

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет  
Кафедра

Комп'ютерної інженерії та управління  
Комп'ютерних інтелектуальних технологій та систем

## **КВАЛІФІКАЦІЙНА РОБОТА** **Пояснювальна записка**

рівень вищої освіти                      другий (магістерський)

Розпізнавання тривимірних об'єктів у режимі реального часу за допомогою згорткових штучних нейронних мереж

Виконав:

студент 2 курсу, групи КІТм-21-2

Сорока Д.О.

Спеціальність 123 Комп'ютерна інженерія

Тип програми    освітньо-професійна

Освітня програма    Комп'ютерні інтелектуальні  
технології

Керівник            проф. Безсонов О.О.

Допускається до захисту

\_\_\_\_\_ (підпис)

Зав. кафедри

(підпис)

О.Г. Руденко

2022 р.

Харківський національний університет радіоелектроніки

Факультет	Комп'ютерної інженерії та управління
Кафедра	Комп'ютерних інтелектуальних технологій та систем
Рівень вищої освіти	другий (магістерський)
Спеціальність	123 Комп'ютерна інженерія
Тип програми	освітньо-професійна
Освітня програма	Комп'ютерні інтелектуальні технології

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові \_\_\_\_\_ Сороці Дмитру Олеговичу \_\_\_\_\_

1. Тема роботи (проекту) Розпізнавання тривимірних об'єктів у режимі реального часу за допомогою згорткових штучних нейронних мереж  
затверджена наказом університету від "07" листопада 2022р. № 1455
2. Термін подання студентом роботи до екзаменаційної комісії \_\_\_\_\_ 2022р.
3. Вихідні дані до роботи (проекту) \_\_\_\_\_
  - 1) Мова програмування Python \_\_\_\_\_
  - 2) Мова програмування JavaScript \_\_\_\_\_
  - 3) Платформа MediaPipe \_\_\_\_\_
4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_
  - 1) Аналіз предметної області \_\_\_\_\_
  - 2) Аналіз існуючих рішень \_\_\_\_\_
  - 3) Аналіз та вибір методу виявлення та розпізнавання 3D-об'єктів \_\_\_\_\_
  - 4) Програмна реалізація та застосування обраних методів \_\_\_\_\_
  - 5) Висновки \_\_\_\_\_
5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням кафедри) \_\_\_\_\_

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно до наказу, зазначеному у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Видача та узгодження теми проекту	08.11.2022	виконано
2.	Аналіз предметної області	09.11.2022 – 15.11.2022	виконано
3.	Аналіз існуючих рішень	16.11.2022	виконано
4.	Аналіз та вибір методу виявлення та розпізнавання 3D-об'єктів	17.11.2022 – 20.11.2022	виконано
5.	Програмна реалізація та застосування обраних методів	21.11.2022 – 30.11.2022	виконано
6.	Оформлення матеріалів атестаційної роботи	01.12.2022 – 06.12.2022	виконано
7.	Перевірка виконаного проекту керівником	07.12.2022	виконано
8.	Захист проекту	20.12.2022	виконано

Дата видачі завдання 08 листопада 2022 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ проф. Безсонов О.О.  
(підпис) (посада, ініціали, прізвище)

## РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 72 сторінки, 28 рисунків, 1 додаток, 16 джерел.

ОКЛЮЗИЯ, 3D, PARE, КОНВЕЄР, CUDA, РОЗПІЗНАВАННЯ, ГЕНЕРАЦІЯ, МОДЕЛЬ, ФУНКЦІЯ, МАСКА, УВАГА, МІТКИ, ПЛОЩИНА

Темою кваліфікаційної роботи є розпізнавання тривимірних об'єктів у режимі реального часу за допомогою згорткових штучних нейронних мереж.

Метою кваліфікаційної роботи є розробка пакету програмного забезпечення для розпізнавання тривимірних об'єктів у режимі реального часу.

Об'єктом дослідження цієї роботи є реалізація методів виявлення та розпізнавання тривимірних об'єктів, а саме за допомогою PARE та платформи MediaPipe.

Предметом дослідження є аналіз зображення за допомогою вищевказаних методів та засобів.

Відповідно до мети кваліфікаційної роботи було виконано поставлені задачі. Проведено аналіз предметної області та підтверджена актуальність розробки програми для виявлення 3D-об'єктів. Проаналізовано існуючі рішення та методи для вирішення цієї задачі. Створено пакет програмного забезпечення, що втілює ідею розпізнавання тривимірних об'єктів за допомогою мов Python та JavaScript, який може приймати на вхід зображення з вебкамери та в режимі реального часу надавати інформацію про розпізнаний об'єкт.

## ABSTRACT

Explanatory note of the qualification work: 72 pages, 28 figures, 1 appendice, 16 sources.

OCCLUSION, 3D, PARE, PIPELINE, CUDA, RECOGNITION, GENERATION, MODEL, FUNCTION, MASK, ATTENTION, LABELS, PLANE

The topic of the qualification work is the recognition of three-dimensional objects in real time using convolutional artificial neural networks.

The purpose of the qualification work is to develop a software package for recognizing three-dimensional objects in real time.

The object of research of this work is the implementation of methods for detection and recognition of three-dimensional objects, namely with the help of PARE and the MediaPipe platform.

The subject of research is image analysis using the above methods and tools.

In accordance with the purpose of the qualification work, the set tasks were completed. The analysis of the subject area was carried out and the relevance of the development of the program for the detection of 3D objects was confirmed. Existing solutions and methods for solving this problem are analyzed. A software package has been created that implements the idea of recognizing three-dimensional objects using Python and JavaScript, which can receive images from a webcam and provide information about the recognized object in real time.

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет  
Кафедра

Комп'ютерної інженерії та управління  
Комп'ютерних інтелектуальних технологій та систем

**АНОТАЦІЯ**  
**КВАЛІФІКАЦІЙНОЇ РОБОТИ**

рівень вищої освіти                      другий (магістерський)

Розпізнавання тривимірних об'єктів у режимі реального часу за допомогою  
згорткових штучних нейронних мереж  
(тема)

Виконав:

студент 2 курсу, групи КІТм-21-2

\_\_\_\_\_ Сорока Д.О.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми    освітньо-професійна

Освітня програма    Комп'ютерні

інтелектуальні технології

Керівник \_\_\_\_\_ проф. Безсонов О.О.

(посада, ініціали, прізвище)

2022 р.

## АНОТАЦІЯ

Сорока Д.О. Розпізнавання тривимірних об'єктів у режимі реального часу за допомогою згорткових штучних нейронних мереж – Магістерська кваліфікаційна робота.

**Актуальність теми дослідження.** Результати ідентифікації динамічних 3D-об'єктів у більшості випадків становлять собою набір вхідних даних для послідовно оброблюючих блоків складних інтелектуальних систем. І швидкість отримання та точність результатів є провідними критеріями в таких системах. Властивості згорткових штучних нейронних мереж дають змогу забезпечити ці вимоги при обробці потокових даних у режимі реального часу, тому дослідження цих мереж для подальшого ефективного використання є актуальною задачею.

У комп'ютерному зорі розпізнавання 3D-об'єктів передбачає розпізнавання та визначення 3D-інформації, такої як поза, об'єм або форма, вибраних користувачем 3D-об'єктів на фотографії чи сканованому діапазоні. Як правило, приклад об'єкта, який потрібно розпізнати, представляється системі зору в контрольованому середовищі, а потім для довільного введення, такого як відеопотік, система визначає місцезнаходження попередньо представленого об'єкта. Це можна зробити як офлайн, так і в реальному часі. Алгоритми для вирішення цієї проблеми спеціалізуються на локації окремого попередньо ідентифікованого об'єкта, і їх можна порівняти з алгоритмами, які працюють із загальними класами об'єктів, такими як системи розпізнавання облич або тривимірне розпізнавання об'єктів. Завдяки низькій вартості та простоті отримання фотографій, значну кількість досліджень було присвячено розпізнаванню тривимірних об'єктів на фотографіях [3].

**Метою кваліфікаційної роботи** є розробка пакету програмного забезпечення для розпізнавання тривимірних об'єктів у режимі реального часу.

**Об'єктом дослідження** роботи є реалізація методів виявлення та розпізнавання тривимірних об'єктів, а саме за допомогою PARE та платформи MediaPipe.

**Предметом дослідження** є аналіз зображення за допомогою вищевказаних методів та засобів.

Відповідно до мети кваліфікаційної роботи було виконано поставлені задачі:

- проведено аналіз предметної області та підтверджена актуальність розробки програми для виявлення 3D-об'єктів;
- проаналізовано існуючі рішення та методи для вирішення цієї задачі, їх недоліки та переваги;
- створено пакет програмного забезпечення, що втілює ідею розпізнавання тривимірних об'єктів за допомогою мов Python та JavaScript, який може приймати на вхід зображення з вебкамери та в режимі реального часу надавати інформацію про розпізнаний об'єкт.

Мова програмування Python широко використовується для наукових досліджень, ось деякі з причин її популярності у цій сфері [5]:

- Python має вбудовану підтримку наукових обчислень. Більшість дистрибутивів Python включають екосистему SciPy (з відкритим кодом), яка включає SciPy (бібліотеку SciPy), пакет числових обчислень під назвою NumPy і кілька незалежних наборів інструментів, кожен з яких відомий як Scikits. Matplotlib, бібліотека 2D-графіків для візуалізації, також є частиною екосистеми SciPy. Matplotlib дуже схожий на MATLAB з точки зору його функціональності та використання, але є відкритим кодом;

- Python має мости до MATLAB або Octave . Python може інсталювати MATLAB Engine API, щоб програми Python могли викликати MATLAB як обчислювальний механізм. Програми MATLAB також можуть викликати функції Python з деякими обмеженнями. Деякі дистрибутиви Python можуть використовувати пакети Python, такі як PyMatbridge, які можуть підтримувати як



MATLAB, так і Octave і дозволяють використовувати MATLAB у Jupyter Notebooks;

- Python – це мова з широкими можливостями розширення . Розробники багато років використовували оболонки Python для програм C/C++. Python використовує зовнішній функціональний інтерфейс C для Python (cFFI) для безпосередньої взаємодії з кодом C. Такі інструменти, як SWIG, спрощують це. Розробники також можуть викликати підпрограми Fortran з Python за допомогою пакета генератора інтерфейсу Fortran для Python, F2Py, який тепер є частиною NumPy;

- Python має дуже хороші параметри введення/виведення (I/O) . До недавнього часу введення/виведення Fortran традиційно базувалося на записах. Python навпаки вже давно підтримує додаткові пакети для підтримки кількох варіантів всіх типів форматів введення-виведення, включаючи формати реального часу та потокові формати;

- Python має сильну підтримку для автоматизації завдань . Вбудовані функції сценаріїв Python і численні пакети мають сильну підтримку для автоматизації завдань. Автоматизація повторюваних завдань і реєстрація даних є простими та не потребують великих зусиль;

- Python може використовувати веб-інтерфейс . Пакети Python, такі як Django та Flask, дозволяють розробляти та використовувати Python як API із веб-інтерфейсом. Ця функція особливо корисна при використанні хмарної інфраструктури як платформи для доступу до систем високопродуктивних обчислень (HPC).

Безпосередня регресія 3D пози та форми людини (HPS) із зображень RGB має багато застосувань у робототехніці, комп'ютерній графіці, AR/VR тощо. Завдання взяти одне зображення або відеоряд як введення та регресії параметрів моделі людського тіла, наприклад, SMPL як вихід. Завдяки глибоким CNN, це завдання має швидкий прогрес. Проте в дикій природі, люди часто знаходяться під

оклюзією або виникає самонакладання частин тіла, через близьку взаємодію з іншими людьми або через закриття об'єктів, таких як меблі чи інший вміст сцени.

Інші методи використовують усі пікселі вхідних даних для прогнозування єдиного набору пози та параметрів форми. Таким чином їх оцінки пози особливо чутливі до навіть невеликих збурень у спостереженнях тіла та його частин. Для найсучасніших методів (SOTA) навіть відносно невеликі оклюзії тільки одного суглоба, може призвести до абсолютно неправдоподібної пози передбачення. Такі методи, як SPIN дуже чутливі до локалізованої оклюзії. Щоб вирішити цю проблему, пропонується метод частково керованого механізму уваги, що здійснює пряму регресію, підхід більш надійний до оклюзії.

Запропонований метод називається Part Attention REgressor (PARE). Він має два завдання: основне – навчання, регресувати тривимірні параметри тіла наскрізним способом, а допоміжним завданням є вивчення ваги уваги частини тіла.

Кожне завдання має власну гілку вилучення функцій, вирівняну за пікселями. Спрямовуємо гілку уваги за допомогою міток сегментації частин на ранніх етапах навчання та продовжуємо без них на пізніх етапах, тому називаємо це увагою, керованою частинами тіла. Наше ключове розуміння полягає в тому, щоб бути стійкими до оклюзій, мережа повинна використовувати піксельне вирівнювання зображення, особливості видимих частин, щоб міркувати про закриті частини.

Виявлення об'єктів – це широко досліджена проблема комп'ютерного зору, але більшість досліджень зосереджено на передбаченні 2D-об'єктів. У той час як 2D-прогнозування забезпечує лише 2D-обмежувальні прямокутники, розширивши передбачення до 3D, можна зафіксувати розмір, положення та орієнтацію об'єкта у просторі, що веде до різноманітних застосувань у робототехніці, безпілотних транспортних засобах, пошуку зображень і доповненій реальності. Хоча виявлення 2D-об'єктів є відносно зрілим і широко використовується в промисловості, виявлення 3D-об'єктів із 2D-зображень є складною проблемою

через брак даних і різноманітність зовнішніх виглядів і форм об'єктів у межах категорії.

Одним з можливих вирішень цієї проблеми є створення конвеєра виявлення 3D-об'єктів у режимі реального часу для повсякденних об'єктів за допомогою MediaPipe [14]. Цей конвеєр виявляє об'єкти на 2D-зображеннях і оцінює їхні пози та розміри за допомогою моделі машинного навчання (ML), навченої на щойно створеному наборі 3D-даних.

Популярним підходом є доповнення даних реального світу синтетичними даними, щоб підвищити точність прогнозу. Однак спроби зробити це часто дають погані, нереалістичні дані або, у випадку фотореалістичної візуалізації, вимагають значних зусиль і обчислень. Новий підхід під назвою «генерація синтетичних даних доповненої реальності» розміщує віртуальні об'єкти в сценах, які мають дані сеансу доповненої реальності, що дозволяє нам використовувати пози камери, виявлені площини поверхні та оцінене освітлення для створення фізично вірогідних місць розташування з освітленням, яке відповідає сцені. Цей підхід (рисунок 3.2.1) дозволяє отримати високоякісні синтетичні дані із відтвореними об'єктами, які дотримуються геометрії сцени та ідеально підходять до реального фону. Поєднуючи дані реального світу та синтетичні дані AR, можемо підвищити точність приблизно на 10%.

Метод PARE добре показав себе при використанні в реальних умовах і може працювати ще краще при збільшенні обчислювальних потужностей. Розроблена за допомогою платформи MediaPipe програма також виконує поставлені задачі з виявлення та розпізнавання об'єктів.

ОКЛЮЗІЯ, 3D, PARE, КОНВЕЄР, CUDA, РОЗПІЗНАВАННЯ, ГЕНЕРАЦІЯ, МОДЕЛЬ, ФУНКЦІЯ, МАСКА, УВАГА, МІТКИ, ПЛОЩИНА

Публікації здобувача за темою роботи:

1. Сорока Д.О. Використання комп'ютерного зору для ідентифікації змісту етикеток харчових продуктів: Результати розвитку наукової думки 2021: матеріали III міжнародної студентської конференції, м. Вінниця 24 грудня 2021. Вінниця. С. 15-16.

2. Сорока Д.О. Виявлення 3D-об'єктів у реальному часі на мобільних пристроях за допомогою MediaPipe: Наука сьогодення: від досліджень до стратегічних рішень: матеріали V міжнародної студентської конференції, м. Київ 2 грудня 2022. Київ. С. 146-147.

Публікації керівника за темою роботи:

1. Intelligent Identification System of the Process Liquid Solutions Composition. Bezsonov, O., Ilyunin, O., Khusanov, A., Rudenko, O., Sotnikov, O., CEUR Workshop Proceedings, 2022, 3171, pp. 960–973 (Scopus)

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	15
ВСТУП.....	16
1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	18
1.1 Актуальність задачі.....	18
1.2 Існуючі рішення .....	19
1.3 Аналіз чутливості оклюзії .....	21
2 МЕТОД PARE.....	24
2.1 Архітектура моделі та втрати .....	26
2.2 Деталі реалізації .....	29
3 ВИЯВЛЕННЯ 3D-ОБ'ЄКТІВ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ MEDIAPIPE.....	31
3.1 Отримання 3D-навчальних даних у реальному світі.....	32
3.2 Генерація синтетичних даних AR .....	33
3.3 Конвеєр ML для виявлення 3D-об'єктів.....	34
3.4 Виявлення та відстеження в MediaPipe .....	36
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	37
4.1 Мова програмування.....	37
4.2 Використані бібліотеки .....	39
4.2.1 CUDA .....	39
4.2.2 NumPy .....	40
4.2.3 Torch .....	42
4.3 Отримані результати за допомогою PARE.....	43
4.4 Отримані результати за допомогою MEDIAPIPE .....	54
ВИСНОВКИ.....	57
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	58

Додаток А Код програми.....**Ошибка! Закладка не определена.**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І  
ТЕРМЕНІВ

PARE – регресор часткової уваги (англ., Part Attention Regressor)

CNN – згортова нейронна мережа (англ., Convolutional Neural Network)

Оклюдія – ситуація, в якій два об'єкти розташовані приблизно на одній лінії і один об'єкт, розташований ближче до віртуальної камери або вікна перегляду, частково або повністю закриває видимість іншого об'єкта

UV розгортка – процес в 3D моделюванні, який полягає в накладанні двовимірного зображення на тривимірну модель. Літерами U і V позначають осі координат площини розгортки, оскільки літери X, Y і Z використовуються для позначення просторових координат

CUDA - програмно-апаратна архітектура паралельних обчислень, яка дозволяє істотно збільшити обчислювальну продуктивність завдяки використанню графічних процесорів фірми Nvidia (англ., Compute Unified Device Architecture)

FPS – кількість кадрів в секунду на екрані монітора (англ., Frames Per Second)

## ВСТУП

Розпізнавання тривимірних об'єктів було дуже активним протягом останнього десятиліття. Спосіб розпізнавання тривимірного об'єкта залежить від властивостей об'єкта. Для простоти багато існуючих алгоритмів зосереджено на розпізнаванні твердих об'єктів, що складаються з однієї частини, тобто об'єктів, просторове перетворення яких є евклідовим рухом. Було використано два загальні підходи до проблеми: підходи до розпізнавання образів використовують інформацію про зовнішній вигляд зображення низького рівня, щоб знайти об'єкт, тоді як геометричні підходи, засновані на ознаках, створюють модель для об'єкта, який потрібно розпізнати, і порівнюють модель із фотографією.

Більшість систем поділяються на три основні категорії:

- системи, які використовують лише дані про інтенсивність;
- системи, які використовують лише дані діапазону;
- системи, які використовують як дані діапазону, так і інтенсивності (іноді включаючи колір).

У системах, заснованих на зображенні інтенсивності, точки та відрізки прямих ліній все ще найчастіше використовують функції. Насправді метод вирівнювання, який отримав популярність, призвів до значної кількості систем, які сліпо зіставляли трійки точок або відрізків лінії із зображення до подібних трійок із моделі, використовуючи невелику кількість контекстної інформації або взагалі не використовуючи її. Усі ці алгоритми роблять припущення, що точки або відрізки ліній є надійними ознаками класу об'єктів, які необхідно розпізнати або знайти, і положення об'єкту можна однозначно визначити з невеликого набору цих ознак. Ці припущення є вірними лише для багатогранних об'єктів або об'єктів із кількома гострими прямими краями. Система може розпізнавати об'єкти які мають плоскі, циліндричні та різьбові поверхні, і вона призначена для обробки оклюзії.



У системах на основі діапазонних зображень примітивні поверхні (зазвичай плоскі або квадратичні) є найбільш загальними рисами, але також використовуються тривимірні відрізки ліній і точки. Оскільки поверхні від даних діапазону є більш надійними, ніж області поверхні від градацій сірого, низка систем

використовують властивості та зв'язки між поверхнями в алгоритмах зіставлення. Цей добре працює для простих об'єктів з невеликою кількістю простих поверхонь.

Системи, які працюють із складнішими поверхнями вільної форми, зазвичай шукають точки інтересу і виконують точкове зіставлення. Знову ж таки, надійне виявлення характерних точок має вирішальне значення [15].

Результати ідентифікації динамічних 3D-об'єктів у більшості випадків становлять собою набір вхідних даних для послідовно оброблюючих блоків складних інтелектуальних систем. І швидкість отримання та точність результатів є провідними критеріями в таких системах. Властивості згорткових штучних нейронних мереж дають змогу забезпечити ці вимоги при обробці потокових даних у режимі реального часу, тому дослідження цих мереж для подальшого ефективного використання є актуальною задачею.

# 1 АКТУАЛЬНІСТЬ ПРОБЛЕМИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Актуальність задачі

У комп'ютерному зорі розпізнавання 3D-об'єктів передбачає розпізнавання та визначення 3D-інформації, такої як поза, об'єм або форма, вибраних користувачем 3D-об'єктів на фотографії чи сканованому діапазоні. Як правило, приклад об'єкта, який потрібно розпізнати, представляється системі зору в контрольованому середовищі, а потім для довільного введення, такого як відеопотік, система визначає місцезнаходження попередньо представленого об'єкта. Це можна зробити як офлайн, так і в реальному часі. Алгоритми для вирішення цієї проблеми спеціалізуються на локації окремого попередньо ідентифікованого об'єкта, і їх можна порівняти з алгоритмами, які працюють із загальними класами об'єктів, такими як системи розпізнавання облич або тривимірне розпізнавання об'єктів. Завдяки низькій вартості та простоті отримання фотографій, значну кількість досліджень було присвячено розпізнаванню тривимірних об'єктів на фотографіях [3].

Незважаючи на значний прогрес, сучасні тривимірні методи оцінки пози та форми людини залишаються чутливими до часткової оклюзії та можуть давати різко неправильні прогнози, хоча більшу частину тіла можна спостерігати. Щоб вирішити цю проблему, можна використати механізм м'якої уваги, який має назву регресор часткової уваги (PARE), він навчається передбачати маски уваги, керовані частинами тіла. Найсучасніші методи покладаються на представлення глобальних ознак, що робить їх чутливими навіть до невеликих оклюзій. Механізм частково керованої уваги PARE долає ці проблеми, використовуючи інформацію про видимість окремих частин тіла, одночасно використовуючи інформацію з сусідніх частин тіла для прогнозування закритих частин. На практиці встановлено,

що PARE досягає більш точних і надійних результатів реконструкції, ніж існуючі підходи як для специфічних оклюзій, так і для стандартних тестів [13].

## 1.2 Існуючі рішення

Ми зосереджуємось на оцінці 3D-форми людини та її пози за допомогою зображення RGB та обговоримо, як обробляються оклюзії в інших підходах при різних сценаріях, наприклад, самооклюзія, оклюзія кадру та оклюзія об'єкта сцени.

Тривимірна поза та форма з одного зображення.

При оцінці фігури та пози людини багато методів виводять параметри тривимірних моделей людського тіла. Початкова обробка передбачає представлення тривимірного тіла за допомогою ключових точок і силуетів. Ці підходи крихкі, потребують ручного введення, використання додаткових даних, наприклад, багатопрозорі зображення, або погано узагальнені зображення в дикій природі. SMPLify був першим автоматизованим методом, який відповідав моделі SMPL на вихід 2D детектора ключової точки. Ласснер та інші використовують силуети разом із ключовими точками під час примірки. В той час як глибокі нейронні мережі регресують параметри SMPL безпосередньо з пікселів. Щоб позбутися 3D наземних похибок в дикій природі, методи використовують втрату повторної проєкції 2D ключової точки та слабкий контроль, використовують проміжні 2D представлення, наприклад, сегментація тіла на частини, 2D розріджені ключові точки та інше. Примітка, що використання сегментації частин дуже відрізняється від PARE, у якому сегментації частин використовується для полегшення м'якої уваги. Колотоурос та інші об'єднують HMR і SMPLify у навчальний цикл. На кожному кроці, HMR ініціалізує SMPLify, який відповідає моделі тіла до 2D-стиків, що забезпечує кращий нагляд за мережею. Наведені вище методи зазвичай чутливі до оклюзії.

Неявна обробка оклюзії (доповнення даних).

В ідеалі регресоване 3D тіло має бути однаковим з оклюзією або без неї. Поточна оцінка пози та форми SOTA методи безпосередньо кодує всю вхідну область як одну функцію CNN після глобального середнього об'єднання шляхом регресії параметрів моделі тіла. Відсутність вирівняної піксельної структури ускладнює мережам явне визначення оцінки про розташування та видимість частин тіла. Загальний спосіб досягнення стійкості до оклюзії в них – це розширення даних. Наприклад, оклюзія кадру часто імітується кадруванням, в той час як оклюзія об'єкта апроксимується шляхом накладання плям об'єкта на зображення. Замість того, щоб застосовувати доповнення до вхідних зображень, Ченг та інші застосовують доповнення до теплових карт, які містять багатшу семантичну інформацію, отже, оклюзії можна змодельовати у більш інтелектуальний спосіб. Хоча ці синтетичні оклюзії корисні, вони не повністю вловлюють складність оклюзій у реалістичних зображеннях, та не дають уявлення про те, як покращити мережеву архітектуру, щоб вона за своєю суттю була більш стійкою до оклюзії.

Явна обробка оклюзії.

Щоб міркувати чіткіше про оклюзії, попередня обробка використовує інформацію про видимість. Наприклад, Ченг та інші уникають включення оклюзії суглобів при обчисленні втрат під час навчання. Така інформація про видимість отримана шляхом представлення тіла людини як набору циліндрів, що нереально і тільки справляється з самооклюзією. Ванг та інші навчилися передбачати мітки оклюзії, щоб обнулити закриті ключові точки перед застосуванням часової згортки до послідовності 2D ключових точок.

Оклюзія типу «особа-особа» особливо поширена та є складним завданням. Для регресії кількох осіб використовують втрату взаємопроникнення, щоб уникнути зіткнення, і порядковий номер втрати для вирішення неоднозначності

глибини. Оцінка всіх людей на зображенні одночасно, унеможлиблює дізнатися про конкретну оклюзію. Поки інші методи використовують навчальні функції для підвищення стійкості до оклюзії між особами, PARE вчиться концентрувати увагу на окремих частинах тіла.

Також можливе використання масок помітності, як видимість інформації для отримання стійкості до оклюзій сцени чи об'єкта. Людські сітки параметризовані UV-картами, де кожний піксель зберігає тривимірне розташування вершини, а оклюзії – це проблема малювання зображення. Вимога точних карт помітності обмежує продуктивність розпізнавання зображень у дикій природі. Крім того, UV-координати можуть призвести до артефактів сітки.

### 1.3 Аналіз чутливості оклюзії

Щоб витягти ознаки з області вхідного зображення  $I$ , поточні підходи прямої регресії використовують ResNet-50 магістраль і беруть функції після глобального середнього пулу (GAP), а потім MLP, який регресує та вдосконалює параметри ітераційно. Для того, щоб дослідити вплив оклюзій на цей тип архітектури, можна систематично покривати різні частини зображення сірим квадратом, щоб проаналізувати, як змінюються карти ознак і результат класифікатора. В контрасті, необхідно провести сіру пляму оклюзії поверх зображення і регресувати пози тіла за допомогою SPIN. Замість обчислення класифікаційної оцінки, є можливість виміряти спільну евклідову відстань між базовою правдою та прогнозованим розташуванням суглобів. Створюємо теплову карту помилок, у якій кожен піксель вказує, яку помилку модель створює для з'єднання  $j$ , коли оклюдер центрується на цьому пікселі. Крім суглобів на тепловій карті, обчислюємо сукупну чутливість оклюзії, що показує, як це впливає на середню похибку суглоба за оклюзією.

Теплові карти помилок для SPIN візуалізуються на рисунку 1.1, для зразка – зображення з набору даних 3DPW. Кожне підзображення відповідає певному суглобу та гарячому регіону – це місця, де оклюзія викликає високу похибку цього суглобу.

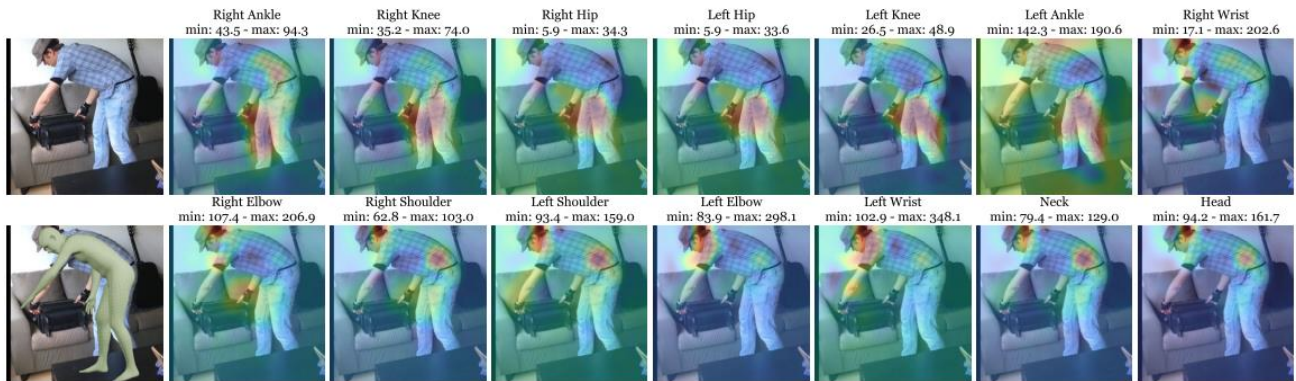


Рисунок 1.1 – Аналіз чутливості оклюзії

Ця візуалізація дозволяє нам зробити кілька спостережень. Помилки низькі у фоновому режимі та високі на тілі. Це свідчить про те, що SPIN навчився звертати увагу на значущі регіони. Суглоби, видимі на вихідному зображенні мають високі помилки, коли вони закриті квадратом, як і очікувалось. Для суглобів, які мають природну оклюзію, мережа покладається на інші регіони, щоб обґрунтувати оклюзію пози. Наприклад, у верхньому рядку рисунку 1.1 спостерігаємо високі помилки для лівої та правої щиколотки, коли оклюзуємо область стегна. Оскільки в мережі відсутні ознаки зображення для оклюзованих частин, слід шукати докази в іншому місці зображення. Такі залежності трапляються не тільки між сусідніми частинами, оклюзія може мати довгострокові наслідки (наприклад, оклюзія тазу спричиняє помилки при розпізнаванні голови).

Ми далі накладаємо оцінене тіло на теплову карту, щоб передати похибку на піксель до видимих вершин. Проводимо цей аналіз над повним набором даних 3DPW, об'єднуємо помилку pervertex у наборі даних і візуалізуємо результат на

модель тіла SMPL, що дає нам одну сітку чутливості оклюзії на суглоб. Наприклад, рисунок 1.2 (а) показує, що лівий лікоть є чутливим до оклюзії обличчя, лівого плеча та області лівого плеча.

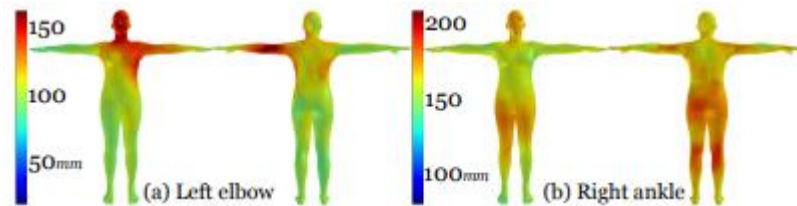


Рисунок 1.2 – Сітки чутливості оклюзії для SPIN

Мітки сегментації частин тіла.

Оскільки у нас є SMPL – робимо анотації для більшості зразків у наших наборах даних, що не потребує додаткових анотацій сегментації частин тіла. Безпосередньо використовуємо анотації SMPL для отримання контролю.

На рис. 1.3 зображено мітки частин тіла. Для кожного суглоба в кінематичного дерева SMPL маємо відповідну мітку для кожної частини тіла.

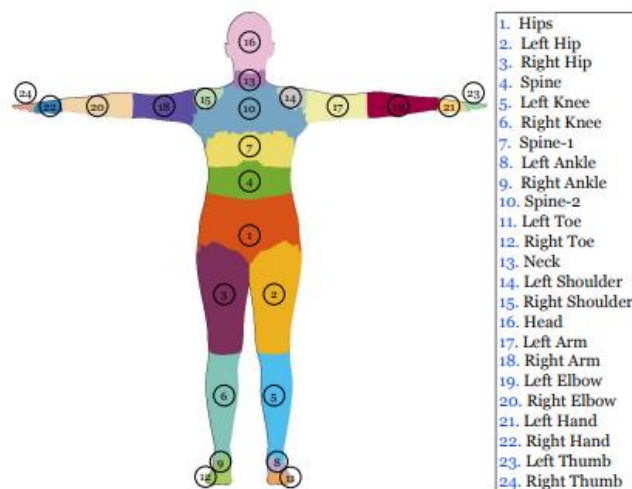


Рисунок 1.3 – Мітки сегментації частин тіла, що використовуються для 2D частина гілки

## 2 МЕТОД PARE

Безпосередня регресія 3D пози та форми людини (HPS) із зображень RGB має багато застосувань у робототехніці, комп'ютерній графіці, AR/VR тощо. Завдання взяти одне зображення або відеоряд як введення та регресії параметрів моделі людського тіла, наприклад, SMPL як вихід. Завдяки глибоким CNN, це завдання має швидкий прогрес. Проте в дикій природі, люди часто знаходяться під оклюзією або виникає самонакладання частин тіла, через близьку взаємодію з іншими людьми або через закриття об'єктів, таких як меблі чи інший вміст сцени.

Інші методи використовують усі пікселі вхідних даних для прогнозування єдиного набору пози та параметрів форми. Таким чином їх оцінки пози особливо чутливі до навіть невеликих збурень у спостереженнях тіла та його частин. Для найсучасніших методів (SOTA) навіть відносно невеликі оклюзії тільки одного суглоба, може призвести до абсолютно неправдоподібної пози передбачення. Такі методи, як SPIN дуже чутливі до локалізованої оклюзії. Щоб вирішити цю проблему, пропонується метод частково керованого механізму уваги, що здійснює пряму регресію, підхід більш надійний до оклюзії.

Запропонований метод називається Part Attention REgressor (PARE). Він має два завдання: основне – навчання, регресувати тривимірні параметри тіла наскрізним способом, а допоміжним завданням є вивчення ваги уваги частини тіла.

Кожне завдання має власну гілку вилучення функцій, вирівняну за пікселями. Спрямовуємо гілку уваги за допомогою міток сегментації частин на ранніх етапах навчання та продовжуємо без них на пізніх етапах, тому називаємо це увагою, керованою частинами тіла. Наше ключове розуміння полягає в тому, щоб бути стійкими до оклюзій, мережа повинна використовувати піксельне вирівнювання зображення, особливості видимих частин, щоб міркувати про закриті частини.



Враховуючи успіх методів, заснованих на увазі, використовуємо знання, отримані з аналізу чутливості оклюзії для концентрації уваги на частинах тіла. Тому контролюємо маску уваги з частиною сегментації, але потім тренуємо мережу наскрізно лише з наглядом за позою, дозволяючи механізму уваги використовувати

всю корисну інформація від організму і оточуючих пікселів. Це дає мережі свободу відвідувати регіони, які він вважає інформативними, у неконтрольований спосіб. Як результат, PARE вчиться покладатися на видимі частини тіла для вдосконалення стійкості до закритих частин і загальна продуктивність у 3D оцінці пози зростає.

Щоб кількісно оцінити показники PARE, провели експерименти на 3DPW, 3DOH і 3DPW-OCC наборах даних. Результати показують, що PARE дає незмінно нижчу похибку, ніж інші сучасні методи, як випадки оклюзії, так і випадки без оклюзії.

Аналіз мотивує нову структуру уваги, керовану частинами тіла, для 3D регресії HPS, яка використовує вирівняні за пікселями локалізовані функції для регресії тіла, пози і форми. Мережа використовує сигнали видимості частини, щоб з'ясувати причини оклюзії суглобів шляхом агрегування ознак з відвідуваних регіонів, і завдяки цьому досягається надійність до оклюзій.

Враховуючи наведені вище спостереження, PARE розроблено з наступних ідей. По-перше, як показано на рис. 1, мережі SOTA вчать звертати увагу на значущі регіони неявно, незважаючи на обмежену просторову інформацію після глобального середнього пулу. Щоб краще зрозуміти, чи є частини тіла видимі чи ні, і знати, чи їх розташування закриті, PARE використовує піксельну структуру, де кожен піксель відповідає області на зображенні та зберігає піксельний рівень репрезентації, а саме об'єм функції. По-друге, оскільки оцінка ваги уваги та вивчення наскрізних тренувальних функцій для 3D-поз – це два різні завдання, PARE – оснащений двома функціональними об'ємами: один з 2D частини, гілка,

яка оцінює ваги уваги, і одна з 3D, гілка тіла, яка виконує регресію параметрів SMPL. Нарешті, щоб змодельовати залежності частин тіла, зазначені вище, PARE використовує сегментацію частин як м'які маски уваги щоб налаштувати внесок кожної функції в 3D-тіло, що розгалужуються по-різному для кожного суглоба [11].

Попередня модель тіла.

SMPL представляє позу тіла та форму  $\theta$ , яка складається з пози  $\theta \in \mathbb{R}^{72}$  і форми  $\beta \in \mathbb{R}^{10}$  параметрів. Використовуємо гендерно-нейтральну модель форми. З урахуванням цих параметрів модель SMPL є диференційованою функцією, яка виводить задану 3D сітку  $M(\theta, \beta) \in \mathbb{R}^{6890 \times 3}$ . Тривимірні місця з'єднання  $J_{3D} = W M \in \mathbb{R}^{J \times 3}$ ,  $J = 24$ , обчислено за допомогою попередньо підготовленого лінійного регресора  $W$ .

## 2.1 Архітектура моделі та втрати

Загальна структура PARE зображена на рис. 2.1 Архітектура працює наступним чином: задано зображення  $I$ , спочатку запускаємо магістраль CNN, щоб отримати об'ємні характеристики, наприклад, перед глобальним середнім рівнем об'єднання для ResNet-50, потім дві окремі гілки виділення ознак для отримання двох об'ємних характеристик зображення. Позначимо 2D частину розгалуження як  $P \in \mathbb{R}^{H \times W \times (J+1)}$ , моделювання  $J$  частина уваги і 1 фонові маски, де  $H$  і  $W$  – висота і ширина об'єму ознаки та зберігається кожен піксель  $(h, w)$ , ймовірність приналежності до частини тіла  $j$ . Іншу гілку, позначену  $F \in \mathbb{R}^{H \times W \times C}$ , яка використовується для оцінки параметрів 3D тіла. Він має однакові просторові розміри  $H \times W$  як  $P$ , але інша кількість каналів,  $C$ .

Нехай  $P_j \in \mathbb{R}^{H \times W}$  і  $F_c \in \mathbb{R}^{H \times W}$  позначають  $j$ -й і  $c$ -й канал  $P$  та  $F$  відповідно, нехай  $F' \in \mathbb{R}^{J \times C}$  представляють остаточний тензор ознак. Кожен елемент у  $F_c$  вносить пропорційний внесок у  $F'$  згідно з відповідним елементом у  $P_j$  після просторової softmax нормалізації  $\sigma$ . Формально, елемент у місці  $(j, c)$  в  $F'$  обчислюється як:

$$F'_{j,c} = \sum_{h,w} \sigma(P_j) \odot F_c, \quad (2.1)$$

де  $\odot$  добуток Адамара.

Іншими словами, використовуємо  $\sigma(P_j)$  як м'яку маску уваги для агрегування характеристик у  $F_c$ . Ця операція може бути ефективно реалізована як скалковий добуток, подібний до існуючих реалізацій уваги:  $F' = \sigma(\tilde{P}) \tilde{F}$ , де  $\tilde{P} \in \mathbb{R}^{HW \times J}$  та  $\tilde{F} \in \mathbb{R}^{HW \times C}$  позначають змінені  $P$  (без фонові маски) і  $F$  відповідно. Ця операція уваги припускає, що якщо певний піксель має вищу вагу уваги, він більше сприяє остаточному представленню  $F'$ . Контролюємо гілку 2D частини  $P$  за допомогою міток сегментації базової правди, що допомагає картам уваги видимих частин сходитися до відповідних регіонів. Для частин оклюзій, це заохочує нульові ваги уваги для всіх пікселів у  $P_j$ , оскільки вони не існують у мітках сегментації базової правди. Карта уваги з усіма 0 вагами є небажаною та, на практиці, також неможливою, оскільки просторова softmax гарантує, що сума всіх елементів дорівнює 1. Тому було прийнято гібридний підхід, який керує гілкою 2D деталей лише на початковому етапі і продовжує навчання без жодного нагляду. Це дозволяє мережі відвідувати інші регіони, щоб оцінити пози оклюзованого суглоба.

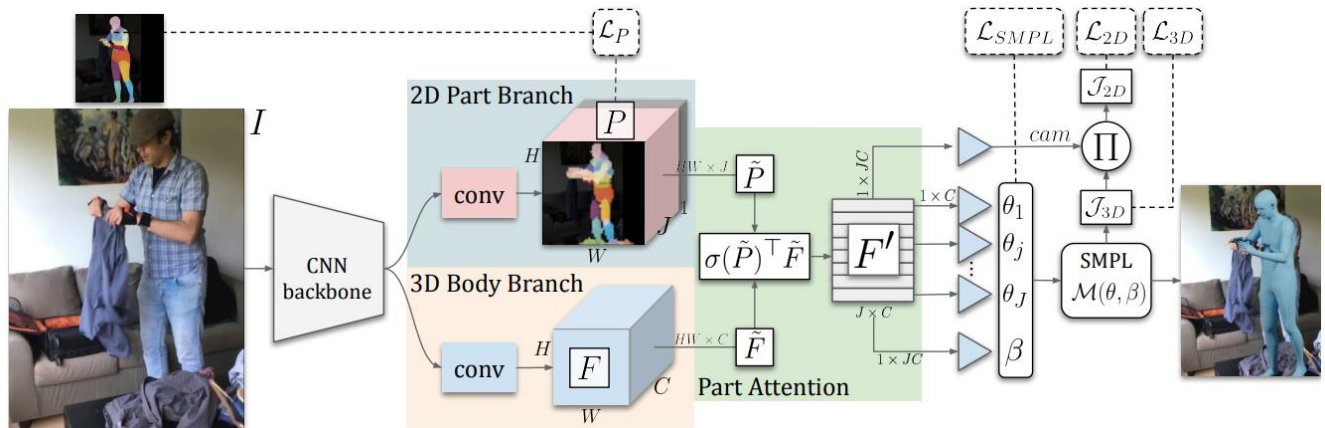


Рисунок 2.1 – Архітектура моделі PARE

Ми беремо повний тензор  $F'$  для того, щоб регресувати форму тіла  $\beta$  та отримати слабо перспективну модель камери з параметрами масштабу та трансляції  $[s, t], t \in \mathbb{R}^2$ , а в кожному рядку  $F'_j$ , також надсилається до різних MLP для прогнозування обертання кожної частини,  $\theta_j$ , параметризований як 6D вектор.

Загалом наші загальні втрати становлять:

$$\mathcal{L} = \lambda_{3D}\mathcal{L}_{3D} + \lambda_{2D}\mathcal{L}_{2D} + \lambda_{SMPL}\mathcal{L}_{SMPL} + \lambda_P\mathcal{L}_P, \quad (2.2)$$

де кожен терм обчислюється як:

$$\begin{aligned} \mathcal{L}_{3D} &= \|J_{3D} - \hat{J}_{3D}\|_F^2, \\ \mathcal{L}_{2D} &= \|J_{2D} - \hat{J}_{2D}\|_F^2, \\ \mathcal{L}_{SMPL} &= \|\Theta - \hat{\Theta}\|_2^2. \end{aligned} \quad (2.3)$$

$$\mathcal{L}_P = \frac{1}{HW} \sum_{h,w} CrossEntropy(\sigma(P_{h,w}), \hat{P}_{h,w}), \quad (2.4)$$

де  $\hat{x}$  представляє основну істину для відповідної змінної  $x$ .

Щоб обчислити втрату двовимірної ключової точки, нам потрібно розташування з'єднань SMPL 3D  $J_{3D}(\theta, \beta) = WM(\theta, \beta)$ , що обчислюються з вершин тіла за допомогою попередньо підготовленого лінійного регресора  $W$ . За допомогою передбачуваної камери зі слабкою перспективою обчислюємо 2D-проекцію 3D-суглобів  $J_{3D}$ , як  $J_{2D} \in \mathbb{R}^{J \times 2} = s\Pi(RJ_{3D}) + t$ , де  $R \in SO(3)$  це матриця обертання камери, а  $\Pi$  – ортографічна проекція,  $\lambda$  – скалярний коефіцієнт для збалансування умов втрат. Дозволяє  $P_{h,w} \in \mathbb{R}^{1 \times 1 \times (J \times 1)}$  позначають фібр  $P$  у цьому місці  $(h, w)$  і  $\hat{P}_{h,w} \in \{0, 1\}^{(J \times 1)}$  позначає основну істину частину мітки в тому самому місці, яка виражена у вигляді одного гарячого вектора. Втрата часткової сегментації  $\mathcal{L}_P$  є втратою крос-ентропії між  $P_{h,w}$  після softmax і  $\hat{P}_{h,w}$ , усереднене по  $H \times W$  елементів. Цей softmax нормалізується вздовж волокна  $P_{h,w}$ , тоді як у формулі 2.1 нормалізується по зрізу  $P_j$ .

## 2.2 Деталі реалізації

Як згадувалося вище, нагляд за частинами тіла через  $\mathcal{L}_P$  застосовується до тензора уваги  $P$  лише на початковому етапі навчання. Пізніше його видаляють, установивши  $\lambda_P$  на нуль, перетворивши механізм уваги в неконтрольовану чисту м'яку увагу. Відсутність частин тіла через оклюзію є основною мотивацією для цієї схеми навчання. Встановлення  $\lambda_P$  до нуля дозволяє механізму уваги також розглядати пікселі за межами самого тіла. Отже, остаточні карти уваги не обов'язково (і часто не нагадують) частину тіла сегментації, як показано на рисунку 2.2. Якщо видно частину тіла, алгоритм фокусується безпосередньо на цій частині; якщо вона закрита, увага може використовувати інші інформативні області зображення.

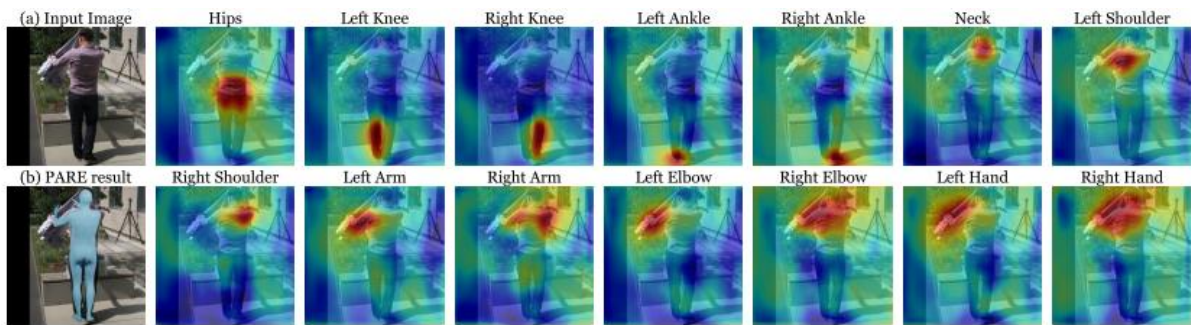


Рисунок 2.2 – Візуалізація роботи PARE

Ми оцінюємо як ResNet-50, так і HRNet-W32 мережі як магістраль. Оскільки ResNet-50 широко використовується в інших методах SOTA вибираємо його як магістраль за замовчуванням для більшості експериментів, якщо інше не зазначено інакше. Витягуємо обсяги функцій  $7 \times 7 \times 2048$  перед об'єднанням глобального середнього. Для гілок 2D і 3D, використовуємо три рази підвищення дискретизації  $2 \times$ , а потім згортку  $3 \times 3$  шари, нанесені з пакетною нормою та ReLU. Кількість ядер становить 256. Для HRNet-W32, оскільки він уже забезпечує об'ємні функції з вищою роздільною здатністю, лише використовуємо два згорткових шари  $3 \times 3$ , нанесені з пакетною нормою і ReLU як гілки 2D та 3D.

Щоб отримати карти уваги частин, застосовуємо  $J + 1 \times 1$  згорткових ядер до 2D-об'єктів частини, щоб зменшити канал виміру. Після отримання  $J \times C$  кінцевої характеристики  $F'$ , використовуємо окремі лінійні шари для прогнозування обертання кожного суглоба SMPL  $\theta_j$ . Регресуємо форму та параметри камери з сплюснутого  $F'$  вектору. Використовуємо фіксований розмір зображення  $224 \times 224$  для всіх експериментів. Оптимізатор Adam з навчанням швидкість  $5 \times 10^{-5}$  і розмір партії 64 використовується для оптимізації нашої моделі. PARE можна наскрізно навчати за один етап, на відміну від інших багатоетапних методів.

### 3 ВИЯВЛЕННЯ 3D-ОБ'ЄКТІВ У РЕАЛЬНОМУ ЧАСІ ЗА ДОПОМОГОЮ MEDIAPIPE

Виявлення об'єктів – це широко досліджена проблема комп'ютерного зору, але більшість досліджень зосереджено на передбаченні 2D-об'єктів. У той час як 2D-прогнозування забезпечує лише 2D-обмежувальні прямокутники, розширивши передбачення до 3D, можна зафіксувати розмір, положення та орієнтацію об'єкта у просторі, що веде до різноманітних застосувань у робототехніці, безпілотних транспортних засобах, пошуку зображень і доповненій реальності. Хоча виявлення 2D-об'єктів є відносно зрілим і широко використовується в промисловості, виявлення 3D-об'єктів із 2D-зображень є складною проблемою через брак даних і різноманітність зовнішніх виглядів і форм об'єктів у межах категорії.

Одним з можливих вирішень цієї проблеми є створення конвеєра виявлення 3D-об'єктів у режимі реального часу для повсякденних об'єктів за допомогою MediaPipe [14]. Цей конвеєр виявляє об'єкти на 2D-зображеннях і оцінює їхні пози та розміри за допомогою моделі машинного навчання (ML), навченої на щойно створеному наборі 3D-даних. Реалізований у MediaPipe, міжплатформовій структурі з відкритим вихідним кодом для побудови конвеєрів для обробки перцептивних даних різних модальностей, Objectron обчислює орієнтовані 3D обмежувальні рамки об'єктів у режимі реального часу (рисунок 3.1).

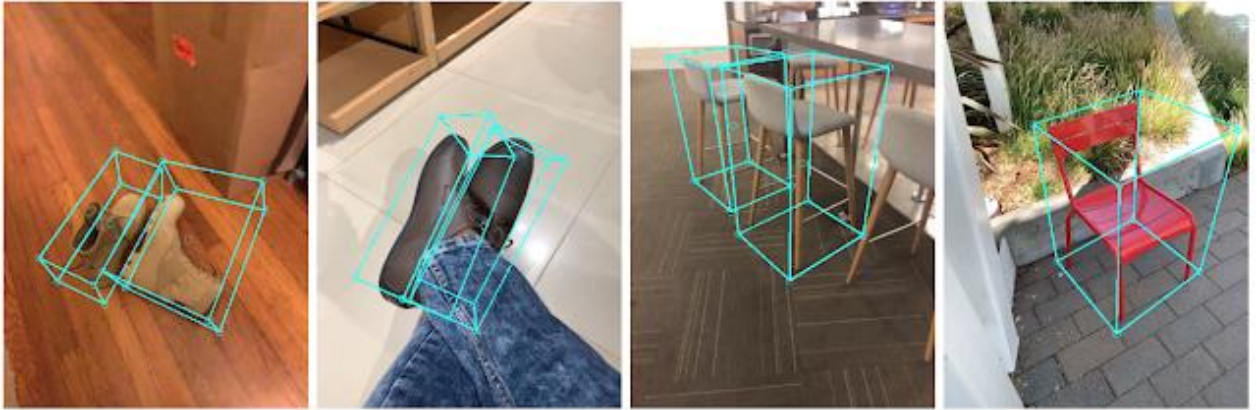


Рисунок 3.1 – Робота програми

### 3.1 Отримання 3D-навчальних даних у реальному світі

Хоча існує велика кількість 3D-даних для вуличних сцен, завдяки популярності досліджень безпілотних автомобілів, які покладаються на датчики 3D-зйомки, такі як LIDAR , набори даних із реальними 3D-анотаціями для більш детальних повсякденних об'єктів вкрай обмежені. Щоб подолати цю проблему, розроблено новий конвєр даних, що використовує дані сеансу мобільної доповненої реальності (AR). З появою ARCore та ARKit сотні мільйонів смартфонів тепер мають можливості доповненої реальності та здатність отримувати додаткову інформацію під час сеансу доповненої реальності, зокрема позу камери, розріджені тривимірні хмари точок , оцінене освітлення та площини поверхні [4].

Для того, щоб позначати основні правдиві дані, створено новий інструмент (рисунок 3.1.2) анотації для використання з даними сеансу AR, який дозволяє анотаторам швидко позначати 3D обмежувальні прямокутники для об'єктів. Цей інструмент використовує перегляд розділеного екрана для відображення 2D відеокадрів, на які накладаються 3D обмежувальні прямокутники ліворуч, поряд із видом, що показує 3D хмари точок, положення камери та виявлені площини



праворуч. Анотатори малюють тривимірні обмежувальні прямокутники в 3D-виді та перевіряють їх розташування, переглядаючи проекції у кадрах двовимірного відео. Для статичних об'єктів нам потрібно лише анотувати об'єкт в одному кадрі та поширювати його розташування на всі кадри, використовуючи інформацію про пози наземної камери правдивості з даних сеансу AR, що робить процедуру дуже ефективною.

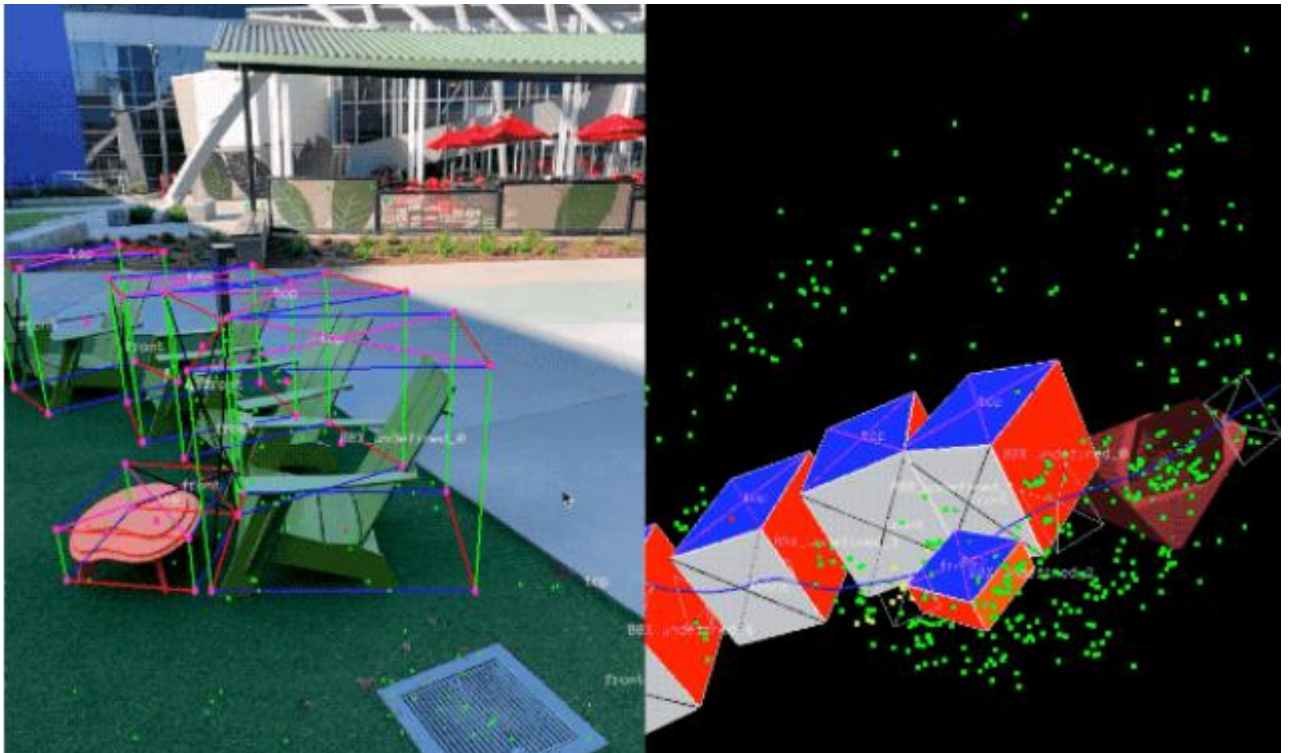


Рисунок 3.2 – Анотація реальних даних для виявлення 3D-об'єктів

### 3.2 Генерація синтетичних даних AR

Популярним підходом є доповнення даних реального світу синтетичними даними, щоб підвищити точність прогнозу. Однак спроби зробити це часто дають погані, нереалістичні дані або, у випадку фотореалістичної візуалізації, вимагають значних зусиль і обчислень. Новий підхід під назвою «генерація синтетичних

даних доповненої реальності» розміщує віртуальні об'єкти в сценах, які мають дані сеансу доповненої реальності, що дозволяє нам використовувати пози камери, виявлені площини поверхні та оцінене освітлення для створення фізично вірогідних місць розташування з освітленням, яке відповідає сцені. Цей підхід (рисунок 3.3) дозволяє отримати високоякісні синтетичні дані із відтвореними об'єктами, які дотримуються геометрії сцени та ідеально підходять до реального фону. Поєднуючи дані реального світу та синтетичні дані AR, можемо підвищити точність приблизно на 10%.

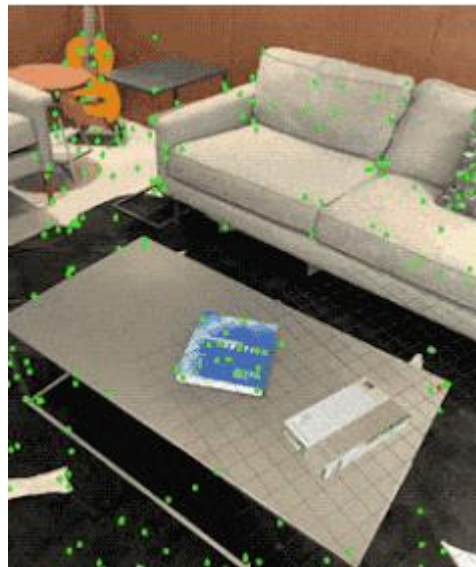


Рисунок 3.3 – Приклад генерації синтетичних даних AR

### 3.3 Конвеєр ML для виявлення 3D-об'єктів

Була створена одноетапна модель для прогнозування пози та фізичного розміру об'єкта на основі одного зображення RGB. Основа моделі має архітектуру кодера-декодера, побудовану на MobileNetv2. Використовуємо багатозадачний підхід до навчання, спільно прогнозуючи форму об'єкта з виявленням і регресією. Завдання форми прогнозує сигнали форми об'єкта залежно від того, яка анотація

базової правди доступна, наприклад сегментація. Це необов'язково, якщо в навчальних даних немає анотації форми. Для завдання виявлення використовуємо анотовані обмежувальні прямокутники та застосовуємо метод Гауса до прямокутника з центром у центроїді прямокутника та стандартними відхиленнями, пропорційними розміру поля. Метою виявлення є прогнозування цього розподілу з піком, що представляє центральне розташування об'єкта. Завдання регресії оцінює двовимірні проекції восьми вершин обмежувальної рамки. Щоб отримати остаточні 3D-координати для обмежувальної рамки, використовуємо добре встановлений алгоритм оцінки пози (EPnP). Він може відновити 3D обмежувальну рамку об'єкта без попереднього знання розмірів об'єкта. Враховуючи 3D обмежувальну рамку, можемо легко обчислити позу та розмір об'єкта. На рисунку 3.4 показано архітектуру мережі та постобробку. Модель досить легка, щоб працювати в режимі реального часу навіть на мобільних пристроях (зі швидкістю 26 FPS на мобільному графічному процесорі Adreno 650).

Приклади результатів роботи мережі зображено на рисунку 3.5.

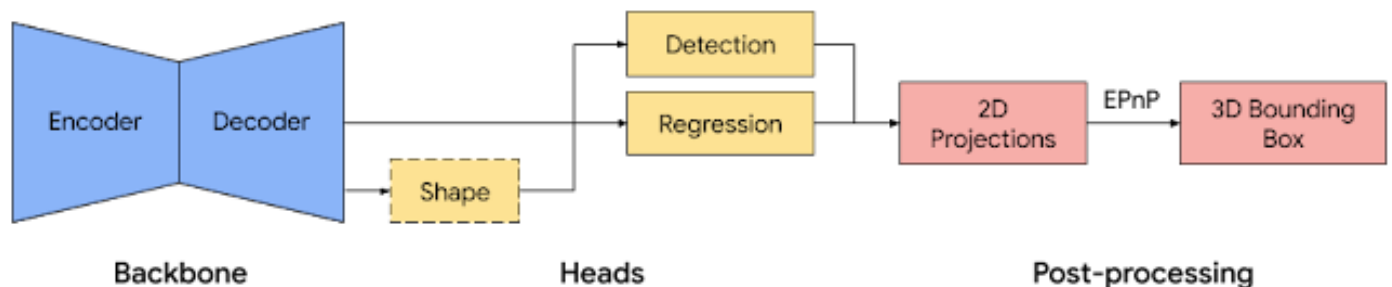


Рисунок 3.4 – Архітектура мережі та постобробка для виявлення 3D-об'єктів

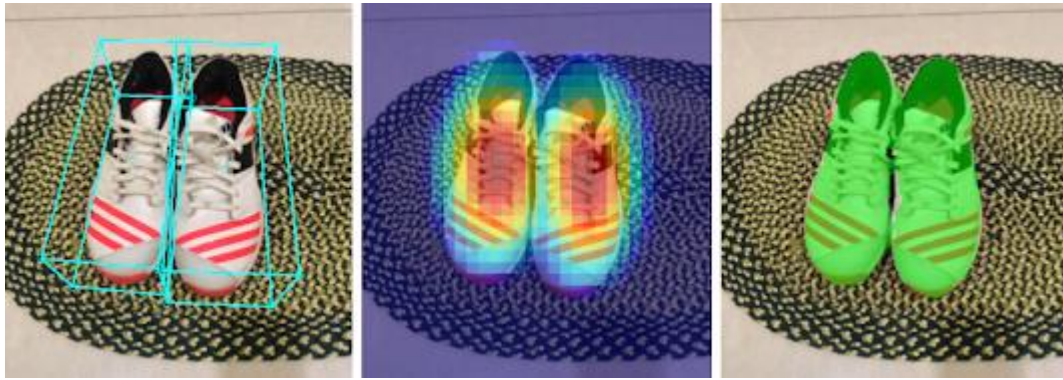


Рисунок 3.5 – Приклади результатів мережі

### 3.4 Виявлення та відстеження в MediaPipe

Коли модель застосовується до кожного кадру, захопленого камерою, вона може страждати від тремтіння через неоднозначність 3D-обмежувальної рамки, оціненої в кожному кадрі. Щоб пом'якшити це, використовуємо фреймворк виявлення та відстеження. Цей фреймворк зменшує потребу запускати мережу на кожному кадрі, дозволяючи використовувати важчі й, отже, точніші моделі, зберігаючи конвеєр у режимі реального часу. Він також зберігає ідентичність об'єкта в кадрах і гарантує, що передбачення є узгодженим у часі, зменшуючи тремтіння.

Для підвищення ефективності нашого конвеєра запускаємо наш вивід моделі лише раз на кілька кадрів. Далі беремо прогноз і відстежуємо його з часом, використовуючи підхід для миттєвого відстеження руху та Motion Stills. Коли робиться новий прогноз, консолідуємо результат виявлення з результатом відстеження на основі області перекриття.

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Мова програмування

Одна з поширених помилок щодо Python, яка продовжує зберігатися, полягає в тому, що Python повільний. Це помилкове уявлення корениться в тому факті, що інтерактивні версії Python, які використовують інтерпретатор, і стандартний Python, який використовує вбудований компілятор під назвою CPython, справді повільні. Але хоча інтерпретатори Python і мова Python можуть бути повільнішими, ніж Fortran або C, код виконання Python не обов'язково повільний. Наукові обчислювальні пакети, такі як SciPy і NumPy, не мають багатьох недоліків стандартного Python.

Крім того, існують інші основні реалізації Python, крім стандартного Python. Ці реалізації, відомі як дистрибутиви, насправді можуть використовуватися більш широко, ніж стандартний дистрибутив Python. Крім того, ви можете скомпілювати Python для прискорення виконання. Деякі реалізації компілятора, такі як компілятор Just-in-Time (JIT) PyPy, можуть створювати код виконання, який може працювати так само швидко або швидше, ніж C.

Мова програмування Python широко використовується для наукових досліджень, ось деякі з причин її популярності у цій сфері [5]:

- Python має вбудовану підтримку наукових обчислень. Більшість дистрибутивів Python включають екосистему SciPy (з відкритим кодом), яка включає SciPy (бібліотеку SciPy), пакет числових обчислень під назвою NumPy і кілька незалежних наборів інструментів, кожен з яких відомий як Scikits. Matplotlib, бібліотека 2D-графіків для візуалізації, також є частиною екосистеми SciPy. Matplotlib дуже схожий на MATLAB з точки зору його функціональності та використання, але є відкритим кодом;

- Python має мости до MATLAB або Octave . Python може інсталиювати MATLAB Engine API, щоб програми Python могли викликати MATLAB як обчислювальний механізм. Програми MATLAB також можуть викликати функції Python з деякими обмеженнями. Деякі дистрибутиви Python можуть використовувати пакети Python, такі як Rumatbridge, які можуть підтримувати як MATLAB, так і Octave і дозволяють використовувати MATLAB у Jupyter Notebooks;

- Python – це мова з широкими можливостями розширення . Розробники багато років використовували оболонки Python для програм C/C++. Python використовує зовнішній функціональний інтерфейс C для Python (cFFI) для безпосередньої взаємодії з кодом C. Такі інструменти, як SWIG, спрощують це. Розробники також можуть викликати підпрограми Fortran з Python за допомогою пакета генератора інтерфейсу Fortran для Python, F2Py, який тепер є частиною NumPy;

- Python має дуже хороші параметри введення/виведення (I/O) . До недавнього часу введення/виведення Fortran традиційно базувалося на записах. Навпаки, Python вже давно підтримує кілька варіантів для введення-виведення та багато додаткових пакетів для підтримки всіх типів форматів введення-виведення, включаючи формати реального часу та потокові формати;

- Python має сильну підтримку для автоматизації завдань . Вбудовані функції сценаріїв Python і численні пакети мають сильну підтримку для автоматизації завдань. Автоматизація повторюваних завдань і реєстрація даних є простими та не потребують великих зусиль;

- Python може використовувати веб-інтерфейс . Пакети Python, такі як Django та Flask, дозволяють розробляти та використовувати Python як API із веб-інтерфейсом. Ця функція особливо корисна при використанні хмарної інфраструктури як платформи для доступу до систем високопродуктивних обчислень (HPC).

## 4.2 Використані бібліотеки

### 4.2.1 CUDA

Обробка великого обсягу даних в реальному часі є досить не простою задачею для комп'ютера, тому необхідного максимально ефективно задіяти всі доступні ресурси, наприклад графічну карту.

CUDA – це паралельна обчислювальна платформа та модель програмування, розроблена NVIDIA для загальних обчислень на власних GPU (графічних процесорах) [9]. CUDA (рисунок 4.1) дозволяє розробникам пришвидшити інтенсивні обчислювальні програми, використовуючи потужність графічних процесорів для розпаралелюваної частини обчислень. Хоча були й інші запропоновані API для графічних процесорів, як-от OpenCL, і конкурентоспроможні графічні процесори від інших компаній, наприклад, AMD, поєднання графічних процесорів CUDA та NVIDIA домінує в кількох сферах застосування, включаючи глибоке навчання, і є основою для деяких з найшвидших комп'ютерів в світі [6].

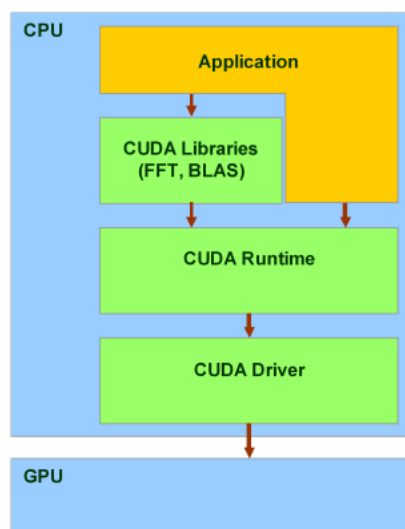


Рисунок 4.1 – Робота CUDA

Набір інструментів CUDA містить бібліотеки, засоби налагодження та оптимізації, компілятор, документацію та бібліотеку середовища виконання для розгортання програм [8]. Він містить компоненти, які підтримують глибоке навчання, лінійну алгебру, обробку сигналів і паралельні алгоритми. Загалом бібліотеки CUDA підтримують усі сімейства графічних процесорів NVIDIA, але найкраще працюють на останньому поколінні, наприклад V100, який може бути втричі швидшим, ніж P100, для навчальних навантажень глибокого навчання; A100 може додати ще 2-кратне прискорення. Використання однієї або кількох бібліотек – це найпростіший спосіб скористатися перевагами графічних процесорів, за умови, що потрібні вам алгоритми реалізовано у відповідній бібліотеці [7].

#### 4.2.2 NumPy

NumPy – це бібліотека для мови програмування Python, яка додає підтримку для великих багатовимірних масивів та матриць разом із великою колекцією математичних функцій високого рівня для роботи з цими масивами. Попередник NumPy, Numeric, спочатку був створений Джимом Хугуніним за участю кількох інших розробників. У 2005 році Тревіс Оліфант створив NumPy, включивши функції конкуруючого Numarray у Numeric із великими змінами. NumPy є програмним забезпеченням з відкритим вихідним кодом і має багато учасників. NumPy – це проект, який фінансується NumFOCUS.

NumPy націлений на еталонну реалізацію Python CPython, яка є неоптимізованим інтерпретатором байт-коду. Математичні алгоритми, написані для цієї версії Python, часто працюють набагато повільніше, ніж скомпільовані еквіваленти через відсутність оптимізації компілятора. NumPy частково вирішує проблему повільності, надаючи багатовимірні масиви, функції та оператори, які ефективно працюють з масивами; їх використання вимагає переписування деякого



коду, переважно внутрішніх циклів, за допомогою NumPy. Використання NumPy у Python надає функціональність, порівнянну з MATLAB, оскільки вони обидва інтерпретуються, і вони обидва дозволяють користувачеві писати швидкі програми, якщо більшість операцій працюють з масивами чи матрицями замість скалярів. Для порівняння, MATLAB може похвалитися великою кількістю додаткових наборів інструментів, зокрема Simulink, тоді як NumPy внутрішньо інтегровано з Python, більш сучасною та повною мовою програмування. Крім того, доступні додаткові пакети Python; SciPy – це бібліотека, яка додає більше функціональних можливостей, подібних до MATLAB, а Matplotlib (рисунок 4.2) – це пакет графіки, який надає функціональні можливості побудови, подібні до MATLAB. Внутрішньо і MATLAB, і NumPy покладаються на BLAS і LAPACK для ефективних обчислень лінійної алгебри. Прив'язки Python широко використовуваної бібліотеки комп'ютерного зору OpenCV використовують масиви NumPy для зберігання та роботи з даними. Оскільки зображення з декількома каналами просто представлені як тривимірні масиви, індексація, нарізка або маскування за допомогою інших масивів є дуже ефективними способами доступу до конкретних пікселів зображення. Масив NumPy як універсальна структура даних у OpenCV для зображень, вилучених точок функцій, ядер фільтрів і багато іншого, значно спрощує робочий процес програмування та налагодження [10].

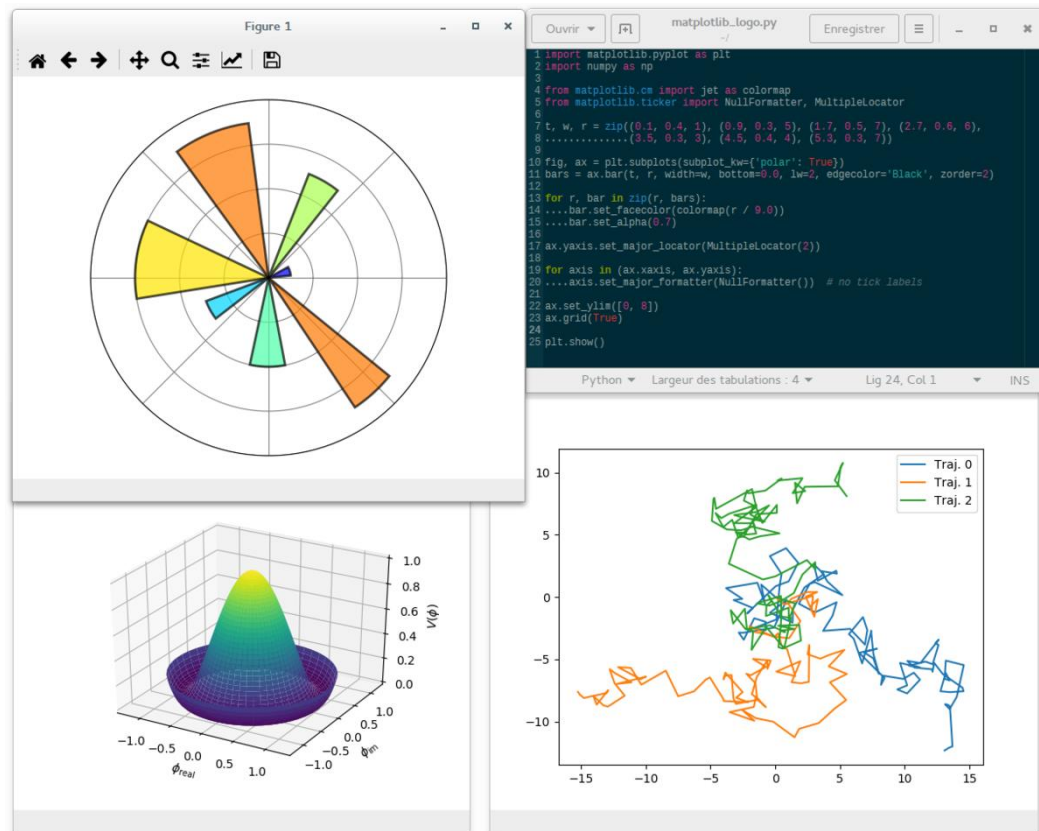


Рисунок 4.2 – Графіки побудовані в Matplotlib

### 4.2.3 Torch

Torch — це бібліотека машинного навчання з відкритим вихідним кодом, наукова обчислювальна основа та мова сценаріїв на основі мови програмування Lua. Він надає широкий спектр алгоритмів для глибокого навчання та використовує мову сценаріїв LuaJIT і базову реалізацію C. Він був створений в IDIAP при EPFL . Станом на 2018 рік Torch більше не знаходиться в активній розробці. Проте PyTorch, який базується на бібліотеці Torch, активно розробляється станом на жовтень 2022 року [12].

Основним пакетом Torch є `torch`. Він забезпечує гнучкий N-вимірний масив або Tensor, який підтримує базові процедури для індексування, нарізання, транспонування, приведення типу, зміни розміру, спільного використання пам'яті

та клонування. Цей об'єкт використовується більшістю інших пакунків і, таким чином, утворює основний об'єкт бібліотеки. Tensor також підтримує математичні операції, такі як max, min, sum, статистичні розподіли, такі як рівномірний, нормальний і багаточленний, а також операції BLAS, такі як скалярний добуток, множення матриця-вектор, множення матриця-матриця та матричний добуток.

### 4.3 Отримані результати за допомогою PARE

Для початку роботи необхідно запустити відповідний скрипт, в процесі виконання якого буде перевірена наявність всіх необхідних бібліотек та розпочнеться обробка кадрів отриманих за допомогою вебкамери (рис 4.3).

```
alex@2f54518976c7:~/work/PARE$ ./run_pare_demo.sh
2022-12-10 09:54:12.790 | INFO      | __main__:main:83 - Frames are already extra
cted in "logs/demo/sample_video_/tmp_images"
logs/demo/sample_video_/tmp_images
2022-12-10 09:54:12.801 | INFO      | __main__:main:116 - Demo options:
Namespace(batch_size=16, beta=1.0, cfg='data/pare/checkpoints/pare_w_3dpw_conf
ig.yaml', ckpt='data/pare/checkpoints/pare_w_3dpw_checkpoint.ckpt', detector='yol
o', display=False, draw_keypoints=False, exp='', image_folder=None, min_cutoff=0
.004, mode='video', no_render=False, no_save=False, output_folder='logs/demo', s
ave_obj=False, sideview=False, smooth=False, staf_dir='/home/mkocabas/developmen
ts/openposetrack', tracker_batch_size=12, tracking_method='bbox', vid_file='data
/sample_video.mp4', wireframe=False, yolo_img_size=416)
2022-12-10 09:54:13.215 | INFO      | pare.models.backbone.hrnet:init_weights:530
- => init weights from normal distribution
2022-12-10 09:54:13.556 | WARNING   | pare.models.backbone.hrnet:init_weights:558
- IMPORTANT WARNING!! Please download pre-trained models if you are in TRAINING
mode!
2022-12-10 09:54:13.557 | INFO      | pare.models.head.pare_head:__init__:125 - "
Keypoint Attention" should be activated to be able to use part segmentation
2022-12-10 09:54:13.557 | INFO      | pare.models.head.pare_head:__init__:126 - 0
verriding use_keypoint_attention
2022-12-10 09:54:13.582 | INFO      | pare.models.head.pare_head:__init__:327 - K
eypoint attention is active
WARNING: You are using a SMPL model, with only 10 shape coefficients.
```

Рисунок 4.3 – Початок роботи PARE

Список всіх необхідних бібліотек, які попередньо необхідно встановити, знаходиться у файлі requirements.txt (рисунок 4.4).

```
1  numpy
2  torch==1.6.0
3  torchvision==0.7.0
4  joblib==0.15.1
5  opencv-python==4.1.1.26
6  trimesh==3.8.12
7  pyrender==0.1.36
8  smplx==0.1.28
9  tqdm==4.47.0
10 loguru==0.3.2
11 alumentations==0.4.6
12 scipy==1.7.0
13 pyyaml==5.3.1
14 yacs==0.1.8
15 matplotlib==3.4.2
16 scikit-image==0.17.2
17 pillow==8.2.0
18 pytorch-lightning==1.1.8
19 numba==0.54.1
20 filterpy==1.4.5
21 flatten-dict==0.2.0
22 fastai==1.0.55
23 jpeg4py==0.1.4
24 chumpy==0.70
25 torchgeometry==0.1.2
26 setuptools==47.3.1
27 git+https://github.com/mkocabas/yolov3-pytorch.git
28 git+https://github.com/mkocabas/multi-person-tracker.git
29 git+https://github.com/giacaglia/pytube.git
```

---

Рисунок 4.4 – Зміст файлу requirements.txt

Процес обробки кожного фрейму отриманого з вебкамери зображено на рисунку 4.5.

```
mc [alex@2f54518976c7]:~/work/PARE | 100%| [Progress Bar] | 1/1 [00:00<00:00, 8.90it/s]
Pare time = 0.11333847045898438
2022-12-10 09:47:43.167 | INFO | __main__:main:152 - Saving output results to "logs/demo/sample_video_/pare_output.pkl".
2022-12-10 09:47:43.167 | INFO | pare.core.tester:render_results:487 - Rendering output video, writing frames to logs/demo/sample_video_/tmp_images_output
To here 1 = 0.0004150867462158203
100%| [Progress Bar] | 1/1 [00:00<00:00, 6.62it/s]
Render time = 0.15217876434326172
Running Multi-Person-Tracker
100%| [Progress Bar] | 1/1 [00:00<00:00, 4.83it/s]
Finished. Detection + Tracking FPS 4.82
Tracking time = 0.20793604850769043
2022-12-10 09:47:43.564 | INFO | pare.core.tester:run_on_video:379 - Running PARE on each tracklet...
0%| [Progress Bar] | 0/1 [00:00<?, ?it/s]
2022-12-10 09:47:43.695 | INFO | pare.core.tester:run_on_video:456 - Converting smpl keypoints 2d to original image coordinate
100%| [Progress Bar] | 1/1 [00:00<00:00, 7.62it/s]
Pare time = 0.13229942321777344
2022-12-10 09:47:43.696 | INFO | __main__:main:152 - Saving output results to "logs/demo/sample_video_/pare_output.pkl".
2022-12-10 09:47:43.696 | INFO | pare.core.tester:render_results:487 - Rendering output video, writing frames to logs/demo/sample_video_/tmp_images_output
To here 1 = 0.00044536590576171875
100%| [Progress Bar] | 1/1 [00:00<00:00, 6.68it/s]
Render time = 0.1508796215057373
Running Multi-Person-Tracker
0%| [Progress Bar] | 0/1 [00:00<?, ?it/s]
```

Рисунок 4.5 – Робота PARE в режимі реального часу

Для того щоб бачити кількість кадрів в секунду, що дозволяє нам оброблювати графічний процесор, в програмі передбачений лічильник FPS (рисунок 4.6). Щоб підвищити кількість кадрів необхідно мати високопродуктивну відеокарту.



Рисунок 4.6 – Лічильник FPS

Після всіх перевірок і налаштувань, з'явиться вікно в якому можна побачити зображення з відеокамери з накладеною поверх маскою пози людини (рисунок 4.7).

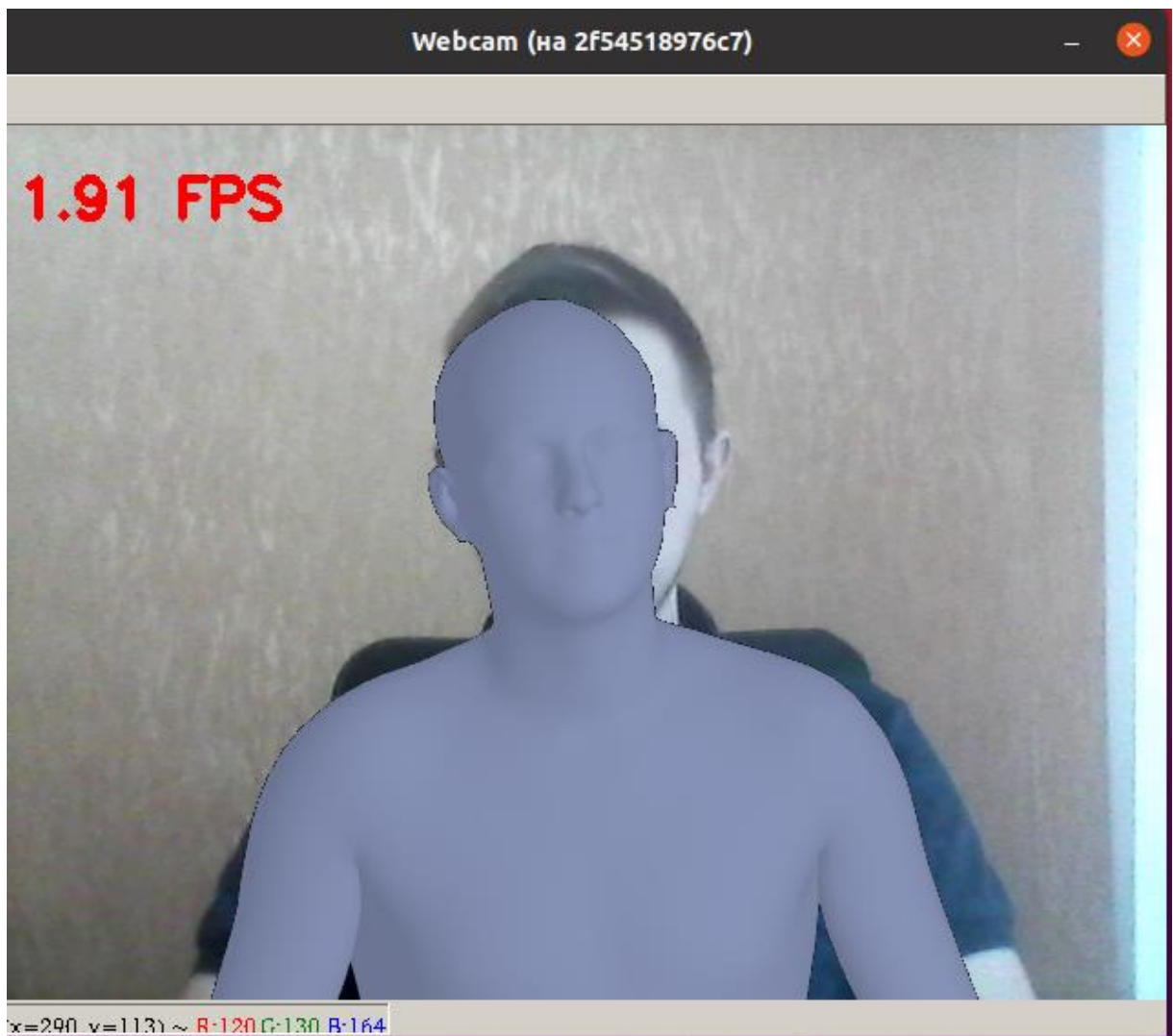


Рисунок 4.7 – Розпізнавання пози людини режимі реального часу

Оскільки обробка кадрів проводиться в реальному часі, при зміні пози на екрані з'явиться відповідна маска (рисунки 4.8, 4.9).

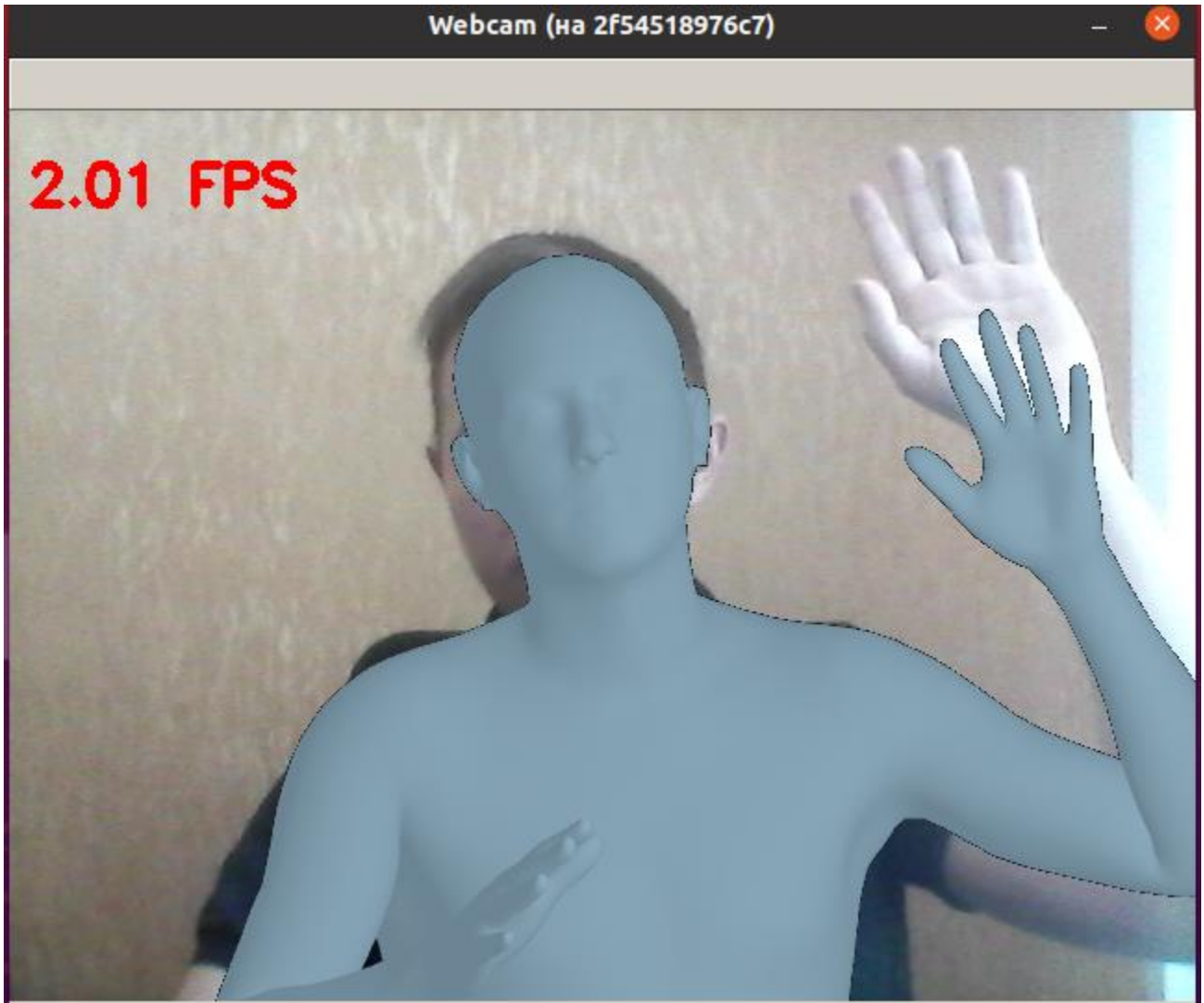


Рисунок 4.8 – Розпізнавання пози, рука піднята догори

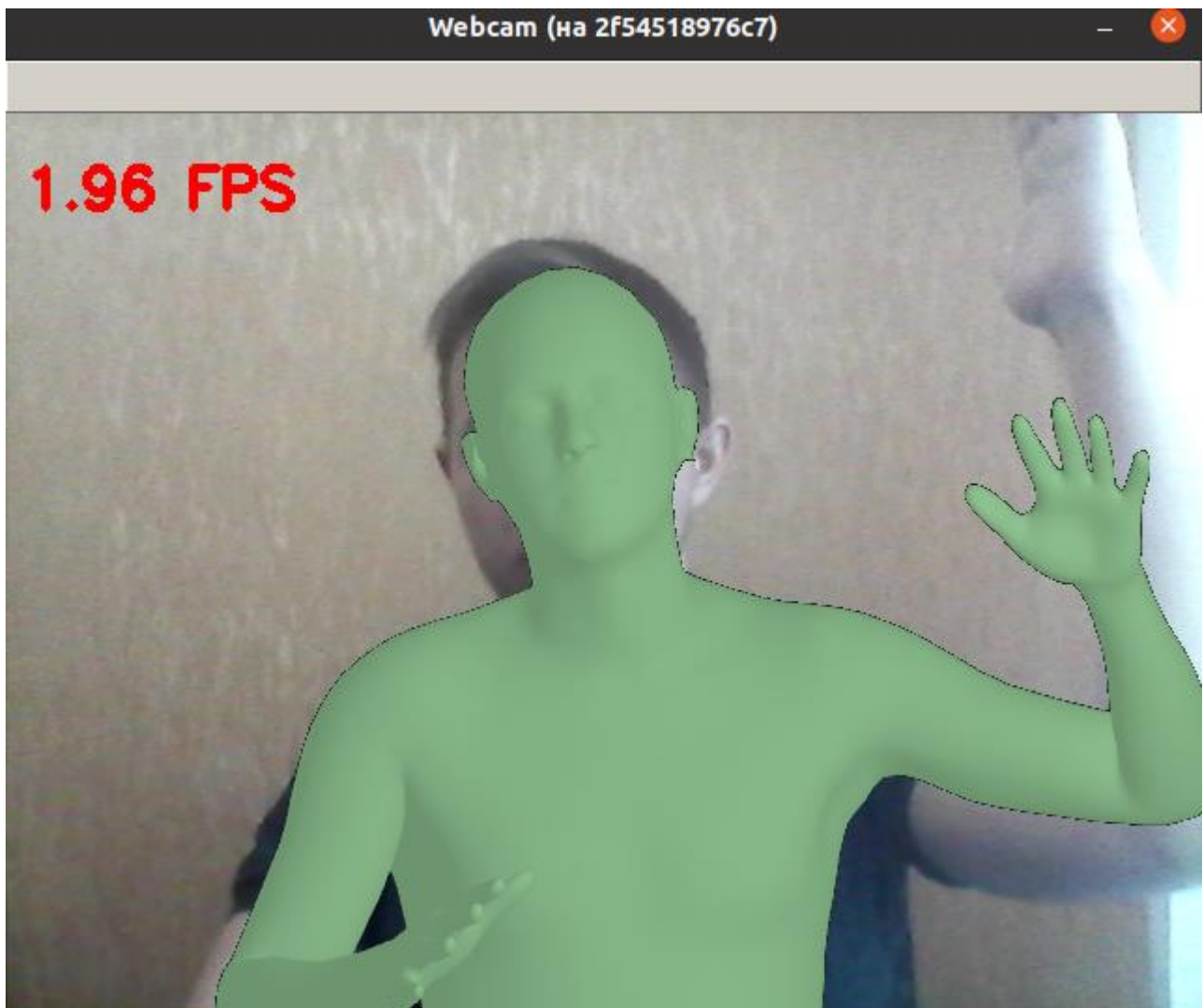


Рисунок 4.9 – Розпізнавання пози, рука піднята догори  
під іншим кутом

Алгоритм PARE відстежує зміни в реальному часі, наприклад, при повороті голови в сторону, модель людини також повертається (рисунок 4.10).



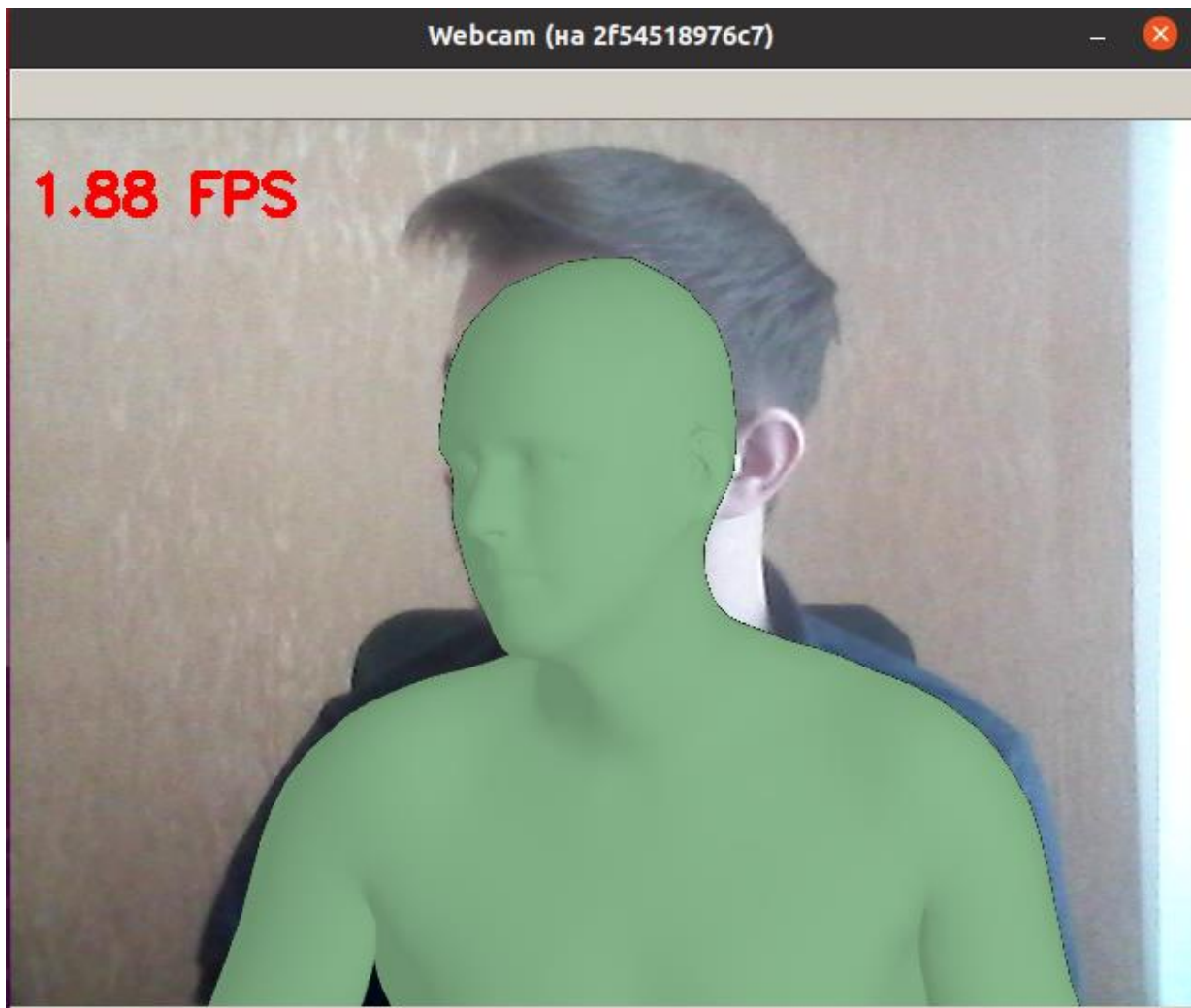


Рисунок 4.10 – Розпізнавання пози, голова повернута в сторону

У більшості випадків під час експериментів на вебкамеру потрапляє верхня частина тіла, але алгоритм налаштований на розпізнавання поз людини в повний зріст (рисунок 4.11).

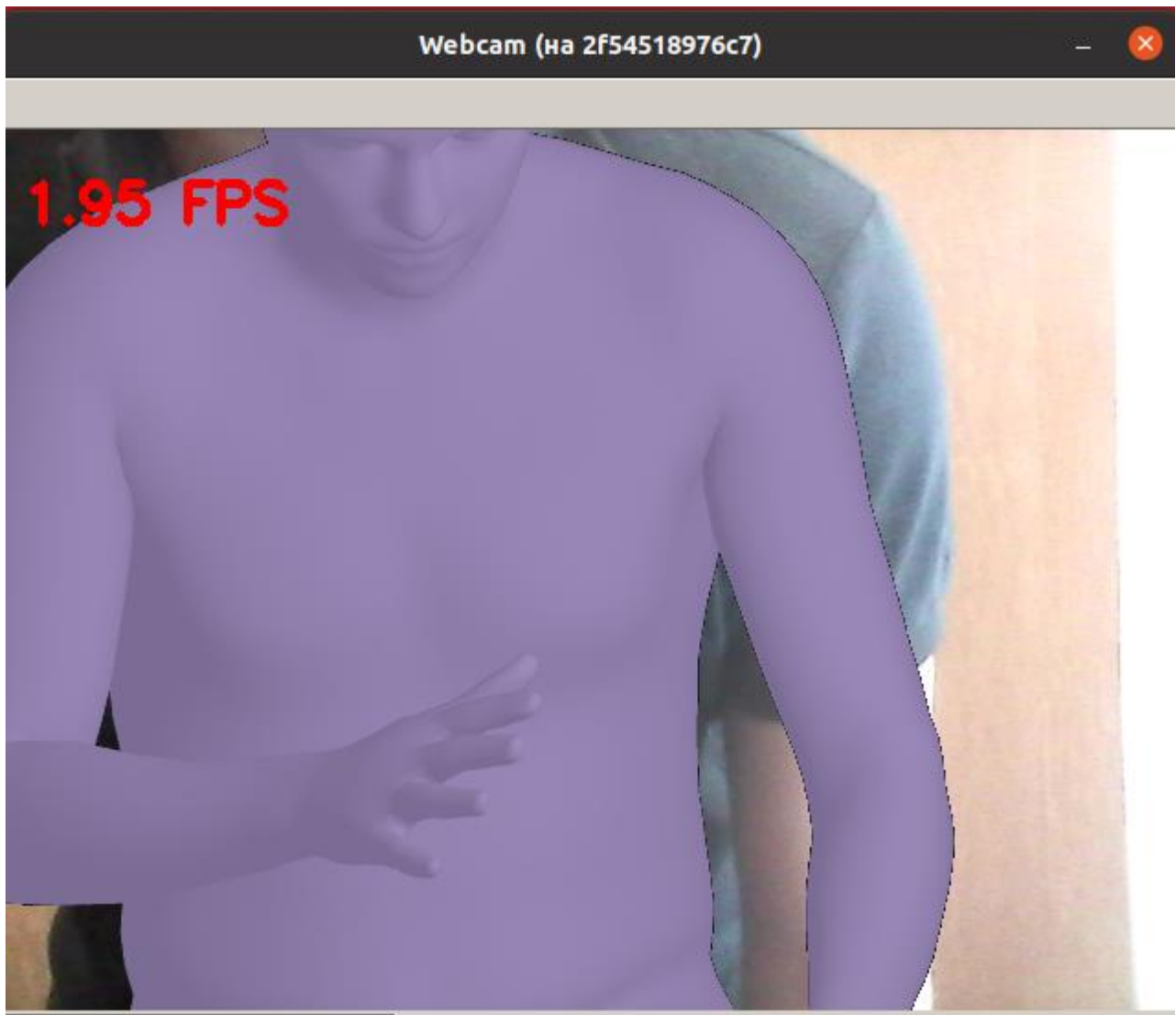


Рисунок 4.11 – Розпізнавання пози, фігура в повний зріст

В процесі зчитування кадрів з вебкамери, при реальних умовах в об'єктив камери можуть потрапляти різноманітні перешкоди, заважаючи тим самим чіткому розпізнаванню об'єктів. Проте, як можемо побачити на рисунках 4.12 та 4.13, алгоритм PARE успішно справляється з цим завданням .

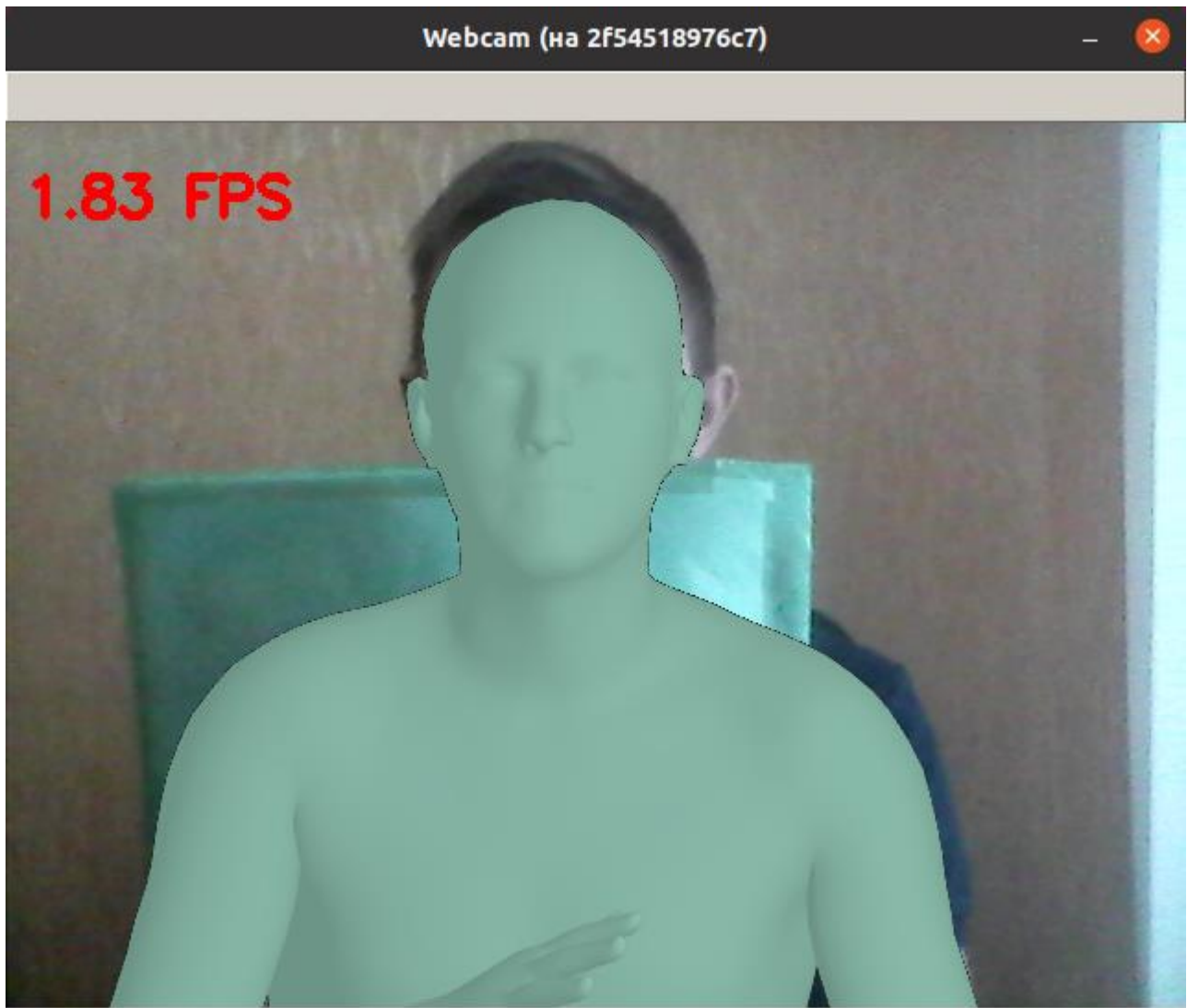


Рисунок 4.12 – Розпізнавання пози з перешкодою у вигляді теки

Тека є одним з прикладів оклюзій, які можуть трапитися при використанні цього метода в реальних умовах, адже людину на фото або відео не завжди чітко видно або видно не всі частини тіла, до чого інші подібні системи досить уразливі.

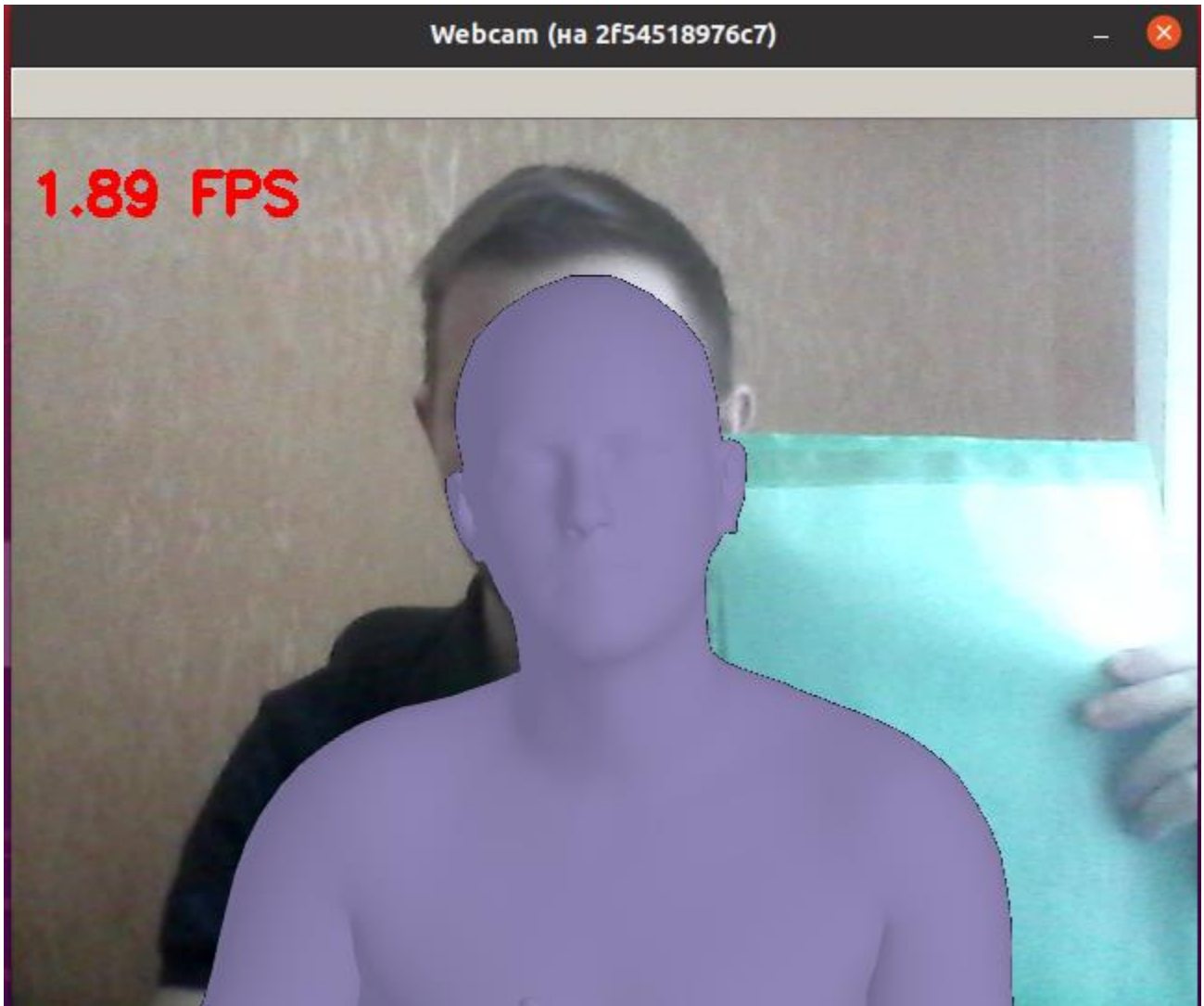


Рисунок 4.13 – Розпізнавання пози з перешкодою у вигляді теки

Щоб додатково перевірити коректність і правильність роботи в реальних умовах необхідні тести з багатьма людьми та перешкодами, оскільки такої можливості не було – перевірено реакцію програми на відсутність людини в кадрі взагалі (рисунок 4.14).



Рисунок 4.14 – Розпізнавання пози, людина відсутня

#### 4.4 Отримані результати за допомогою MediaPipe

Результати розпізнавання та оцінки розташування об'єкту в просторі за допомогою створеної програми в MediaPipe зображено на рисунках 4.15 та 4.16.

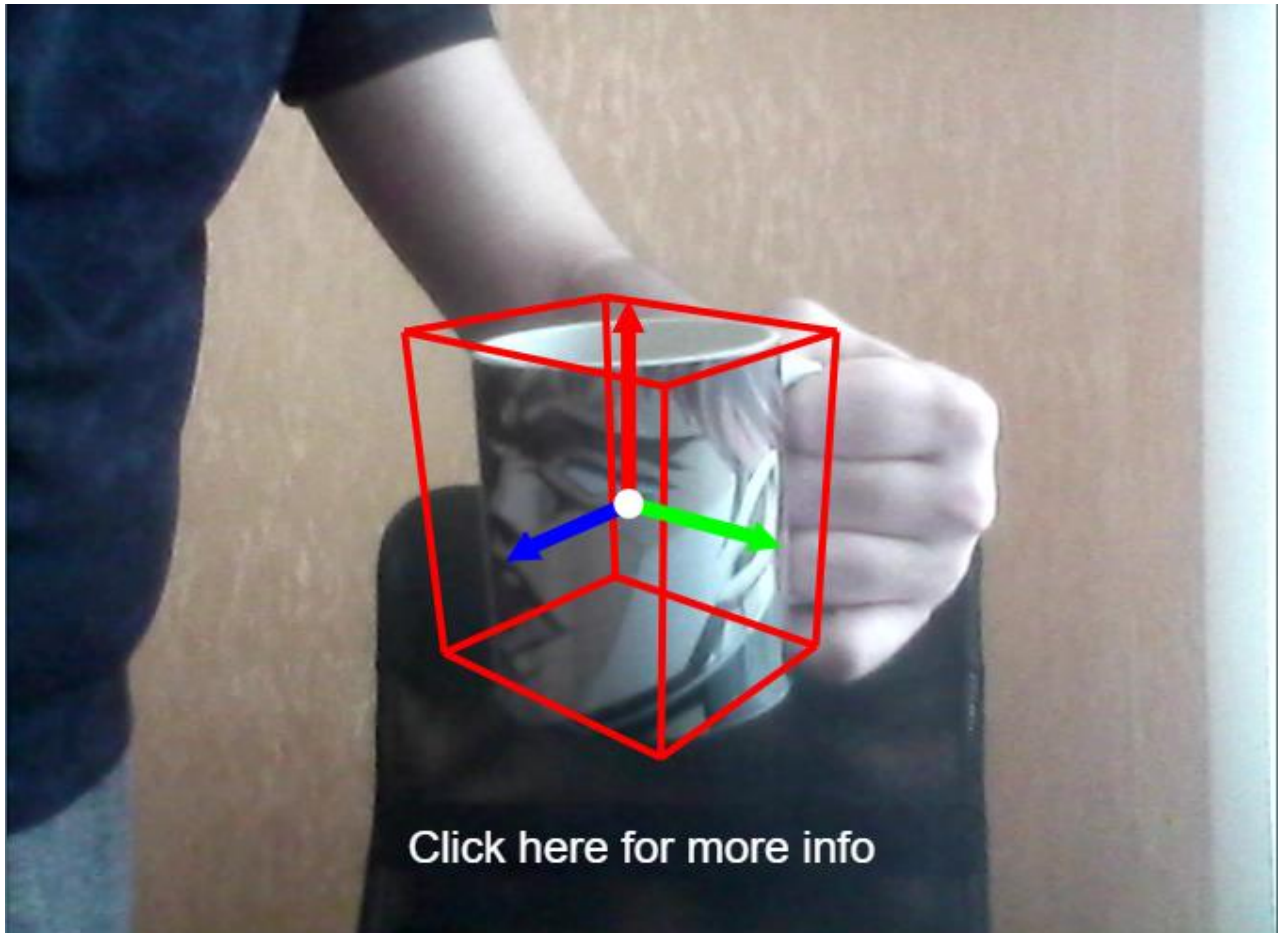


Рисунок 4.15 – Розпізнавання об'єкту (чашки)

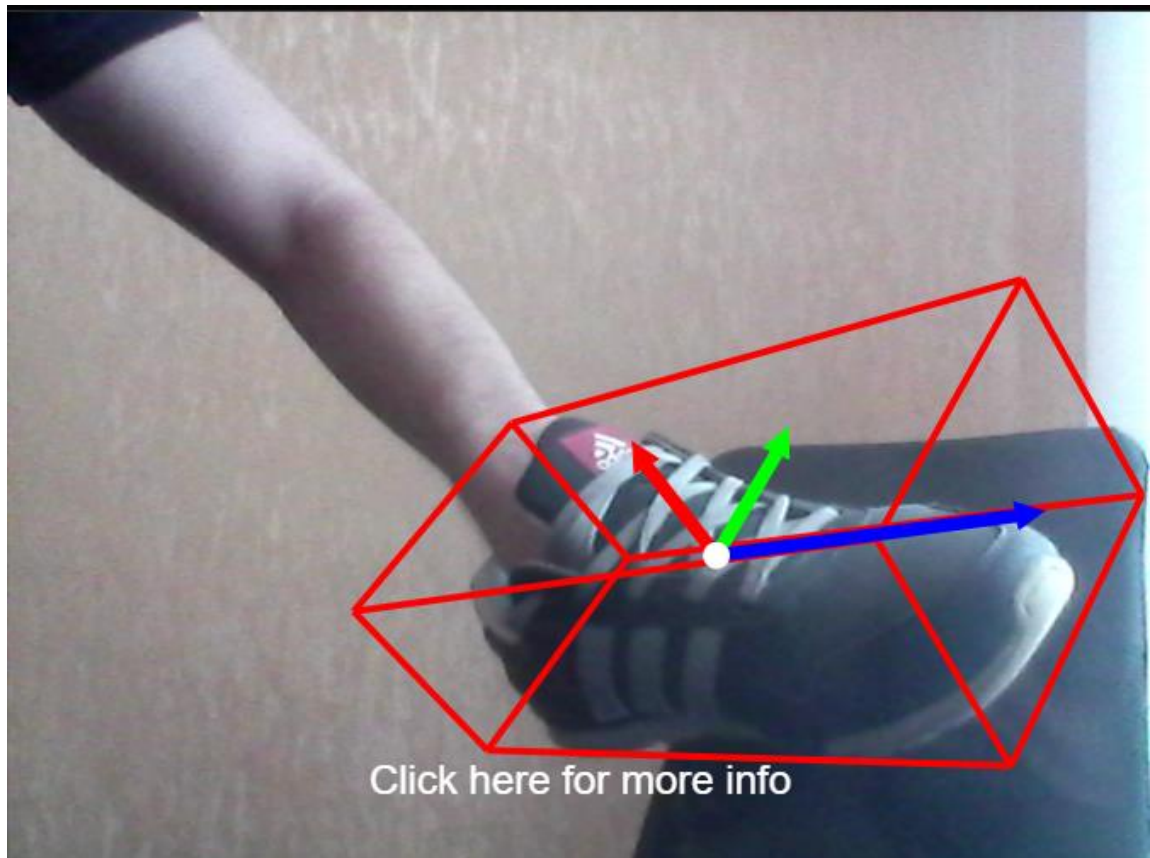


Рисунок 4.16 – Розпізнавання об’єкту (взуття)

Налаштування для різних моделей показані на рисунках 4.17 та 4.18. У вікні налаштувань можемо побачити лічильник FPS, вибір камери, ввімкнення/вимкнення режиму селфі, вибір моделі (в нашому випадку чашка та взуття), максимальну кількість об’єктів на сцені та інше.

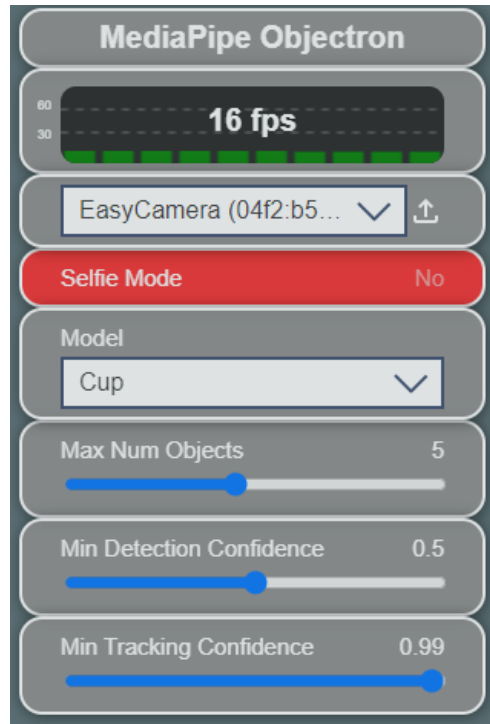


Рисунок 4.17 – Налаштування для розпізнавання об'єкту (чашки)

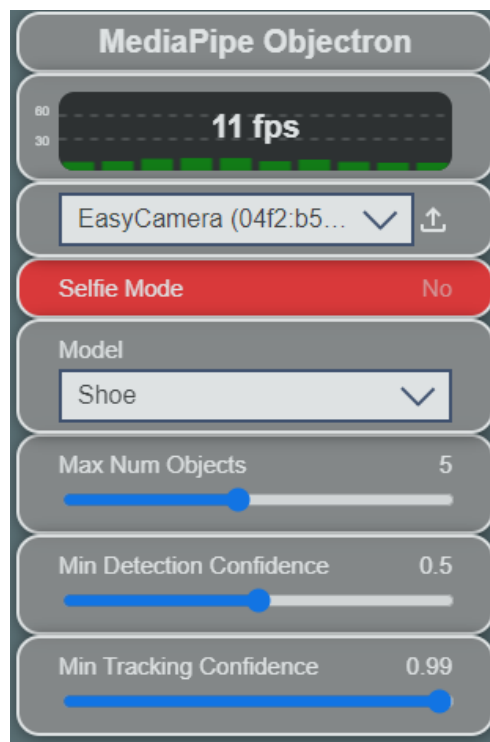


Рисунок 4.18 – Налаштування для розпізнавання об'єкту (взуття)



## ВИСНОВКИ

В ході виконання кваліфікаційної роботи створено комплекс програмного забезпечення для виявлення та розпізнавання тривимірних об'єктів в реальному часі, а саме: розпізнавання поз людей та оцінка положення предметів в просторі та їх розмірів.

Відповідно до мети кваліфікаційної роботи було виконано поставлені задачі:

- проведено аналіз предметної області та підтверджена актуальність розробки програми для виявлення 3D-об'єктів;
- проаналізовано існуючі рішення та методи для вирішення цієї задачі, їх недоліки та переваги;
- створено пакет програмного забезпечення, що втілює ідею розпізнавання тривимірних об'єктів за допомогою мов Python та JavaScript, який може приймати на вхід зображення з вебкамери та в режимі реального часу надавати інформацію про розпізнаний об'єкт.

Метод PARE добре показав себе при використанні в реальних умовах і може працювати ще краще при збільшенні обчислювальних потужностей. Розроблена за допомогою платформи MediaPipe програма також виконує поставлені задачі з виявлення та розпізнавання об'єктів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сорока Д.О. Використання комп'ютерного зору для ідентифікації змісту етикеток харчових продуктів: *Результати розвитку наукової думки 2021*: матеріали III міжнар. студ. конф., м. Вінниця, 24 груд. 2021 р. Вінниця, 2021. С. 15-16.
2. Сорока Д.О. Виявлення 3D-об'єктів у реальному часі на мобільних пристроях за допомогою MediaPipe: *Наука сьогодні: від досліджень до стратегічних рішень*: матеріали V міжнар. студ. конф., м. Київ, 2 груд. 2022 р. Київ, 2022. С. 146-147.
3. 3D object recognition [Електронний ресурс] – Режим доступу : [www/URL: https://en.wikipedia.org/wiki/3D\\_object\\_recognition](http://www/URL: https://en.wikipedia.org/wiki/3D_object_recognition)
4. Real-Time 3D Object Detection on Mobile Devices with MediaPipe [Електронний ресурс] – Режим доступу : [www/URL: https://ai.googleblog.com/2020/03/real-time-3d-object-detection-on-mobile.html?m=1](http://www/URL: https://ai.googleblog.com/2020/03/real-time-3d-object-detection-on-mobile.html?m=1)
5. Python for scientific research [Електронний ресурс] – Режим доступу: [www/URL: https://developer.ibm.com/blogs/use-python-for-scientific-research/](http://www/URL: https://developer.ibm.com/blogs/use-python-for-scientific-research/)
6. What is CUDA? Parallel programming for GPUs [Електронний ресурс] – Режим доступу: [www/URL: https://www.infoworld.com/article/3299703/what-is-cuda-parallel-programming-for-gpus.html](http://www/URL: https://www.infoworld.com/article/3299703/what-is-cuda-parallel-programming-for-gpus.html)
7. Shane Cook CUDA programming. A Developer's guide to parallel computing with GPUs. Elsevier, 2013. – 591p.
8. Rob Farber CUDA application design and development. Elsevier, 2011. – 324p.
9. Jason Sanders CUDA by Example: An Introduction to General-Purpose GPU Programming / Jason Sanders, Edward Kandrot. - Addison-Wesley Professional, 2010. - 312p.

10. NumPy [Электронный ресурс] – Режим доступа: [www/URL: https://en.wikipedia.org/wiki/NumPy](http://www.URL:https://en.wikipedia.org/wiki/NumPy)
11. PARE: Part Attention Regressor for 3D Human Body Estimation [Электронный ресурс] – Режим доступа: [www/URL: https://pare.is.tue.mpg.de/](http://www.URL:https://pare.is.tue.mpg.de/)
12. PyTorch [Электронный ресурс] – Режим доступа: [www/URL: https://en.wikipedia.org/wiki/PyTorch](http://www.URL:https://en.wikipedia.org/wiki/PyTorch)
13. Occlusion culling [Электронный ресурс] – Режим доступа: [www/URL: https://en.wikipedia.org/wiki/Hidden-surface\\_determination#Occlusion\\_culling](http://www.URL:https://en.wikipedia.org/wiki/Hidden-surface_determination#Occlusion_culling)
14. MediaPipe [Электронный ресурс] – Режим доступа: [www/URL: https://mediapipe.dev/](http://www.URL:https://mediapipe.dev/)
15. 3D Object Recognition and Pose with Relational Indexing – Режим доступа: [www/URL: https://homes.cs.washington.edu/~shapiro/mauro.pdf](http://www.URL:https://homes.cs.washington.edu/~shapiro/mauro.pdf)
16. Intelligent Identification System of the Process Liquid Solutions Composition. Bezsonov, O., Plyunin, O., Khusanov, A., Rudenko, O., Sotnikov, O., CEUR Workshop Proceedings, 2022, 3171, pp. 960–973 (Scopus)