,82	34	27261,82
Інженер-розробник програмного забезпечення	16830,00	765,00	34	26010,00
Консультант (провідний фахівець з дослідження біометричних характеристик людини)	16000,00	727,27	10	7272,73
Лаборант	6900,00	313,64	10	3136,36
Всього				63680,91

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.5)$$

де C_i - погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i - час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.6)$$

де M_M - розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), прийmemo $M_M=6700,00$ грн;

K_i - коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_c - мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p - середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ - тривалість зміни, год.

$$C_1 = 6700,00 \cdot 1,10 \cdot 1,35 / (22 \cdot 8) = 56,53 \text{ грн.}$$

$$Z_{pl} = 56,53 \cdot 7,30 = 412,68 \text{ грн.}$$

Таблиця 5.6 - Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника, грн
Установка обладнання для розробки програмного забезпечення	7,30	2	1,10	56,53	412,68
Підготовка робочого місця розробника	6,10	3	1,35	69,38	423,21

програмного забезпечення					
Інсталяція програмного забезпечення для розпізнавання зображення	5,65	5	1,70	87,37	493,62
Компіляція програмних блоків	4,00	4	1,50	77,09	308,35
Налагодження програмних блоків	4,30	5	1,70	87,37	375,68
Формування бази даних нейромережі	12,00	4	1,50	77,09	925,06
Попереднє тренування нейромережі	8,00	3	1,35	69,38	555,03
Всього					3493,63

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.7)$$

де $H_{\text{дод}}$ - норма нарахування додаткової заробітної плати. Прийmemo 12%.

$$Z_{\text{дод}} = (63680,91 + 3493,63) \cdot 12 / 100\% = 8060,94 \text{ грн.}$$

5.3.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{H_{\text{зн}}}{100\%} \quad (5.8)$$

де $H_{\text{зн}}$ - норма нарахування на заробітну плату. Приймаємо 22%.

$$Z_n = (63680,91 + 3493,63 + 8060,94) \cdot 22 / 100\% = 16551,81 \text{ грн.}$$

5.3.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які

придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж».

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{ej}, \quad (5.9)$$

де H_j - норма витрат матеріалу j -го найменування, кг;

n - кількість видів матеріалів;

C_j - вартість матеріалу j -го найменування, грн/кг;

K_j - коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j - маса відходів j -го найменування, кг;

C_{ej} - вартість відходів j -го найменування, грн/кг.

$$M_1 = 4,0 \cdot 195,00 \cdot 1,03 - 0 \cdot 0 = 803,40 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.7 - Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Офісний папір Ultra Plus	195,00	4,0	0	0	803,40
Папір для записів Papers Light A5	114,00	3,0	0	0	352,26
Органайзер офісний OFFICE	194,00	3,0	0	0	599,46
Канцелярське приладдя (набір офісного працівника)	172,00	4,0	0	0	708,64
Картридж для принтера Erixon EZ2500	920,00	2,0	0	0	1895,20

Диск оптичний New CD-RW	28,00	3,0	0	0	86,52
Flesh-пам'ять King 64 GB	144,00	1,0	0	0	148,32
Тека для паперів BOX	105,00	3,0	0	0	324,45
Всього					4918,25

5.3.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» відсутні.

5.3.5 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{\text{спец}} = \sum_{i=1}^k C_i \cdot C_{\text{пр.}i} \cdot K_i, \quad (5.10)$$

де C_i - ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{\text{пр.}i}$ - кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i - коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1, 10 \dots 1, 12$);

k - кількість найменувань устаткування.

$$B_{\text{спец}} = 15430,00 \cdot 1 \cdot 1,03 = 15892,90 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.8 - Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
---------------------------	------------------	-------------------------	------------------

Мультимедійний проектор SAMSUNG PLC-XW300	1	15430,00	15892,90
Біометричний сканер	1	25699,00	26469,97
Всього			42362,87

5.3.6 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{\text{прог}} = \sum_{i=1}^k C_{\text{инрг}} \cdot C_{\text{прог.і}} \cdot K_i, \quad (5.11)$$

де $C_{\text{инрг}}$ - ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прог.і}}$ - кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i - коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k - кількість найменувань програмних засобів.

$$B_{\text{прог}} = 8410,00 \cdot 1 \cdot 1,03 = 8662,30 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.9 - Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows	1	8410,00	8662,30
Прикладний пакет Microsoft Office	1	7770,00	8003,10
Середовище розробки Visual Studio	1	11320,00	11659,60
Згорткова нейронна мережа	1	13400,00	13802,00
Програмне забезпечення програмування C#	1	5100,00	5253,00
Всього			47380,00

5.3.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{обл} = \frac{Ц_{б}}{T_{г}} \cdot \frac{t_{вик}}{12}, \quad (5.12)$$

де $Ц_{б}$ - балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$ - термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{г}$ - строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (26799,00 \cdot 2) / (3 \cdot 12) = 1488,83 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.10 - Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер розробника програмного забезпечення	26799,00	3	2	1488,83
Робоче місце інженера-програміста	8250,00	3	2	458,33
Пристрої передачі даних	6540,00	6	2	181,67
Оргтехніка	7850,00	5	2	261,67
Приміщення лабораторії розробки інформаційних систем	426000,00	25	2	2840,00

Електронно-обчислювальний комплекс обробки зображень на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642	75900,00	3	2	4216,67
Всього				9447,17

5.3.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{\text{ени}}}{\eta_i}, \quad (5.13)$$

де W_{yi} - встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i - тривалість роботи обладнання на етапі дослідження, год;

C_e - вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 7,50$ грн;

$K_{\text{ени}}$ - коефіцієнт, що враховує використання потужності, $K_{\text{ени}} < 1$;

η_i - коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,22 \cdot 240,0 \cdot 7,50 \cdot 0,95 / 0,97 = 396,00 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.11 - Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер розробника програмного забезпечення	0,22	240,0	396,00
Робоче місце інженера-програміста	0,10	240,0	180,00
Пристрої передачі даних	0,03	240,0	54,00
Оргтехніка	0,65	1,4	6,83
Електронно-обчислювальний комплекс обробки	0,40	240,0	720,00

зображень на базі QUBE QB i9 10900KF RTX 3060 TI 8GB 1642			
Мультимедіа проектор SAMSUNG PLC-XW300	0,20	12,0	18,00
Всього			1374,83

5.3.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (5.14)$$

де H_{cv} - норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (63680,91 + 3493,63) \cdot 20 / 100\% = 13434,91 \text{ грн.}$$

5.3.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.15)$$

де H_{cn} - норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 30\%$.

$$B_{cn} = (63680,91 + 3493,63) \cdot 30 / 100\% = 20152,36 \text{ грн.}$$

5.3.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\epsilon} = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (5.16)$$

де H_{ib} - норма нарахування за статтею «Інші витрати», прийmemo $H_{ib} = 50\%$.

$$I_{\epsilon} = (63680,91 + 3493,63) \cdot 50 / 100\% = 33587,27 \text{ грн.}$$

5.3.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.17)$$

де $H_{нзв}$ - норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{нзв} = 105\%$.

$$B_{нзв} = (63680,91 + 3493,63) \cdot 105 / 100\% = 70533,27 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{заг} = Z_o + Z_p + Z_{доп} + Z_n + M + K_{\epsilon} + B_{спец} + B_{прг} + A_{обл} + B_{\epsilon} + B_{св} + B_{сп} + I_{\epsilon} + B_{нзв}. \quad (4.18)$$

$$B_{заг} = 63680,91 + 3493,63 + 8060,94 + 16551,81 + 4918,25 + 0,00 + 42362,87 + 47380,00 + 9447,17 + 1374,83 + 13434,91 + 20152,36 + 33587,27 + 70533,27 = 334978,21 \text{ грн.}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ZB = \frac{B_{заг}}{\eta}, \quad (5.19)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta=0,9$.

$$ZB = 334978,21 / 0,9 = 372198,01 \text{ грн.}$$

5.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

Результати дослідження проведені за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» передбачають комерціалізацію протягом 4-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

ΔN - збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	600	1200	1500	800

N - кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 11000 осіб;

C_o - вартість програмного продукту у році до впровадження результатів розробки, прийmemo 7999,00 грн;

$\pm\Delta C_o$ - зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 359,52 грн.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.20)$$

де λ - коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ - коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 37\%$;

ϑ - ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (359,52 \cdot 11000,00 + 8358,52 \cdot 600) \cdot 0,83 \cdot 0,37 \cdot (1 - 0,18/100\%) = 2258793,73 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (359,52 \cdot 11000,00 + 8358,52 \cdot 1800) \cdot 0,83 \cdot 0,37 \cdot (1 - 0,18/100\%) = 4784624,04 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (359,52 \cdot 11000,00 + 8358,52 \cdot 3300) \cdot 0,83 \cdot 0,37 \cdot (1 - 0,18/100\%) = 7941911,93 \text{ грн.}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (359,52 \cdot 11000,00 + 8358,52 \cdot 4100) \cdot 0,83 \cdot 0,37 \cdot (1 - 0,18/100\%) = 9625798,81 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^i}, \quad (5.21)$$

де $\Delta\Pi_i$ - збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T - період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ - ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau=0,15$;

t - період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} ПП &= 2258793,73/(1+0,15)^1 + 4784624,04/(1+0,15)^2 + 7941911,93/(1+0,15)^3 + \\ &+ 9625798,81/(1+0,15)^4 = 1964168,46 + 3617863,17 + 5221936,01 + 5503581,71 = \\ &= 16307549,36 \text{ грн.} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки:

$$PV = k_{инв} \cdot 3B, \quad (5.22)$$

де $k_{инв}$ - коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв}=2$;

$3B$ - загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 372198,01 грн.

$$PV = k_{инв} \cdot 3B = 2 \cdot 372198,01 = 744396,03 \text{ грн.}$$

Абсолютний економічний ефект E_{abc} для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = III - PV \quad (5.23)$$

де III - приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 16307549,36 грн;

PV - теперішня вартість початкових інвестицій, 744396,03 грн.

$$E_{abc} = III - PV = 16307549,36 - 744396,03 = 15563153,33 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій E_e , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.24)$$

де E_{abc} - абсолютний економічний ефект вкладених інвестицій, 15563153,33 грн;

PV - теперішня вартість початкових інвестицій, 744396,03 грн;

$T_{ж}$ - життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримування позитивних результатів від її впровадження, 4 роки.

$$E_e = T_{ж} \sqrt[4]{1 + \frac{E_{abc}}{PV}} - 1 = (1 + 15563153,33 / 744396,03)^{1/4} = 1,16.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{min} :

$$\tau_{min} = d + f, \quad (5.25)$$

де d - середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,1$;

f - показник, що характеризує ризикованість вкладення інвестицій, приймемо 0,35.

$\tau_{min} = 0,1 + 0,35 = 0,45 < 1,16$ свідчить про те, що внутрішня економічна дохідність інвестицій E_g , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» доцільно.

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_g}, \quad (5.26)$$

де E_g - внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 1,16 = 0,86 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Висновки до розділу

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж» становить 33,7 бала, що, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

При оцінюванні за технічними параметрами, згідно узагальненого коефіцієнту якості розробки, науково-технічна розробка переважає існуючі аналоги приблизно в 1,48 рази.

Також термін окупності становить 0,86 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Методи розпізнавання жестів людини за допомогою згорткових нейронних мереж».

ВИСНОВКИ

В магістерській кваліфікаційній роботі розроблено систему розпізнавання жестів людини. Проаналізовано існуючі методи розпізнавання жестів і проекти, технічно близькі до них. Було вирішено ідентифікувати скелетовані дані, попередньо оброблені алгоритмом морфологічної трансформації, і використати архітектуру нейронної мережі CNN.

Відповідно до вимог системи розроблено основні класи та компоненти системи та встановлюються зв'язки між ними. Детально розглянуто реалізацію окремих прецедентів.

Під час розробки системи було досягнуто наступних результатів:

- проведено аналіз аналогів програмного забезпечення та наукової літератури з предметної області;
- розроблено алгоритми попередньої обробки даних;
- створено навчальні і тестові зразки;
- виконано навчання та тестування моделі нейронних мереж для розпізнавання жестів людини в реальному часі.

Удосконалений метод «знизу вгору» розроблено для двовимірної оцінки пози людини на зображеннях кількох осіб, який можна ефективно застосувати в галузі транспорту, особливо автономних транспортних засобів і соціальних роботів. Результати показують, що розроблена система перевершує попередні методи в режимі низької роздільної здатності та порівнюється з існуючими методами в режимі високої роздільної здатності.

У майбутньому планується робота над підвищенням точності розпізнавання. Для цього необхідно істотно розширити навчальні вибірки і, можливо, відкоригувати топологію нейронної мережі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Poselet conditioned pictorial structures / L. Pishchulin [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2013. - С. 588-595
2. Toshev A., Szegedy C. Deeppose: Human pose estimation via deep neural networks// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2014. - С. 1653-1660.
3. Zhe Cao, Tomas Simon, Shih-En Wei Yaser, Sheikh The Robotics Institute, Carnegie Mellon, University: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields // - Режим доступа: <https://arxiv.org/pdf/1611.08050.pdf>
4. L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4929-4937, 2016
5. Du Y., Wang W., Wang L. Hierarchical recurrent neural network for skeleton based action recognition. // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015. - Vol. 7. - No. 12. - P. 1110-1118.
6. H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in ICCV, 2017.
7. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in ICCV, 2017.
8. Girshick R. Fast R-CNN. Режим доступа: <https://arxiv.org/pdf/1504.08083>
9. Козлов В.А., Потапов А.С. Анализ методов выделения движущихся объектов на видеопоследовательностях с шумами. // Научнотехнический вестник информационных технологий, механики и оптики. Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики», 2011. - Т. 3. - № 73. - С. 39-43.
10. Lucas B.D., Kanade T. An iterative image registration technique with an application to stereo vision. // Proc. 7th Int. Jt. Conference on Artificial Intelligence, 1981. - Vol. 2. - P. 674-679.

11. Horn B.K.P., Schunck B.G. Determining optical flow. // *Artif. Intell.* 44 Elsevier, 1981. - Vol. 17. - No. 1-3. - P. 185-203.

12. Bruhn A., Weickert J., Schnörr C. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. // *International Journal of Computer Vision*, Kluwer Academic Publishers, 2005. - Vol. 61. - No. 3. - P. 1-21.

13. Отслеживание движения и алгоритмы сопровождения ключевых точек. НОУ Интуит. Режим доступа: www.intuit.ru/studies/courses/10622/1106/lecture/18022

14. Abeysinghe S.S. Segmentation-free skeletonization of grayscale volumes for shape understanding. S.S. Abeysinghe, M. Baker, W. Chiu, T. Ju. // *IEEE International Conference on Shape Modeling and Applications*, 2008. - P. 63-71.

15. Крашенникова Ю.С. Использование процедуры «скелетизации» для выделения линий на спутниковых изображениях. Ю.С. Крашенникова, Е.А. Лупян, Т.А. Немченко, М.Ю. Захаров. // *Исследование Земли из космоса*, 1994. - Т. 6. - С. 43- 51.

16. Применение волнового алгоритма для нахождения скелета растрового изображения. Распознавание образов и искусственный интеллект. Режим доступа: ocrai.narod.ru/vectory.html

17. McCulloch W.S., Pitts W. A Logical Calculus of the Ideas Immanent in Nervous Activity. *The Bulletin of Mathematical Biophysics*. 1943. vol. 5, no. 4. pp. 115-133. DOI: 10.1007/BF02478259

18. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, 1(4):541-551, Winter 1989

19. Habibi, Aghdam, Hamed. Guide to convolutional neural networks: a practical application to traffic-sign detection and classification. Heravi, Elnaz Jahani, Cham, Switzerland.

20. Bing Xu, Naiyan Wang, Tianqi Chen, Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network (2015).

21. Backpropagation In Convolutional Neural Networks Режим доступа: <http://www.jefkine.com/general/2016/09/05/backpropagation-inconvolutional-neural-networks/>
22. Felzenszwalb P., McAllester D., Ramanan D. A discriminatively trained, multiscale, deformable part model // Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. - IEEE. 2008. - С. 1-8.
23. Yang Y., Ramanan D. Articulated pose estimation with flexible mixtures-of-parts // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. - IEEE. 2011. - С. 1385-1392.
24. Parsing occluded people / G. Ghiasi [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2014. - С. 2401-2408.
25. Modeling Instance Appearance for Recognition-Can We Do Better Than EM / A. Chou [и др.] // International Workshop on Structured Prediction: Tractability, Learning, and Inference. - 2013.
26. Yang Y., Ramanan D. Articulated pose estimation with flexible mixtures-of-parts // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. - IEEE. 2011. - С. 1385-1392.
27. Park D., Ramanan D. N-best maximal decoders for part models // 2011 International Conference on Computer Vision. - IEEE. 2011. - С. 2627-2634
28. Toshev A., Szegedy C. DeepPose: Human pose estimation via deep neural networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2014. - С. 1653-1660.
29. Bulat A., Tzimiropoulos G. Human pose estimation via convolutional part heatmap regression // European Conference on Computer Vision. - Springer. 2016. - С. 717-732.
30. YOLO: Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/COCO> 2018 Keypoint Detection Task // Режим доступа: <http://cocodataset.org/#keypoints-2018>
31. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition Режим доступа: <https://arxiv.org/abs/1512.03385>

32. Reddit MachineLearning: How does DenseNet compare to ResNet and Inception? Режим доступу: https://www.reddit.com/r/MachineLearning/comments/67fds7/d_how_does_densenet_compare_to_resnet_and/
33. A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In Advances in neural information processing systems, pages 5574-5584, 2017. 2, 4
34. Офіційний ресурс бібліотеки OpenCV // Режим доступу: <https://opencv.org/>
35. COCO 2018 Keypoint Detection Task // Режим доступу: <http://cocodataset.org/#keypoints-2018>
36. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ - Режим доступу: http://kafpe.kpi.ua/wp-content/uploads/2015/04/roz_startap_proektiv_met_vk.pdf
37. Лотоцька Ю.В., Халус О.А. Аналіз методів автоматичного реферування тексту за допомогою нейронних мереж // II Всеукраїнська науково-практична конференція молодих вчених та студентів - 2019 - с. 9 - 12
38. Лотоцька Ю.В., Халус О.А. Розпізнавання пози людини у реальному часі // III всеукраїнська науково-практична конференція молодих вчених та студентів - 2019
39. Yan, B.; Mei, L. Design of intelligent invigilator system based on artificial vision. J. Phys. Conf. Ser. 2021, 1881, 042054.
40. Fang, Y.; Ye, J.; Wang, H. Realization of Intelligent Invigilation System Based on Adaptive Threshold. In Proceedings of the 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 15-18 May 2020; pp. 201-205.
41. Yepuri, V.K.; Pamu, G.C.; Kodali, N.; Pradyмна, L.V. Examination Management Automation System. Int. Res. J. Eng. Technol. 2018, 5, 2773-2779.
42. Xu, T.; Xiao, H. Application of SSD core detection algorithm in intelligent visual monitoring of examination room. J. Phys. Conf. Ser. 2021, 2037, 012085.
43. Malhotra, M.; Chhabra, I. Automatic Invigilation Using Computer Vision. In Proceedings of the 3rd International Conference on Integrated Intelligent Computing

Communication & Security (ICIIC 2021), Bangalore, India, 6-7 August 2021; pp. 130-136.

44. Ben Jabra, M.; Koubaa, A.; Benjdira, B.; Ammar, A.; Hamam, H. COVID-19 Diagnosis in Chest X-rays Using Deep Learning and Majority Voting. *Appl. Sci.* 2021, 11, 2884

45. Ahmad, I.; Ullah, I.; Khan, W.U.; Rehman, A.U.; Adrees, M.S.; Saleem, M.Q.; Cheikhrouhou, O.; Hamam, H.; Shafiq, M. Efficient Algorithms for E-Healthcare to Solve Multiobject Fuse Detection Problem. *J. Health Eng.* 2021, 2021, 9500304.

46. Adil, M.; Simon, R.; Khatri, S.K. Automated Invigilation System for Detection of Suspicious Activities during Examination. In *Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates, 4-6 February 2019; pp. 361-366.

47. Ranieri, C.M.; Vargas, P.A.; Romero, R.A.F. Uncovering Human Multimodal Activity Recognition with a Deep Learning Approach. In *Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, 19-24 July 2020; pp. 1-8.

48. Pandey, I.R.; Raj, M.; Sah, K.K.; Mathew, T.; Padmini, M.S. Face Recognition Using Machine Learning. *Int. Res. J. Eng. Technol.* 2019, 6, 3772-3776.

49. Jiang, H.; Learned-Miller, E. Face detection with the faster R-CNN. In *Proceedings of the 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, Washington, DC, USA, 30 May-3 June 2017; pp. 650-657.

50. Besnassi, M.; Neggaz, N.; Benyettou, A. Face detection based on evolutionary Haar filter. *Pattern Anal. Appl.* 2020, 23, 309-330.

51. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. - Вінниця : ВНТУ, 2021. - 42 с.

52. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепка - Вінниця : ВНТУ, 2016. - 113 с.

ДОДАТКИ

Додаток А

**ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**

Назва роботи: « Інтелектуальна система автоматизованого детектування пози оператора»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

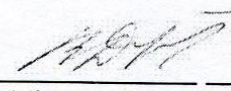
Оригінальність 89,4% Схожість 10,6%

Аналіз звіту подібності (відмітити потрібне)


✓ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

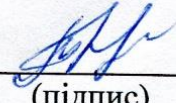
Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Володимир ДУБОВОЙ
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

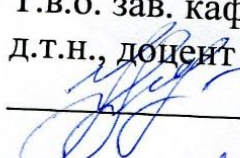
Автор роботи  Дмитро ЧЕРНАТА
(підпис) (прізвище, ініціали)

Керівник роботи  Тетяна ГРИЦУК
(підпис) (прізвище, ініціали)

Додаток Б

ВНТУ

ЗАТВЕРДЖЕНО


Т.в.о. зав. кафедри КСУ ВНТУ,
д.т.н., доцент
Марія ЮХИМЧУК“06” листопада 2023 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

Інтелектуальна система автоматизованого детектування пози оператора

08-33.МКР.018.00.000 ТЗ

Студент групи 2АКІТ-22м
ПідписДмитро ЧЕРНАТА

Ім'я ПРІЗВИЩЕ

Керівник


Підписдоц. Кадр. КСУТетяна ГРИЩУК

Ім'я ПРІЗВИЩЕ

Вінниця 2023

1. Назва та галузь застосування

1.1. Назва - Інтелектуальна система автоматизованого детектування пози оператора.

1.2. Галузь застосування - детектування пози оператора.

2. Підстава для проведення розробки.

Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від №247 від 18-09-2023р.

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є поліпшення виявлення та розуміння поведінки людей автомобільними бортовими комп'ютерами за рахунок удосконалення методу розпізнавання пози людини у режимі реального часу.

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Poselet conditioned pictorial structures / L. Pishchulin [и др.] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2013.
2. Toshev A., Szegedy C. Deeppose: Human pose estimation via deep neural networks// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. - 2014.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- провести аналіз існуючих алгоритмів та програмних аналогів у предметній області;
- реалізувати алгоритм попередньої обробки вхідних даних для подачі на нейронну мережу;
- розробити систему розпізнавання пози людини у відеопотоці у режимі реального часу;
- провести дослідження ефективності розроблених алгоритмів та порівняти із існуючими аналогами на обраному наборі даних.

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- Windows, macOS або Linux.;
- Мова програмування Python;
- бібліотека OpenCV.

5.2.2. Умови експлуатації системи:

- забезпечення доступу до камер;
- робота на мобільних та веб-додатках;
- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз методів, принципів, підходів і засобів реалізації задачі автоматизації процесами в об'єкті управління відповідно до теми кваліфікаційної роботи. Постановка задач дослідження «03»_09__ 2023 р.
2. Постановка цілей та задач дослідження «_12_»_09___ 20_23_ р.
3. Математичний опис роботи нейронної мережі «_04_»_10___ 20_23_ р.
4. Розробка програмного забезпечення системи на мові Python «_19_»_10___ 20_23_ р.

6.2 Графічні матеріали:

1. Розробка графіків нейронної мережі « 28 » 09 20 23 р.
2. Розробка схем нейронної мережі « 14 » 10 20 23 р.
3. Тестування програмного забезпечення « 21 » 10 20 23 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «28» 11 2023 р.
- 7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «14» 12 2023 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «20» 12 2023 р.

Додаток В

ЛІСТИНГ ОСНОВНОГО МОДУЛЯ

```
# Using conda
conda install -c conda-forge opencv# Using pip
pip install mediapipe
import cv2
import mediapipe as mp
import time

class PoseDetector:

    def __init__(self, mode = False, upBody = False, smooth=True, detectionCon = 0.5, trackCon
= 0.5):

        self.mode = mode
        self.upBody = upBody
        self.smooth = smooth
        self.detectionCon = detectionCon
        self.trackCon = trackCon

        self.mpDraw = mp.solutions.drawing_utils
        self.mpPose = mp.solutions.pose
        self.pose = self.mpPose.Pose(self.mode, self.upBody, self.smooth, self.detectionCon,
self.trackCon)

    def findPose(self, img, draw=True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.pose.process(imgRGB)
        if self.results.pose_landmarks:
```

```

        if draw:
            self.mpDraw.draw_landmarks(img, self.results.pose_landmarks,
self.mpPose.POSE_CONNECTIONS)

    return img, self.results.pose_landmarks, self.mpPose.POSE_CONNECTIONS

def getPosition(self, img, draw=True):
    lmList= []
    if self.results.pose_landmarks:
        for id, lm in enumerate(self.results.pose_landmarks.landmark):
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmList.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)
    return lmList

import cv2
import time
import pose_estimation_class as pm
import mediapipe as mp
import argparse

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", required=True,
                help="path to our input video")
ap.add_argument("-o", "--output", required=True,
                help="path to our output video")
ap.add_argument("-s", "--fps", type=int, default=30,
                help="set fps of output video")
ap.add_argument("-b", "--black", type=str, default=False,

```

```
        help="set black background")
args = vars(ap.parse_args())

pTime = 0
black_flag = eval(args["black"])
cap = cv2.VideoCapture(args["input"])
out = cv2.VideoWriter(args["output"], cv2.VideoWriter_fourcc(*"MJPG"),
                      args["fps"], (int(cap.get(3)), int(cap.get(4))))

detector = pm.PoseDetector()

while(cap.isOpened()):
    success, img = cap.read()

    if success == False:
        break

    img, p_landmarks, p_connections = detector.findPose(img, False)

    # use black background
    if black_flag:
        img = img * 0

    # draw points
    mp.solutions.drawing_utils.draw_landmarks(img, p_landmarks, p_connections)
    lmList = detector.getPosition(img)

    cTime = time.time()
    fps = 1 / (cTime - pTime)
    pTime = cTime
```



```
out.write(img)
cv2.imshow("Image", img)
cv2.waitKey(1)

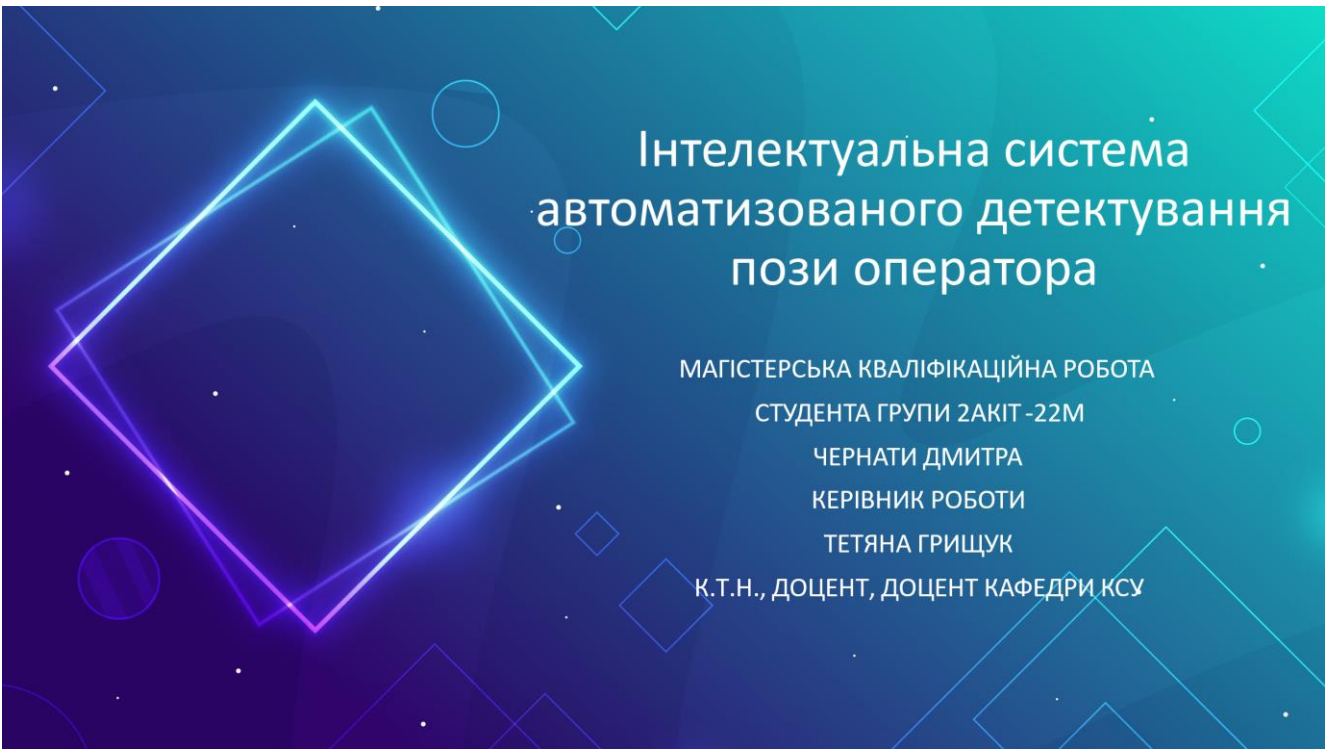
cap.release()
out.release()
cv2.destroyAllWindows()
```

Додаток Г

ІЛЮСТРАТИВНА ЧАСТИНА

Інтелектуальна система автоматизованого детектування пози оператора
(тема)

1. Об'єкт, предмет та мета дослідження;
2. Задачі дослідження;
3. Структурна схема використання системи;
4. Схема роботи алгоритму системи розпізнавання пози людини;
5. Архітектура системи розпізнавання пози людини;
6. Візуалізація розробленої системи та порівняння із OpenPose на наборі даних nuScenes;
7. Візуалізація розробленої системи та порівняння із Mask RCNN на наборі даних nuScenes;
8. Показники у відсотках, оцінені на датасеті COCO, встановлені при оптимальній роздільній здатності
9. Економічна частина;
10. Висновки;



Інтелектуальна система автоматизованого детектування пози оператора

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СТУДЕНТА ГРУПИ ЗАКІТ -22М

ЧЕРНАТИ ДМИТРА

КЕРІВНИК РОБОТИ

ТЕТЯНА ГРИЦУК

К.Т.Н., ДОЦЕНТ, ДОЦЕНТ КАФЕДРИ КСУ

Метою дослідження є покращення виявлення та розуміння людської поведінки бортовими комп'ютерами шляхом удосконалення методів визначення поз людини в реальному часі.

Об'єктом дослідження є процес розпізнавання жестів людини в режимі реального часу з потокового відео.

Предметом дослідження є математичні моделі та методи розпізнавання поз людини на зображенні.

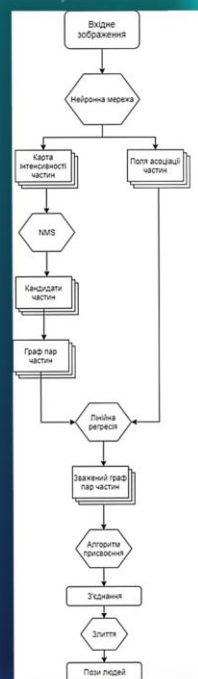
Задачі дослідження:

- аналіз існуючих алгоритмічних програмних аналогів предметній області;
- реалізувати алгоритми попередньої обробки вхідних даних для подачі в нейронні мережі;
- розробка системи розпізнавання жестів людини в відеопотоці в реальному часі;
- визначення ефективності розробленого алгоритму та порівняння його з існуючими аналогічними алгоритмами на вибраних наборах даних.

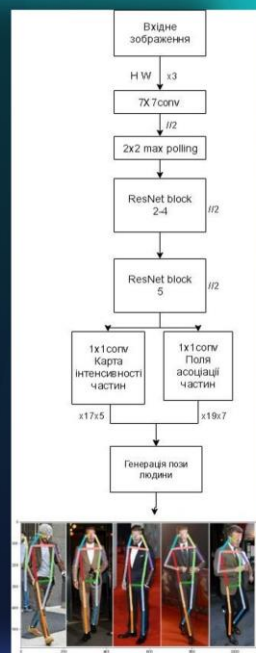
Структурна схема використання системи



Схема роботи алгоритму системи розпізнавання пози людини



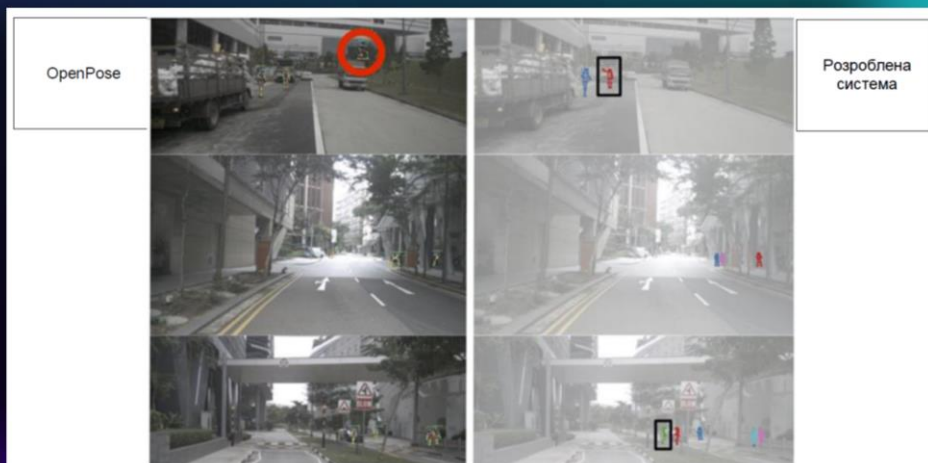
Архітектура системи розпізнавання пози людини



Візуалізація розробленої системи та порівняння OpenPose на наборі даних Scenes



Візуалізація розробленої системи та порівняння Mask RCNN на наборі даних CityScenes



Показники у відсотках оцінені на датасеті COCO, встановлені при оптимальній роздільній здатності

	AP	AP ^M	AP ^L
Mask R-CNN	62.1	58.1	70.6
OpenPose	60.9	56.8	67.1
PersonLab	64.4	62.3	72.1
Наш метод	66.5	62.7	72.9

Економічна частина

Загальні витрати на завершення науково-дослідної роботи та оформлення її становлять - 372000 грн.

Спрогнозований чистий дохід - 16307549 грн.

Період окупності - 0,86 р.

Висновки

- проведено аналіз аналогів програмного забезпечення та наукової літератури з предметної області;
- розроблено алгоритми попередньої обробки даних;
- створено навчальні і тестові зразки;
- виконано навчання та тестування моделі нейронних мереж для розпізнавання жестів людини в реальному часі.



Дякую за увагу