

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

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

рівень вищої освіти перший (бакалаврський)

**РОЗРОБКА ВЕБЗАСТОСУНКУ ДЛЯ АНАЛІЗУ ВІДЕО ЗА**  
**ПОСИЛАННЯМ**  
(тема)

Виконав:  
студент 4 курсу, групи ІТІНФ-19-2

Глушков А.А.  
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник доц. Любченко В.А.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Кобилін О.А.  
(прізвище, ініціали)

2023 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)Кафедра Інформатики  
(повна назва)Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУстудентові Глушкову Артему Андрійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Розробка вебзастосунку для аналізу відео за посиланням

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 2 червня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека аналізу відео MediaPipe, мова програмування Python, середовище розробки VS Code.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз основних методів аналізу відео.2. Алгоритми аналізу відео за посиланням.3. Реалізація вебзастосунку для аналізу відео за посиланням.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми обробки фотореалістичних зображень, постановка задачі, сучасні методи обробки зображень, вибір методу реалізації, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-17.04.23	
3	Аналіз літератури з досліджуваної проблеми	18.04.23-20.04.23	
4	Аналіз технічних і програмних засобів	21.04.23-30.04.23	
5	Розробка методу	01.05.23-14.05.23	
6	Програмна реалізація	15.05.23-25.05.23	
7	Оформлення пояснювальної записки	26.05.23-30.05.23	
8	Перевірка на плагіат	01.06.23	
9	Рецензування	02.06.23	
10	Підготовка презентації та доповіді	03.06.23-04.06.23	
11	Занесення роботи в електронний архів	05.06.23	
12	Попередній захист кваліфікаційної роботи	11.06.23	

Дата видачі завдання 10 квітня 2023 р.

Студент \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_ доц. Любченко В.А.  
(підпис) (посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 63 с., 36 рис., 30 джерел.

СЕГМЕНТАЦІЯ ВІДЕО, ВІДСТЕЖЕННЯ ПОЗИ, ЗАСТОСУВАННЯ МАСКИ НА ОБЛИЧЧЯ, PYTHON FLASK, MEDIAPYPE.

Об'єктом роботи є реалізація застосунку для аналізу відео за посиланням.

Мета даної роботи полягає в розробці вебзастосунку, який дозволить користувачам проводити аналіз відеоконтенту за посиланням. Ця програма буде заснована на технологіях машинного навчання та комп'ютерного зору, які дозволять автоматично розпізнавати об'єкти на відео.

Використано методи аналізу відео, застосування сегментації на відео. Проведено порівняльний аналіз результатів аналізу відео, отриманих з використанням розроблених методів, з результатами, отриманими за допомогою існуючих алгоритмів та програм.

У результаті роботи здійснена програмна реалізація застосунку для аналізу відео за посиланням.

VIDEO SEGMENTATION, POSTURE TRACKING, FACE MASK, PYTHON FLASK, MEDIAPYPE.

The object of the work is the implementation of an application for video analysis by link.

The purpose of this work is to develop a web application that will allow users to analyze video content using a link. This program will be based on machine learning and computer vision technologies, which will allow automatic recognition of objects in the video.

Methods of video analysis, application of video segmentation are used. A comparative analysis of the video analysis results obtained using the developed methods with the results obtained using existing algorithms and programs was carried out.

As a result of the work, the software implementation of the application for video analysis by link was carried out.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	7
Вступ.....	8
1 Огляд основних методів для аналізу відео за посиланням .....	9
1.1 Поняття відео.....	9
1.1.1 Формати відео .....	10
1.1.2 Поняття при обробці відео.....	11
1.2 Перетворення відео.....	12
1.3 Фільтрація та корекція відео.....	14
1.4 Застосування аналізу відео .....	16
1.4.1 Відеоспостереження та безпека .....	16
1.4.2 Медицина та охорона здоров'я .....	17
1.4.3 Автомобільна промисловість .....	18
1.5 Постановка задачі .....	20
2 Алгоритми аналізу відео .....	21
2.1 Застосування фільтрів .....	21
2.2 Застосування аналізу відео .....	21
2.2.1 Виявлення та відстеження об'єктів.....	22
2.2.2 Розпізнавання жестів та рухів .....	25
2.2.3 Вилучення ознак та аналіз .....	27
2.2.4 Сегментація відео .....	30
2.2.5 Накладання маски на відео .....	32
2.3 Отримання ендпоінту за посиланням з ютубу.....	33
3 Реалізація аналізу відео за посиланням .....	35
3.1 Обґрунтування вибору середовища програмної реалізації .....	35
3.2 Програмна реалізація.....	44
3.3 Інструкція користувача .....	47
3.4 Тестування розробленої моделі.....	49

	6
3.4.1 Причини проведення тестування .....	50
3.4.2 Основні етапи тестування .....	51
3.4.3 Тестування застосунку .....	52
Висновки .....	60
Перелік джерел посилання .....	61

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ**

АТЗ – автономних транспортних засобів

NPM – Node Package Manager (менеджер пакетів вузлів)

## ВСТУП

Аналіз відео – це область інформаційних технологій, яка займається отриманням корисної інформації з відео-контенту. Вона включає різні технології, такі як комп'ютерний зір, машинне навчання, обробка зображень і відео, аналіз даних та інші. Аналіз відео може бути використаний для досягнення різних цілей, включаючи розпізнавання об'єктів та осіб на відео, визначення емоційного стану людей, забезпечення безпеки та інші.

Одним із основних застосувань аналізу відео є відеоспостереження. Системи відеоспостереження використовуються у різних сферах діяльності, включаючи безпеку, промисловість, медицину та транспорт. Системи відеоспостереження можуть забезпечити захист майна, забезпечення безпеки на робочому місці та у громадських місцях, а також допомогти у розслідуванні злочинів.

Також аналіз відео може бути використаний для забезпечення безпеки. Наприклад, за допомогою аналізу відео можна виявляти порушення дорожніх правил, визначати наявність зброї чи небезпечних предметів на відео, а також шукати злочинців за відеозаписами.

Розвиток технологій аналізу відео допомагає створювати нові можливості для використання відеоконтенту. Сучасні алгоритми машинного навчання та комп'ютерного зору дозволяють автоматично розпізнавати об'єкти на відео, визначати їх кількість та тип, аналізувати емоційний стан людей на відео, що розширює можливості аналізу та використання відео-контенту

Актуальність роботи полягає у устрої сучасного світу. У світі все більше і більше відео контенту публікується в Інтернеті. Згідно зі статистикою, на YouTube щодня завантажується понад мільйон годин відео, а на Facebook – понад 8 мільярдів відео проглядаються щодня. Разом з тим ці величезні обсяги відео контенту можуть бути використані не тільки в розважальних цілях, але і для вирішення бізнес-завдань, контролю безпеки та інших цілей.

# 1 ОГЛЯД ОСНОВНИХ МЕТОДІВ ДЛЯ АНАЛІЗУ ВІДЕО ЗА ПОСИЛАННЯМ

## 1.1 Поняття відео

Відео – це формат мультимедійного контенту, який є послідовністю кадрів (зображень), що програватимуться з певною швидкістю, створюючи ілюзію руху. Відео можна записувати за допомогою відеокамери або інших пристроїв, а також створювати за допомогою комп'ютерних програм та редакторів.

Відео можна подати у вигляді формули  $f$ , що описує його основні характеристики. Ось одна можлива математична вистава відео:

$$V(t, x, y) = F(t) * I(x, y), \quad (1.1)$$

де  $V(t, x, y)$  є значення відео в момент часу  $t$  і позиції  $(x, y)$ ;

$F(t)$  позначає функцію часу, яка визначає, які кадри повинні бути відображені в певний момент часу. Ця функція може бути задана дискретно чи безперервно залежно від контексту;

$I(x, y)$  є інтенсивністю пікселя в позиції  $(x, y)$  у кожному кадрі. Ця функція визначає яскравість, колір та інші характеристики пікселя.

Формула вказує, що значення відео в певний момент часу та позицію залежить від відповідного кадру (визначеного функцією часу) та інтенсивності пікселя у цій позиції.

Однак варто зазначити, що це спрощене уявлення, і насправді відео може мати складніші параметри, такі як колірний простір, альфа-канали для прозорості, а також інші атрибути, такі як аудіоінформація, стиснення.

### 1.1.1 Формати відео

Існує безліч форматів відео файлів, які визначають спосіб організації та зберігання відеоданих. Кожен формат має свої особливості, функції, що підтримуються, і ступінь стиснення. Ось деякі з найпоширеніших форматів відео.

AVI (Audio Video Interleave) – є одним із найстаріших і найбільш широко використовуваних форматів відео. Він підтримує різні кодеки та може включати відео та аудіо доріжки в одному файлі. Однак файли AVI можуть бути досить великими за розміром.

MP4 (MPEG-4 Part 14) – є одним з найбільш популярних форматів відео для Інтернету, потокового відтворення та мобільних пристроїв. Він забезпечує гарну якість відео при порівняно невеликому розмірі файлу та підтримує різні кодеки.

MOV (QuickTime) – розроблений компанією Apple і широко використовується на платформі Mac. Цей формат підтримує різні відео та аудіо кодеки та забезпечує високу якість відео. Файли MOV можуть бути відтворені на комп'ютерах та пристроях, які підтримують QuickTime.

WMV (Windows Media Video) – є форматом, розробленим Microsoft для використання на платформі Windows. Він забезпечує гарний ступінь стиснення відео та підтримує різні кодеки. WMV файли можуть бути відтворені на комп'ютерах, які використовують Windows Media Player, а також на пристроях, які підтримують формат Windows Media.

MKV (Matroska Video) – є контейнерним форматом, який може містити відео, аудіо та субтитри в одному файлі. Він дозволяє використовувати різні відео та аудіо кодеки, а також підтримує високу якість відео та багатомовні субтитри.

FLV (Flash Video) – є форматом, розробленим Adobe для відтворення відео в Adobe Flash Player. Він забезпечує гарне стиснення відео та є популярним для потокового відтворення відео в Інтернеті.

WebM – є відкритим форматом відео, розробленим для відтворення в Інтернеті. Він забезпечує гарний ступінь стиснення і підтримує високу якість відео за низької потреби в пропускній здатності.

### 1.1.2 Поняття при обробці відео

Обробка зображення – це процес перетворення та модифікації цифрового зображення з метою покращення його якості, аналізу його вмісту або створення нових візуальних ефектів. Вона широко застосовується у різних галузях, включаючи фотографію, медицину, робототехніку, комп'ютерний зір, графіку та дизайн.

Попередня обробка. Включає такі операції, як видалення шуму, поліпшення контрасту, різкості і яскравості, корекцію кольору і балансу білого. Метою попередньої обробки є підготовка зображення для подальшого аналізу чи поліпшення візуального сприйняття.

Фільтрування: Застосування фільтрів до зображення для видалення або придушення певних частот або характеристик, таких як шум, розмиття або деталі. Фільтри можуть бути лінійними (наприклад розмиття Гауса, медіанний фільтр) або нелінійними (наприклад, фільтр різкості, фільтр перетворення).

Сегментація. Процес поділу зображення на більш прості та зрозумілі частини, такі як об'єкти або регіони, що ґрунтуються на їх властивостях, таких як колір, текстура, контур або яскравість. Сегментація дозволяє виділити цікаві області зображення для подальшого аналізу чи маніпуляцій.

Вилучення ознак. Вилучення характеристик або особливостей зображення, які можуть бути використані для його класифікації, розпізнавання чи аналізу. Прикладами таких ознак можуть бути текстура, гістограма кольору, крайові дескриптори або кутові точки.

Реконструкція та реставрація. Процеси відновлення втраченої або пошкодженої інформації у зображенні. Це може включати відновлення

розмитості, видалення дефектів або відновлення деталей, які були втрачені під час захоплення або зберігання зображення.

Розпізнавання образів. Процес класифікації та розпізнавання об'єктів чи образів на основі їх характеристик чи ознак. Це може включати розпізнавання облич, визначення об'єктів на зображенні, розпізнавання тексту та інші види класифікації. На рисунку 1.1 можна побачити приклад роботи розпізнавання образів

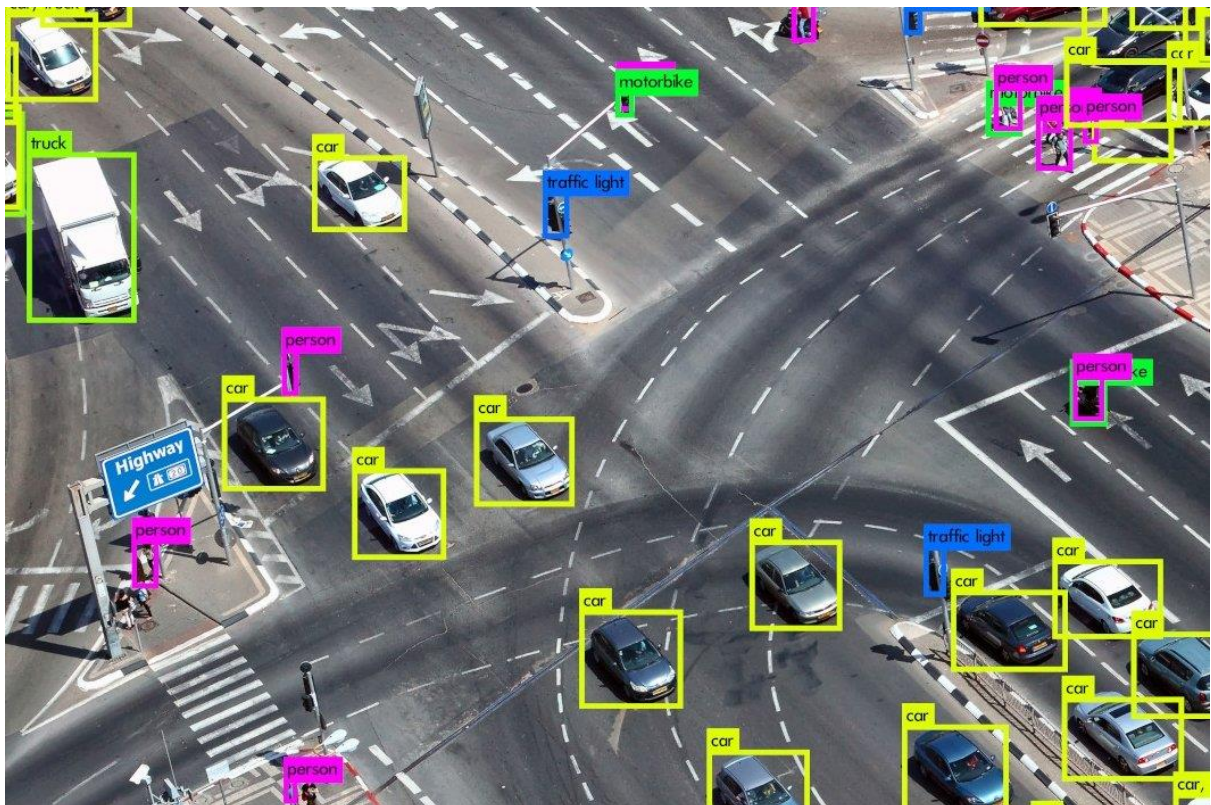


Рисунок 1.1 – Зображення розпізнавання образів [1]

## 1.2 Перетворення відео

Геометричні перетворення – дозволяють змінювати форму, розмір, орієнтацію та розташування відео. Вони можуть використовуватися для виправлення перспективних спотворень, повороту відео, зміни масштабу та обрізання кадрів.

В рамках геометричних перетворень відео можуть використовуватися такі формули:

а) зміна масштабу (Scaling):

$$x' = s_x * x, \quad (1.2)$$

$$y' = s_y * y, \quad (1.3)$$

де  $(x, y)$  – координати пікселя у вихідному відео;

$(x', y')$  – координати пікселя після зміни масштабу;

$s_x$  та  $s_y$  – коефіцієнти масштабування по осях  $x$  та  $y$  відповідно;

б) поворот (Rotation):

$$x' = x * \cos(\theta) - y * \sin(\theta), \quad (1.4)$$

$$y' = x * \sin(\theta) - y * \cos(\theta), \quad (1.5)$$

де  $(x, y)$  – координати пікселя у вихідному відео;

$(x', y')$  – координати пікселя після повороту;

$\theta$  – кут повороту;

в) перспективне спотворення (Perspective Distortion):

$$x' = \frac{H11 * x + H12 * y + H13}{H31 * x + H32 * y + H33}, \quad (1.6)$$

$$y' = \frac{H21 * x + H22 * y + H23}{H31 * x + H32 * y + H33}, \quad (1.7)$$

де  $(x, y)$  – координати пікселя у вихідному відео;

$(x', y')$  – координати пікселя після перспективного спотворення;

*H11-H33* – елементи матриці перетворення.

### 1.3 Фільтрація та корекція відео

Фільтрування та корекція відео – це процеси, спрямовані на покращення якості та зовнішнього вигляду відеоматеріалу. Вони включають застосування різних фільтрів та корекцій для усунення небажаних ефектів, підвищення різкості, корекції колірного балансу та контрасту, а також інших видів обробки.

Одним із основних фільтрів, що використовуються при фільтрації відео, є фільтр розмиття Blur Filter. Фільтр розмиття використовується для згладжування зображення, зменшення шумів та видалення дрібних деталей. Він працює шляхом усереднення значень пікселів навколо кожного пікселя, що створює ефект розмиття. Формула для фільтра розмиття може виглядати так:

$$I'(x, y) = \left(\frac{1}{N}\right) * \Sigma(I(x', y')), \quad (1.8)$$

де  $I'(x, y)$  – значення пікселя після застосування фільтра розмиття;

$I(x', y')$  – значення пікселів на околиці  $(x, y)$ ;

$N$  – кількість пікселів на околі.

Ще одним поширеним фільтром є фільтр підвищення різкості (Sharpening Filter). Він використовується для покращення чіткості зображення, роблячи його більш різким та деталізованим. Формула для фільтра підвищення різкості може бути представлена таким чином:

$$I'(x, y) = I(x, y) + k * \left(I(x, y) - \Sigma(I(x', y'))\right), \quad (1.9)$$

де  $I'(x, y)$  – значення пікселя після застосування фільтра підвищення різкості;

$I(x, y)$  – значення вихідного пікселя;

$I(x', y')$  – значення пікселів в околиці  $(x, y)$ ;

$k$  – коефіцієнт посилення різкості.

Фільтр зміни кольору (Color Filter) – це метод, що дозволяє змінити колірну гаму та насиченість відео. З його допомогою можна створювати ефекти, змінювати загальну палітру кольорів або підкреслювати певні відтінки.

Фільтр RGB to BGR (Red-Green-Blue to Blue-Green-Red) використовується для зміни порядку кольорових компонентів у зображенні або відео. Він змінює колірний простір з RGB на BGR, де синій (B) стає першим компонентом, зелений (G) – другим, а червоний (R) – третім.

Формула для застосування фільтра RGB to BGR для кожного пікселя відео може бути наступною:

$$B' = R ; G' = G ; R' = B, \quad (1.10)$$

де  $R', G', B'$  – нові значення колірних компонентів пікселя після застосування фільтра. Значення червоного  $R$  компонента замінюється на значення синього  $B$ , значення зеленого  $G$  залишається без змін, а значення синього  $B$  замінюється на значення червоного  $R$ .

Застосування фільтра RGB to BGR може бути корисним, наприклад, під час роботи з певними бібліотеками або інструментами, які потребують зображень або відео в колірному просторі BGR.

Фільтрування відео можна використовувати для усунення шуму, викликаного різними факторами, такими як низьке освітлення або стиснення відео. Чисте зображення дозволяє отримати точніші результати під час аналізу відео, особливо при розпізнаванні об'єктів або детекції руху.

Зміна контрасту та яскравості відео може допомогти виділити важливі деталі або об'єкти на відео. Це може бути корисним при аналізі

відеоспостереження, де необхідно виявити підозрілу поведінку або ідентифікувати об'єкти в освітлених областях.

Зміна кольору відео може допомогти розпізнавати та класифікувати об'єкти на відео. Наприклад, завдання розпізнавання трафікових знаків, фільтрація та зміна кольору можуть допомогти виділити знаки на дорозі, зробити їх яскравішими і відмінними від навколишнього фону.

Фільтрування та зміна кольору можуть використовуватись для сегментації та виділення об'єктів на відео. Шляхом зміни кольору заданого об'єкта або заданої області можна виділити його від навколишнього фону, що корисно, наприклад, завдання відстеження об'єктів або аналізі руху.

#### 1.4 Застосування аналізу відео

Аналіз відео є потужним інструментом із різними застосуваннями. Він включає вилучення інформації з відео, виявлення об'єктів і подій, аналіз руху і дій, класифікацію і трекінг об'єктів, а також інші обчислювальні завдання.

##### 1.4.1 Відеоспостереження та безпека

Аналіз відео використовується для виявлення та розпізнавання об'єктів, виявлення руху, відстеження об'єктів або людей на відеозаписах. Це може застосовуватись у системах відеоспостереження, громадської безпеки, контролю доступу тощо. Аналітичні алгоритми дозволяють автоматично визначати підозрілу поведінку, виявляти інциденти та покращувати загальну безпеку.

Виявлення руху. Використовується для виявлення руху об'єктів на відео. Це дозволяє системі автоматично визначати, коли відбувається рух усередині області спостереження та вживати відповідних заходів, наприклад, надсилати

оповіщення про порушення безпеки або записувати події для подальшого аналізу.

Розпізнавання облич. Використовується для розпізнавання облич людей на відео. Це дозволяє системі відеоспостереження автоматично ідентифікувати людей, аналізувати їхню поведінку, визначати відвідувачів чи потенційно небезпечні особи.

Аналіз поведінки. Використовується для аналізу поведінки людей на відеозаписах. Наприклад, система відеоспостереження може виявити аномальну поведінку, таку як залишені об'єкти, неправильний рух, бійки та інші потенційно небезпечні ситуації.

Ліцензійний розпізнавач. Використовується для розпізнавання номерних знаків на транспортних засобах. Це дозволяє системі відеоспостереження автоматично реєструвати автомобілі, відстежувати їх рух та аналізувати дані для різних цілей, таких як контроль паркування або пошук викрадених автомобілів.

Аналіз черги та потоків людей. Використовується для аналізу потоків людей та управління чергами у місцях публічного доступу, таких як вокзали, аеропорти, торгові центри та інші місця з великим потоком відвідувачів. Це дозволяє системі визначити пікові навантаження, попереджати про переповнення та ефективно керувати потоком людей.

#### 1.4.2 Медицина та охорона здоров'я

В аналізі відео використовуються методи комп'ютерного зору та машинного навчання для виявлення та класифікації медичних станів, аналізу рухів пацієнтів, стеження за життєвими показниками тощо. Це може застосовуватися в галузі діагностики, реабілітації, моніторингу пацієнтів та телемедицини.

Аналіз рухів та реабілітація. Використовуються для оцінки рухів пацієнтів, аналізу біомеханіки та реабілітації. За допомогою алгоритмів комп'ютерного зору та трекінгу рухів можна автоматично вимірювати та аналізувати параметри рухів, такі як діапазон руху, швидкість, симетрія та координація. Це допомагає оцінити ефективність реабілітаційних програм, відстежувати прогрес пацієнтів та розробляти персоналізовані терапевтичні плани.

Діагностика та моніторинг. Використовуються для діагностики та моніторингу різних медичних станів. Наприклад, аналіз відео може допомогти у виявленні симптомів неврологічних розладів, таких як паркінсонізм або епілепсія шляхом аналізу рухів і особливостей поведінки пацієнтів. Також аналіз відео може бути використаний для моніторингу пацієнтів із хронічними захворюваннями, щоб виявити зміни у їхньому стані та запобігти можливим ускладненням.

Аналіз відео має значне значення у телемедицині, де пацієнти можуть спілкуватися з лікарями чи фахівцями через відеозв'язок. Аналіз відео може допомогти лікарям у дистанційній діагностиці, оцінці стану пацієнта та наданні медичної допомоги на основі візуальних даних.

#### 1.4.3 Автомобільна промисловість

В аналізі відео застосовуються для виявлення дорожніх знаків, розпізнавання об'єктів на дорозі, визначення смуг руху, попередження про зіткнення та інші функції, пов'язані з автономними транспортними засобами та безпекою на дорозі.

Детекція та розпізнавання об'єктів: Аналіз відео може бути використаний для детекції та розпізнавання об'єктів на дорозі, таких як автомобілі, пішоходи, велосипедисти та дорожні знаки. Це дозволяє автомобілю або системі АТЗ виявляти та відстежувати навколишні об'єкти,

приймати рішення на основі їх руху та поведінки, та забезпечувати безпечне водіння.

Автоматичне емерджентне гальмування та попередження про зіткнення: Аналіз відео може бути використаний для виявлення потенційних зіткнень та попередження водія чи системи АТЗ про можливу небезпеку. Система може аналізувати відеопотоки, визначати швидкість і відстань до інших об'єктів, автоматично активувати гальма або попередити водія про небезпеку.

Розпізнавання дорожньої розмітки та навігація: Аналіз відео може допомогти у розпізнаванні дорожньої розмітки, такої як лінії розмітки, знаки та світлофори. Це допомагає в системах автоматичної навігації та автоматичного керування автомобілями, забезпечуючи точне розпізнавання та розуміння дорожніх умов.

Детекція втоми та невідповідності: Аналіз відео може бути використаний для визначення втоми та невідповідності водія. Система може аналізувати вираз обличчя, очі, міміку та інші ознаки втоми, а також відстежувати дії водія, щоб виявити відволікання чи незвичайну поведінку. Це допомагає у попередженні можливих аварійних ситуацій та забезпеченні безпеки на дорозі.

Навчання та аналіз поведінки: Аналіз відео може використовуватися для навчання систем АТС та аналізу поведінки на дорозі. Записи відео з автомобілів або АТС можуть бути використані для створення навчальних наборів даних, навчання моделей машинного навчання та покращення алгоритмів автоматичного керування. Це допомагає у розробці більш ефективних та безпечних систем автоматичного керування транспортними засобами.

## 1.5 Постановка задачі

Методи та алгоритми аналізу відео за посиланням є актуальними на сьогоднішній день, бо такі технології знаходять все більше застосувань в різних сферах, включаючи безпеку, медицину, рекламу, розваги та багато інших.

Методи аналізу відео за посиланням охоплюють широкий спектр задач, включаючи розпізнавання облич, визначення об'єктів, виявлення руху, визначення емоцій, визначення активності та багато іншого. Основні алгоритми, використовувані для аналізу відео, включають такі техніки як візуальне виявлення та відслідковування об'єктів, аналіз руху, класифікацію зображень та глибинне навчання.

Наприклад, в сфері безпеки відеоаналітика використовується для виявлення підозрілих дій, відслідковування руху осіб та об'єктів, виявлення вторгнень та виявлення небезпечних ситуацій. В медицині аналіз відео може використовуватись для виявлення патологій на зображеннях, моніторингу пацієнтів та аналізу руху для діагностики та реабілітації.

Об'єктом роботи є реалізація застосунку для аналізу відео за посиланням.

Мета даної роботи полягає в розробці вебзастосунку, який дозволить користувачам проводити аналіз відеоконтенту за посиланням. Ця програма буде заснована на технологіях машинного навчання та комп'ютерного зору, які дозволять автоматично розпізнавати об'єкти на відео.

Головні завдання проекту включають:

- а) аналіз та вибір сучасних методів та алгоритмів для аналізу відео;
- б) створення інтуїтивно зрозумілого та зручного інтерфейсу користувача, що дозволить легко використовувати застосунок та досягати бажаних результатів;
- в) розробка функцій, які дозволять користувачам здійснювати аналіз відео.

## 2 АЛГОРИТМИ АНАЛІЗУ ВІДЕО

### 2.1 Застосування фільтрів

Переведення зображень у колірну модель BGR для аналізу відео має особливе значення в контексті певних бібліотек та алгоритмів комп'ютерного зору, таких як OpenCV.

У деяких бібліотеках та алгоритмах комп'ютерного зору колірна модель BGR використовується за замовчуванням. Наприклад, у бібліотеці OpenCV, яка широко застосовується для аналізу відео, зображення часто подаються у колірній моделі BGR.

Переведення зображення з RGB до BGR може бути необхідним, якщо ви хочете використовувати алгоритми, призначені для роботи із зображеннями у колірній моделі BGR. Деякі алгоритми або моделі машинного навчання можуть вимагати саме цієї моделі для правильної роботи.

### 2.2 Застосування аналізу відео

MediaPipe пропонує потужні інструменти для аналізу відео в режимі реального часу. З його допомогою ви можете обробляти відеопотоки, отримувати різні характеристики та виконувати різні завдання аналізу.

Однією з переваг MediaPipe є його гнучкість та легкість інтеграції з іншими інструментами та фреймворками розробки. Він підтримує різні мови програмування, такі як C++, Python та Java, і може бути використаний на різних платформах, включаючи настільні комп'ютери, мобільні пристрої та вбудовані системи.

### 2.2.1 Виявлення та відстеження об'єктів

Mediapipe надає модулі для виявлення та відстеження об'єктів у відео. Це може бути корисно для завдань, таких як автоматичне лічильник об'єктів, стеження за об'єктами, що рухаються, або детектування певних сцен.

Детекція об'єкта. Спочатку на кожному кадрі відеопослідовності виконується детекція об'єкта. Для цього використовуються попередньо навчені моделі глибокого навчання, такі як Single Shot MultiBox Detector (SSD) або Faster R-CNN, які здатні виявляти та обмежувати прямокутну область, де знаходиться об'єкт. Для кожного виявленого об'єкта виходить прямокутна рамка з координатами положення на кадрі.

Відстеження об'єкта. Після детекції об'єкта використовується алгоритм відстеження, який дозволяє відстежувати об'єкт у наступних кадрах відео. Одним із популярних алгоритмів відстеження є Kalman Filter або його модифікації. Він використовує модель руху об'єкта та оновлює його стан на кожному кадрі, враховуючи інформацію вимірювань та передбачення руху.

Тривимірна реконструкція. Після виявлення та відстеження об'єкта на кожному кадрі Mediapipe Objectron виконує тривимірну реконструкцію об'єкта. Це здійснюється за допомогою геометричних методів, використовуючи інформацію про положення та розміри об'єкта в кожному кадрі відео. В результаті виходить тривимірна модель об'єкта, яка може бути використана для подальшого аналізу чи візуалізації. Візуалізація показується на рисунках 2.1-2.4 на цих зображеннях показані одні з об'єктів, які можуть виявити Objectron.

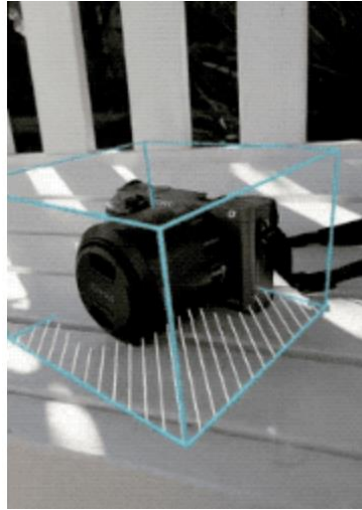


Рисунок 2.1 – Зображення роботи Медіаріре Об'єктрон [2]



Рисунок 2.2 – Зображення роботи Медіаріре Об'єктрон [2]



Рисунок 2.3 – Зображення роботи Медіаріре Об'єктрон [2]

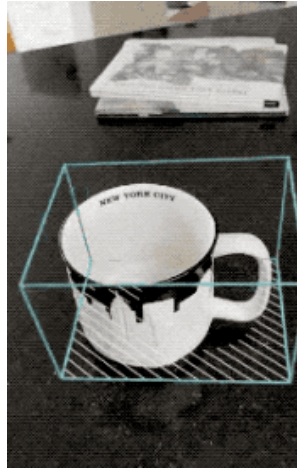


Рисунок 2.4 – Зображення роботи Mediapipe Objectron [2]

Двоступінчастий трубопровід працює в 3 рази швидше, ніж одноступінчастий з аналогічною чи найкращою точністю.

Перший етап використовує детектор об'єктів для пошуку 2D-обрізання об'єкта. Другий етап бере кадрівання зображення та оцінює 3D-обмежувальну рамку. У той же час це також обчислює 2D-кадрівання об'єкта для наступного кадру, тому об'єкт Детектору не потрібно запускати кожен кадр.

На рисунку 2.5 зображена візуалізація двоступінчастого трубопроводу.

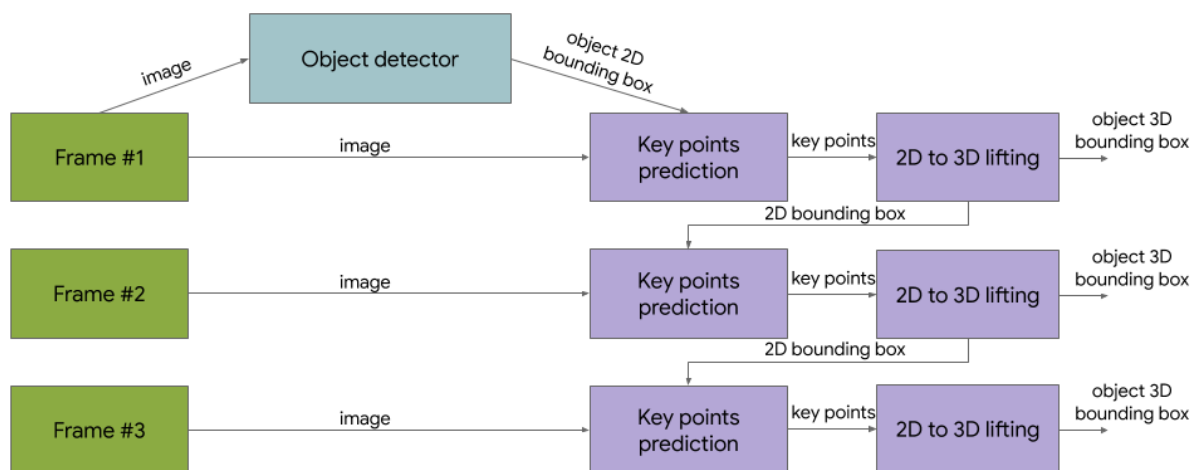


Рисунок 2.5 – Зображення мережевої архітектури та постобробки для двоетапного виявлення 3D-об'єктів [2]

### 2.2.2 Розпізнавання жестів та рухів

Mediarpipe надає модуль Hands, який спеціалізується на розпізнаванні жестів і рухів рук у відеопослідовності. Цей модуль використовує глибоку нейронну мережу для виявлення та відстеження ключових точок на руках, а потім застосовує алгоритми для розпізнавання жестів та рухів.

Детекція рук. На кожному кадрі відеопослідовності виконується детекція рук. Для цього застосовується попередньо навчена нейронна мережа, яка здатна виявляти та обмежувати прямокутні області, де знаходяться руки. Для кожної виявленої руки виходить прямокутна рамка з координатами положення на кадрі.

Відстеження ключових точок. Після детекції рук застосовується алгоритм відстеження ключових точок, що дозволяє відслідковувати положення та рух ключових точок на кожній руці у наступних кадрах відео. Для цього використовуються методи, що базуються на оптичному потоці або глибинних даних, щоб оновлювати координати ключових точок на кожному кадрі.

Розпізнавання жестів та рухів. Після відстеження ключових точок на руках застосовуються алгоритми для розпізнавання жестів та рухів. Це може бути реалізовано з використанням методів класифікації або зіставлення жестів із попередньо визначеними шаблонами. Нейронні мережі також можуть використовуватися для навчання моделі розпізнавання жестів на великому наборі даних. На рисунку 2.6 зображено як будується орієнтири на руці за допомогою яких можна класифікувати жести та на рисунках 2.6-2.8 показано як працює на справжній руці.

Інтерпретація результатів. Результати розпізнавання жестів та рухів можуть бути інтерпретовані для певних дій чи команд. Наприклад, певний жест може бути пов'язаний з виконанням певної дії, як сигнал управління системою або взаємодії з додатком.

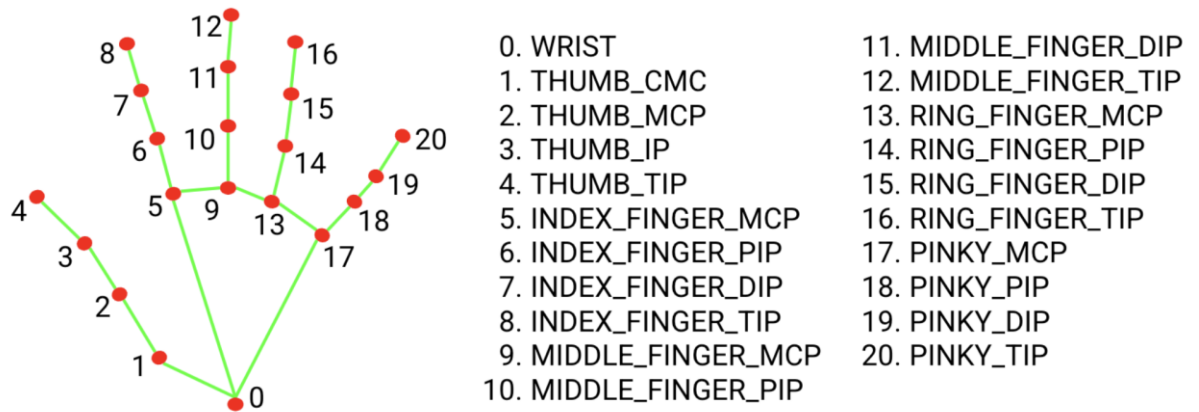


Рисунок 2.6 – Зображення ручних орієнтирів [3]



Рисунок 2.7 – Зображення роботи Hand [3]

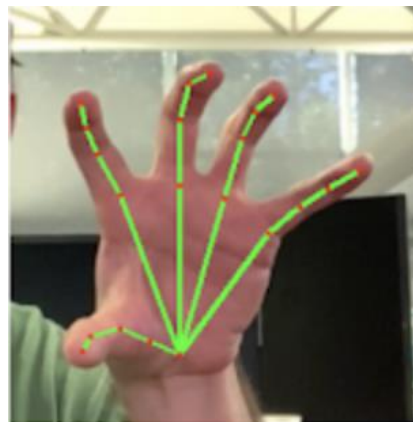


Рисунок 2.8 – Зображення роботи Hand [3]

### 2.2.3 Вилучення ознак та аналіз

Mediarpipe надає різні модулі та інструменти для отримання ознак та аналізу даних у відео. Один з таких модулів – це модуль «Pose», який спеціалізується на добуванні ознак пози людини та аналізі рухів.

Детекція людського тіла. На кожному кадрі відеопослідовності виконується детекція людського тіла. Для цього використовується заздалегідь навчена нейронна мережа, яка здатна виявляти області, де знаходиться людське тіло. Для кожної виявленої людини виходить прямокутна рамка з координатами її положення на кадрі.

Процес детекції людського тіла включає такі кроки:

а) підготовка вхідних даних: Вихідне відео розбивається на окремі кадри, і кожен кадр піддається передобробці для підготовки до подальшої обробки. Це може містити зміну розміру кадру, нормалізацію значень пікселів або застосування інших технік попередньої обробки;

б) завантаження моделі детекції: У Mediarpipe доступна попередньо навчена модель детекції людського тіла. Вона може бути завантажена з бібліотеки Mediarpipe або імпортована із зовнішніх джерел;

в) застосування моделі детекції: Кожен кадр подається на вхід моделі детекції, яка аналізує і визначає області, що містять людське тіло. Це може бути реалізовано з використанням методів, заснованих на згорткових нейронних мережах, таких як SSD (Single Shot MultiBox Detector) або інших подібних архітектур;

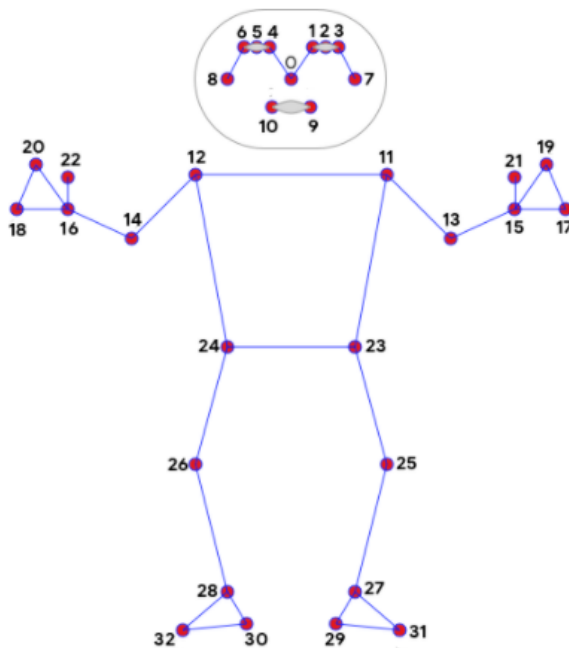
г) обробка результатів детекції Після застосування моделі детекції виходять координати прямокутних рамок, які описують положення виявлених людських тіл на кадрі. Ці рамки можуть бути використані для відображення чи подальшого аналізу;

г) повторення для наступних кадрів: Щоб забезпечити безперервне відстеження людських тіл протягом усього відео, алгоритм детекції застосовується для кожного кадру послідовно.

Відстеження ключових точок. Після детекції людського тіла застосовується алгоритм відстеження ключових точок, що дозволяє відслідковувати положення та рух ключових точок на тілі у наступних кадрах відео. Для цього використовуються методи, що базуються на оптичному потоці або глибинних даних, щоб оновлювати координати ключових точок на кожному кадрі.

Витяг ознак пози. Після відстеження ключових точок застосовуються алгоритми для отримання ознак пози людини. Це може включати обчислення кутів між суглобами, довжину та відносне положення кінцівок, а також інші характеристики, що описують позу тіла.

Аналіз та інтерпретація пози. Вилучені ознаки пози можуть бути аналізовані для визначення певних поз, рухів чи дій. Наприклад, певна поза або рух може бути пов'язана з виконанням певної дії або використовуватись для керування системою або взаємодії з програмою. На рисунку 2.9 зображено як будується орієнтири на тілі, а на рисунках 2.10 та 2.11 зображено приклад роботи.



- |                    |                      |
|--------------------|----------------------|
| 0. nose            | 17. left_pinky       |
| 1. left_eye_inner  | 18. right_pinky      |
| 2. left_eye        | 19. left_index       |
| 3. left_eye_outer  | 20. right_index      |
| 4. right_eye_inner | 21. left_thumb       |
| 5. right_eye       | 22. right_thumb      |
| 6. right_eye_outer | 23. left_hip         |
| 7. left_ear        | 24. right_hip        |
| 8. right_ear       | 25. left_knee        |
| 9. mouth_left      | 26. right_knee       |
| 10. mouth_right    | 27. left_ankle       |
| 11. left_shoulder  | 28. right_ankle      |
| 12. right_shoulder | 29. left_heel        |
| 13. left_elbow     | 30. right_heel       |
| 14. right_elbow    | 31. left_foot_index  |
| 15. left_wrist     | 32. right_foot_index |
| 16. right_wrist    |                      |

Рисунок 2.9 – Зображення орієнтирів пози [4]

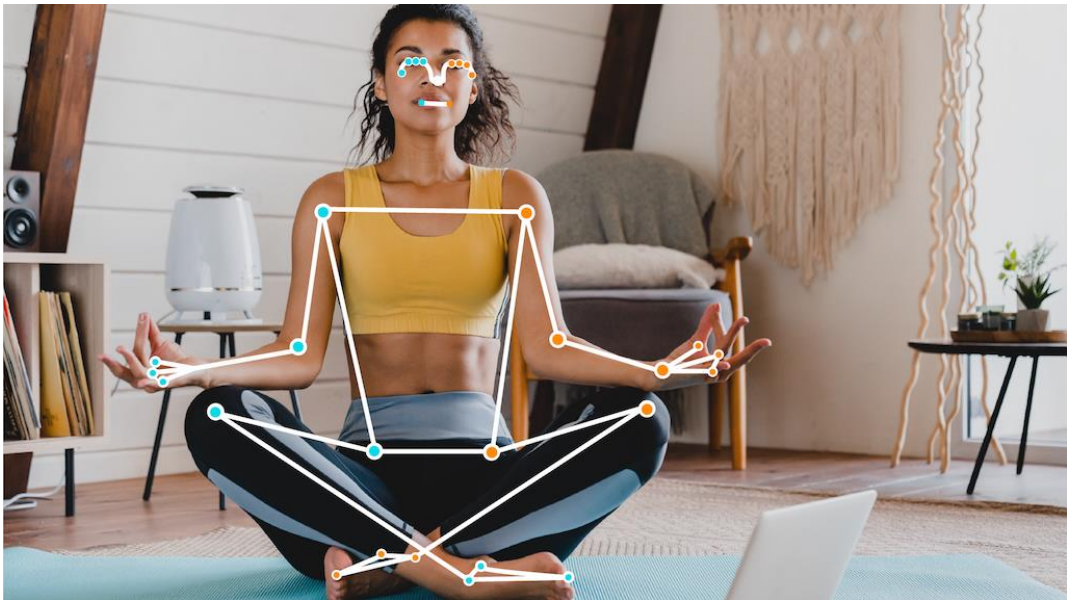


Рисунок 2.10 – Зображення прикладу роботи Pose [5]



Рисунок 2.11 – Зображення прикладу роботи Pose [6]

Модель виявлення людини/пози (детектор BlazePose). Детектор заснований на моделі BlazeFace, яка використовується для виявлення осіб у

MediaPipe. Цю модель адаптували для виявлення людського тіла, використовуючи її як проксі. Детектор явно передбачає дві додаткові віртуальні ключові точки, які повністю описують центр тіла людини, її обертання та масштаб у вигляді кола. Натхнення було взято з ідеєю вітрувіанської людини Леонардо да Вінчі зображено на рисунку 2.12 і передбачаємо також середину стегон людини, радіус кола, що охоплює всю людину, і кут нахилу лінії, що з'єднує плече та середню точку кульшового суглоба.

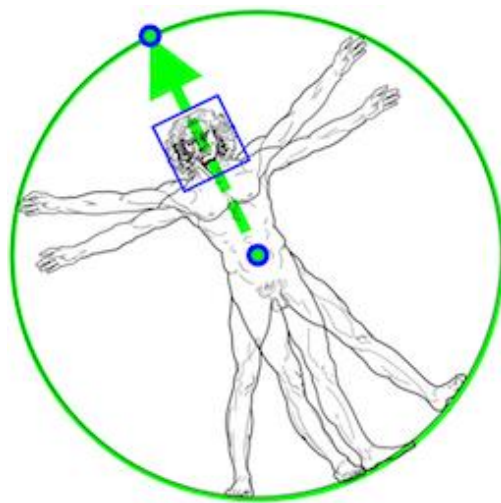


Рисунок 2.12 – Зображення вітрувіанської людини яка вирівнялася за допомогою двох віртуальних ключових точок, передбачених детектором BlazePose, на застосунок до обмежувальної рамки обличчя [4]

#### 2.2.4 Сегментація відео

MediaPipe надає набір інструментів для сегментації відео, включаючи алгоритми та моделі, які можуть бути використані для виділення та аналізу об'єктів на відео.

Один з підходів, що використовуються в MediaPipe для сегментації відео, заснований на використанні нейронних мереж, таких як модель DeepLab, яка широко використовується в області сегментації зображень та

відео. DeepLab використовує згорткові нейронні мережі для виділення об'єктів на відео та створення масок сегментації, які вказують на належність кожного пікселя до певного об'єкта або класу об'єктів.

Процес сегментації відео за допомогою MediaPipe включає такі кроки:

- а) завантаження попередньо навченої моделі сегментації, наприклад моделі DeepLab;
- б) розбиття відео на окремі кадри;
- в) застосування моделі сегментації для кожного кадру відео для створення маски сегментації;
- г) поєднання масок сегментації з різних кадрів, щоб отримати сегментоване відео;
- г) візуалізація результатів сегментації, наприклад, відображення сегментованих об'єктів на оригінальному відео.

MediaPipe забезпечує зручний інтерфейс для завантаження моделей сегментації, їх застосування до відео та обробки результатів. Це дозволяє легко інтегрувати сегментацію відео в різні програми, такі як системи відеоспостереження, редактори відео та візуальні ефекти. Приклад роботи сегментації у MediaPipe зображен на рисунку 2.13.

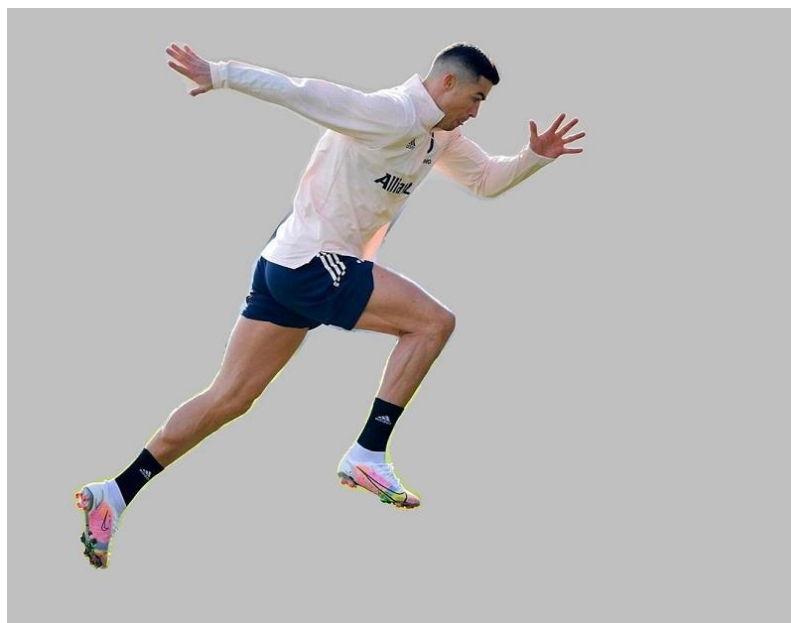


Рисунок 2.13 – Зображення роботи сегментації [6]

Відео можна сегментувати на основі різних властивостей, таких як колір, текстура, рух і глибина. Комбінування кількох ознак може покращити точність та якість сегментації.

У сегментації відео можна використовувати як семантичні методи, створені задля розпізнавання і класифікацію об'єктів, і інстансні методи, які відстежують окремі екземпляри об'єктів.

### 2.2.5 Накладання маски на відео

Для накладання маски на відео за допомогою MediaPipe необхідно виконати такі кроки:

а) завантажити відео: Спочатку потрібно завантажити відео, на яке ви хочете накласти маску. Ви можете використовувати функції OpenCV або інші інструменти для читання відео;

б) ініціалізація MediaPipe: Імпортуйте необхідні бібліотеки MediaPipe та створіть екземпляр відповідної моделі, яка використовується для сегментації. Наприклад, ви можете використовувати модель DeepLab для сегментації відео;

в) обробка кадрів відео: Прочитайте кожен кадр відео та передайте його у модель сегментації MediaPipe для отримання маски сегментації. Це можна досягти за допомогою функцій MediaPipe для обробки кадрів зображень;

г) накладення маски на відео: Після отримання маски сегментації ви можете застосувати її до кожного кадру відео. Накладіть маску на оригінальний кадр відео, використовуючи операції піксельного злиття або маскування, доступні у вибраній бібліотеці для обробки зображень або відео;

г) записування відео з маскою: Запишіть змінені кадри з маскою в новий відеофайл. Це можна зробити за допомогою функцій запису відео у бібліотеці для роботи з відео.

Роботу маски можна спостерігати на малюнку 2.14.

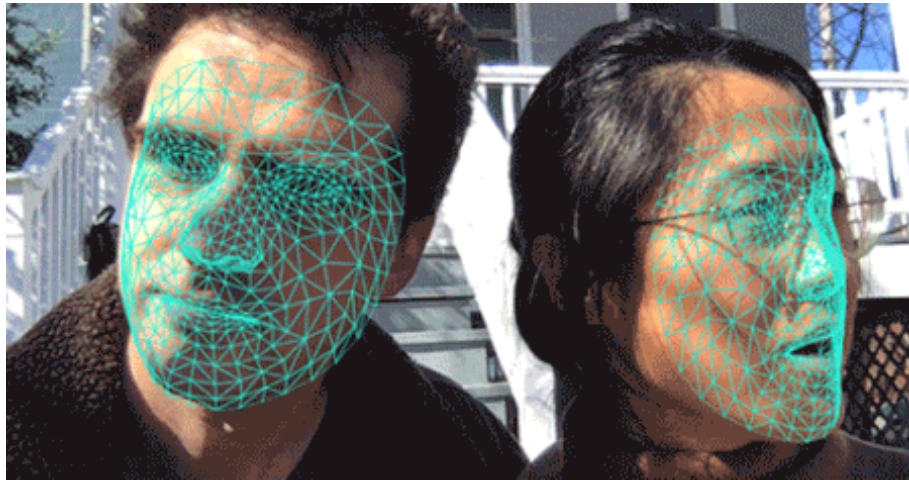


Рисунок 2.14 – Зображення роботи маски [7]

### 2.3 Отримання ендпоінту за посиланням з ютубу

YouTube-dl – це інструмент командного рядка для завантаження відео та аудіофайлів із різних відеохостингів, включаючи YouTube. Для отримання посилання на відео YouTube-dl використовує методи, що ґрунтуються на аналізі вебсторінки, що містить відео. Ось основні кроки, які YouTube-dl виконує для отримання посилання на відео:

а) пошук сторінки відео: YouTube-dl приймає URL відео або його ідентифікатор і завантажує відповідну вебсторінку за допомогою запиту HTTP;

б) аналіз сторінки: YouTube-dl аналізує вміст вебсторінки, щоб знайти інформацію про відео, включаючи його заголовок, ідентифікатор, формати та посилання;

в) вилучення інформації про формати: YouTube-dl витягує інформацію про різні формати відео, доступні для завантаження. Ця інформація включає роздільну здатність, кодек, бітрейт та інші характеристики;

- г) вибір оптимального формату: YouTube-dl вибирає найкращий формат відео, враховуючи переваги користувача, доступність та інші фактори;
- г) отримання прямого посилання: Після вибору формату YouTube-dl виконує додаткові запити до API відеохостингу або інших ресурсів, щоб отримати пряме посилання на відеофайл;
- д) завантаження відео: Отримане пряме посилання використовується для завантаження відео за допомогою YouTube-dl або інших інструментів завантаження.

YouTube-DL також відомий своєю гнучкістю та можливістю завантаження відео з незвичайних та несподіваних джерел. Наприклад, YouTube-DL може бути використаний для завантаження відео з популярних сайтів, таких як TED Talks, BBC iPlayer, Twitch, Reddit і навіть вебсайтів організацій новин. Завдяки цій універсальності YouTube-DL став одним із найпопулярніших інструментів для завантаження відео з різних онлайн-ресурсів. Приклад роботи зображен на рисунках 2.15- 2.17.

```
PS C:\Users\glysh> yt-dlp -f best -g "https://youtu.be/W4ZGUgugwHw"
```

Рисунок 2.15 – Зображення роботи YouTube-dl. Приклад команди

```
WARNING: "-f best" selects the best pre-merged format which is often not the best option.
To let yt-dlp download and merge the best available formats, simply do not pass any format selection.
If you know what you are doing and want only the best pre-merged format, use "-f b" instead to suppress this warning
```

Рисунок 2.16 – Зображення роботи YouTube-dl. Попередження

```
https://rr1---sn-punu5gjhv83g-ig3e.googlevideo.com/videoplayback?expire=1684590316&ei=jHpoZP2LFpnIyAXY0IHQ8A&ip=46.172.84.206&id=o-ACVUVp5-lHEABvjw
8FurKdFX7sz4IIYeej6xh-vKcjn6itag=22&source=youtube&request=yes&mh=kJ6mm=31%2C29&mn=sn-punu5gjhv83g-ig3e%2Csn-3c27sn7z6ms=au%2Crdumv=m&mv=1&pl
=24&initcwndbps=14759006&spc=qEK7B80D-BW2R14SF5N2W6yA6pJ5Qw6vprv=1&svpuc=1&time=video%2Fmp4&cnr=14&ratebypass=yes&dur=2061.397&limit=168021869392142&
nt=1684568218&fvip=1&fexp=24007246%2C24363393%2C51000014&c=ANDROID&txp=6318224&params=expire%2Cei%2Cip%2Cid%2Citag%2Csource%2Crequ%2Cspc%2Cv
prv%2Csvpuc%2Ctime%2Cmn%2Cratebypass%2Cdur%2Cint%2Csig=A008036mRQIgcLmkvYIeNkLnsFkz01Y_17FG5ince0TzabHproccIQCFe45P91n-mq01VmjDL_f6xjBl20b0Jke
YEmktqCQ%3D%30&lspans=m%2Cm%2Ccn%2Ccs%2Cav%2Cmv%2Cpl%2Cinitcwndbps&lsig=AG3c_xAWRQIHAPPyzVWp_6B9AhdaxE5uNYHbdHoEwuGmVWdb19sP00A18G-pxeb8Vv5l
V1U2k_qie3oGnhKanXWBKfkB0DSkrk%3D%3D
```

Рисунок 2.17 – Зображення роботи YouTube-dl. Посилання

### 3 РЕАЛІЗАЦІЯ АНАЛІЗУ ВІДЕО ЗА ПОСИЛАННЯМ

#### 3.1 Обґрунтування вибору середовища програмної реалізації

В рамках кваліфікаційної роботи було створено вебзастосунок, призначений для аналізу відео за посиланням. Цей додаток використовує різні алгоритми фільтрації, сегментації, геометричних перетворень та виявлення об'єктів при обробці відео [8-16]. Для реалізації було обране середовище Visual Studio Code – це популярне інтегроване середовище розробки інтегроване середовище розробки, яке надає зручність та широкі можливості для програмування на різних мовах, включаючи Python. Вибір Visual Studio Code для реалізації вашого вебзастосунку на Python може мати кілька обґрунтувань:

а) кросплатформенність. Visual Studio Code підтримує різні операційні системи, такі як Windows, macOS і Linux, що дозволяє розробляти та тестувати ваш додаток на різних платформах;

б) розширення для Python. Visual Studio Code має багато розширень, спеціально розроблених для роботи з Python, які надають додаткові функціональні можливості, такі як підказки коду, перевірка синтаксису, відладка та автодоповнення коду;

в) інтегровані інструменти. Visual Studio Code має багато корисних інструментів, таких як вбудована консоль, керування версіями Git, відладчик та підтримка віртуальних середовищ (наприклад, через використання віртуальних середовищ Python, таких як `virtualenv` або `Conda`);

г) зручність в роботі. Visual Studio Code має інтуїтивно зрозумілий інтерфейс, швидкість роботи та можливості налаштування, що полегшує розробку і забезпечує зручне середовище для програмування.

Для розробки вебзастосунка було обрано мову програмування Python. Це одна з найпопулярніших мов програмування для аналізу даних, включаючи

аналіз відео. Існує кілька причин, чому Python широко використовується для аналізу відео:

а) велика кількість бібліотек: Python має широкий спектр бібліотек для обробки даних, включаючи відео. Наприклад, Open Source Computer Vision Library – потужна бібліотека, яка надає функціональність для обробки та аналізу відео. Існує також багато інших спеціалізованих бібліотек для розпізнавання об'єктів, виявлення руху, відстеження об'єктів тощо [17-24];

б) простота використання: Python має зрозумілий синтаксис і легко вивчається, що полегшує розробку і реалізацію алгоритмів аналізу відео. Python також має багато документації і прикладів коду, що сприяє швидкому розпочатку роботи;

в) спільнота розробників: Python має велику та активну спільноту розробників, яка постійно розширює функціональність мови та розробляє нові бібліотеки для аналізу відео. Це означає, що зазвичай можна знайти готові рішення для багатьох завдань аналізу відео і отримати підтримку спільноти при потребі;

г) інтеграція з іншими мовами: Python добре поєднується з іншими мовами програмування. Це дозволяє використовувати швидкі обчислення, написані на мовах, таких як C++ або CUDA, для вимогливих завдань обробки відео, а використовувати Python для керування та інтеграції різних етапів аналізу;

г) масштабованість: Python добре підходить для розробки прототипів і експериментів з аналізом відео. Крім того, існують інструменти, такі як NumPy і SciPy, які дозволяють ефективно працювати з великими обсягами даних і виконувати обчислювально важкі завдання [25-30].

Загалом, Python є гнучким і потужним інструментом для аналізу відео, який має багато ресурсів, бібліотек і спільноту розробників, що сприяє розробці високоякісних програм для обробки і аналізу відео.

Для розробки вебзастосунку я обрав Flask – це легкий та простий у використанні фреймворк для створення бекенду вебзастосунку. Його основні переваги:

а) простота використання. Flask має простий і зрозумілий синтаксис, що дозволяє швидко розпочати роботу з розробкою бекенду. Він не нав'язує складних або обов'язкових компонентів, дозволяючи вам гнучко структурувати свій код;

б) масштабованість. Flask дозволяє легко масштабувати ваш бекенд відповідно до зростаючих потреб вашого проєкту. Ви можете використовувати розширення та бібліотеки Flask, щоб додати необхідну функціональність і взаємодію з іншими інструментами;

в) підтримка RESTful API. Flask добре підходить для створення RESTful API, яке є стандартним підходом для розробки бекенду вебзастосунку. Він надає зручні засоби для визначення маршрутів, обробки запитів та передачі даних у форматі JSON;

г) розширюваність: Flask має велику кількість розширень та плагінів, які дозволяють додавати нові функціональності до вашого бекенду. Це означає, що ви можете використовувати готові рішення для автентифікації, роботи з базами даних, кешування та багато іншого, що спрощує розробку і прискорює час виконання проєкту;

г) активна спільнота: Flask має велику та активну спільноту розробників, що означає, що ви можете легко знайти документацію, підтримку та приклади коду при виникненні питань або потребі у допомозі.

Для аналізу відео застосовується MediaPipe – це потужна і універсальна бібліотека для аналізу відео, розроблена Google. Вона надає розробникам широкий спектр функціональності і інструментів для розпізнавання облич, виявлення об'єктів, відстеження руху, розпізнавання жестів, розпізнавання вимірів об'єктів і багато іншого. MediaPipe пропонує готові моделі і компоненти, які можуть бути використані для розробки різноманітних завдань обробки відео. Вона має простий і зрозумілий інтерфейс програмування (API),

що дозволяє розробникам швидко і легко використовувати її функціональність. Одна з особливостей MediaPipe є підтримка різних платформ, включаючи мобільні пристрої (Android, iOS) і робочі станції (Windows, macOS, Linux). Це дозволяє розробляти програми для аналізу відео, які можуть працювати на різних пристроях і платформах. Бібліотека MediaPipe також підтримує роботу з різними форматами медіа, включаючи відео з файлів і потокове відео. Вона забезпечує гнучкість в обробці відеоданих з різних джерел. Загальний висновок полягає в тому, що MediaPipe є потужною і зручною у використанні бібліотекою для розробки програм для аналізу відео. Вона надає розробникам широкий спектр інструментів і можливостей, що дозволяє ефективно виконувати різноманітні завдання обробки відео.

Вибір Rest API для взаємодії між клієнтом та сервером здійснюється з таких причин: Rest API є стандартним і широко використовуваним підходом до розробки вебсервісів. Він забезпечує простоту і зручність взаємодії шляхом використання HTTP протоколу та стандартних методів передачі даних. Rest API дозволяє передавати дані у популярних форматах, таких як JSON або XML, що спрощує обробку та обмін даними між клієнтом та сервером. Цей підхід дозволяє гнучко розділити функціональність між клієнтом та сервером, що полегшує масштабування системи та розвиток окремих компонентів незалежно один від одного. Rest API також підтримує можливість кешування результатів запитів, що дозволяє зменшити навантаження на сервер та покращити продуктивність системи. Завдяки використанню Rest API, можливо забезпечити взаємодію з різними клієнтами, включаючи вебзастосунок, мобільні додатки, IoT – пристрої та інші, що робить його універсальним і гнучким рішенням для забезпечення комунікації між клієнтом та сервером. У підсумку, Rest API є популярним вибором для реалізації взаємодії між клієнтом та сервером завдяки своїй стандартності, простоті використання, масштабованості та підтримці різних типів клієнтів.

Вибір пав на вебзастосунок тому що він може бути обґрунтований кількома причинами. Вебзастосунок є кросплатформеними і можуть

працювати на різних операційних системах без необхідності розробки окремих версій. Крім того, вебзастосунку легше оновлювати, оскільки оновлення можуть бути реалізовані централізовано на сервері. Користувачам не потрібно встановлювати додаткове програмне забезпечення, оскільки вебзастосунки працюють у веббраузерах, що забезпечує зручний доступ. Вебзастосунки також мають доступ до різноманітних вебсервісів і можуть інтегруватися з іншими системами. Розгортання вебзастосунків на вебсерверах полегшує їх поширення серед користувачів і забезпечує більший контроль над доступом.

Для отримання відео за посиланням з YouTube була використана бібліотека `youtubedl` (YouTube-dl). Ця бібліотека є потужним інструментом для скачування відео та аудіо з різних відеохостингових платформ, включаючи YouTube. YouTube-dl надає можливість отримати пряме посилання на відео за його URL-адресою. Це дозволяє розробникам використовувати цю бібліотеку для отримання відео-файлів з YouTube та подальшого їх використання в вебзастосунках. Використання `youtubedl` у вебзастосунку дозволяє зручно та ефективно отримувати відео з YouTube за їх URL-адресами, що відкриває широкі можливості для розробки функціональності, пов'язаної з аналізом та обробкою цих відео.

Для розробки вебінтерфейсу були використані HTML і CSS. HTML є основною мовою розмітки, використовуваною для створення структури та вмісту вебсторінок. Він дозволяє визначити різноманітні елементи, такі як заголовки, параграфи, таблиці, форми, зображення і посилання. CSS використовується для визначення зовнішнього вигляду та стилізації вебсторінок. Він дозволяє змінювати кольори, шрифти, розміри, відступи, фонові зображення та інші візуальні властивості елементів HTML. CSS дозволяє розміщувати елементи на сторінці, встановлювати розташування, створювати анімації та забезпечувати адаптивність до різних пристроїв та екранних розмірів. Використання HTML і CSS дозволяє розробникам створювати привабливий, структурований та користувачами зручний

вебінтерфейс. HTML відповідає за структуру і семантику сторінки, в той час як CSS забезпечує її оформлення і вигляд. Разом вони дозволяють створювати естетично привабливі та функціональні вебінтерфейси для взаємодії з користувачами.

Вебзастосунок використовував фреймворк *Bulma* для побудови свого користувацького інтерфейсу. *Bulma* є одним з популярних CSS-фреймворків, який дозволяє швидко та ефективно стилізувати вебсторінки без необхідності писати власний CSS-код з нуля. *Bulma* має простий та зрозумілий синтаксис класів, що полегшує використання і навчання фреймворку. Це дозволяє розробникам швидко створювати стильові елементи і компоненти. Підтримує респонсивний дизайн, що означає, що сторінки, побудовані за допомогою фреймворку, автоматично пристосовуються до різних розмірів екрану та пристроїв. Це забезпечує коректне відображення контенту на будь-яких пристроях, включаючи комп'ютери, планшети та смартфони. Пропонує широкий набір класів і компонентів, які можна комбінувати та налаштовувати за потребою. Він також дозволяє додавати власні стилі і розширювати функціональність за допомогою власних CSS-правил. *Bulma* використовує *Flexbox* для розташування елементів на сторінці, що дозволяє легко контролювати розміщення і вирівнювання елементів на різних екранах і в різних контекстах. Для вебзастосунку також був використаний JavaScript, який є мовою програмування, широко використовуваною для розробки динамічного та інтерактивного контенту на вебсторінках. JavaScript дозволяє додавати різноманітну функціональність до вебзастосунку. Він використовується для реалізації взаємодії з користувачем, обробки подій, валідації даних, асинхронного завантаження і обміну даними з сервером, анімацій та багатьох інших завдань. За допомогою JavaScript можна динамічно змінювати вміст сторінки, маніпулювати елементами DOM (Document Object Model), валідувати форми, виконувати AJAX-запити до сервера, маніпулювати анімаціями та багато іншого. JavaScript є важливою складовою розробки вебзастосунку, оскільки він дозволяє зробити сторінку динамічною та

інтерактивною. Використання JavaScript разом з HTML і CSS дозволяє створювати багатофункціональні вебзастосунку з високою ступенем взаємодії та зручності для користувачів.

Для реалізації постійної трансляції в одну сторону, можна використовувати технологію під назвою Server-Sent Events (SSE). Server-Sent Events – це механізм, який дозволяє серверу відправляти безперервні оновлення клієнту через однонаправлене з'єднання. Це означає, що сервер може ініціювати передачу даних клієнту без його прямої запити. SSE забезпечує постійне підключення, що дозволяє передавати дані у реальному часі. У вебзастосунку, що реалізує постійну трансляцію в одну сторону, сервер буде неперервно відправляти оновлення до клієнта через SSE. Це можуть бути дані, які оновлюються в реальному часі, такі як повідомлення, статуси, дані сенсорів або будь-яка інша інформація, яку потрібно передавати безперервно. Для реалізації SSE вебзастосунку, на стороні сервера використовується спеціальний механізм, який створює SSE з'єднання і надсилає оновлення до клієнта за допомогою спеціальних HTTP-заголовків. Клієнт, у свою чергу, прослуховує SSE з'єднання і отримує оновлення в реальному часі. Ця технологія особливо корисна в сценаріях, де потрібно постійно оновлювати дані на стороні клієнта без необхідності повторних запитів. Вона забезпечує ефективну та економічну передачу даних, оскільки немає необхідності в постійних запитах з боку клієнта. Застосування постійної трансляції в одну сторону за допомогою SSE може включати такі сценарії, як реальні часи оновлення сторінок, сповіщення, моніторинг даних, онлайн-чати та багато іншого, де важлива постійна передача оновлень від сервера до клієнта без зайвих запитів.

Ще був застосован WebStorm – це інтегроване середовище розробки (IDE) від компанії JetBrains, призначене для розробки вебзастосунку. Вона володіє широким набором функцій та інструментів, які допомагають спростити та прискорити процес розробки. WebStorm підтримує різні технології та мови програмування, які використовуються у веброботці,

включаючи HTML, CSS, JavaScript, TypeScript, PHP, Node.js та інші. Він забезпечує зручне середовище розробки з підсвічуванням синтаксису, автодоповненням коду, інтегрованим налагоджувачем, системою контролю версій та іншими корисними функціями. Однією з головних особливостей WebStorm є можливість автоматичного виявлення та виправлення помилок у коді, що допомагає розробникам підвищити ефективність і якість свого коду. Він також забезпечує інтеграцію з популярними фреймворками та бібліотеками, такими як React, Angular, Vue.js, та надає інструменти для роботи із системами збирання та тестування. WebStorm має дружній інтерфейс користувача, який дозволяє розробникам комфортно працювати і керувати своїм проектом. Він забезпечує можливість швидко навігувати за кодом, здійснювати рефакторинг, аналізувати продуктивність програми та багато іншого. В цілому, WebStorm є потужним та зручним інструментом для веброзробки, який допомагає розробникам створювати якісні та продуктивні вебзастосунку.

У розробці вебзастосунку була використана операційна система Windows. Windows є однією з найпопулярніших операційних систем, що широко використовуються як для особистого використання, так і в комерційній сфері. Windows забезпечує зручне та інтуїтивно зрозуміле середовище для розробки вебзастосунку. Вона надає розробникам доступ до різноманітних інструментів, бібліотек та фреймворків, які дозволяють ефективно створювати, налагоджувати та тестувати вебзастосунку. Операційна система Windows має широку сумісність із різними інструментами розробки, мовами програмування та серверними технологіями, що робить її популярним вибором для веброзробки. Більшість інструментів та фреймворків, таких як Visual Studio, WebStorm, Sublime Text, Git та багато інших, доступні для роботи під Windows. Windows також забезпечує підтримку різних серверних технологій, таких як Apache, Nginx, Microsoft IIS та інших, що дозволяє розробляти та розгортати вебпрограми на різних серверних платформах. Загалом вибір операційної системи Windows для розробки

вебпрограми обумовлений її широкою популярністю, зручністю використання, підтримкою різноманітних інструментів розробки та сумісністю з різними технологіями веброзробки.

Для розробки вебзастосунку був використаний npm. Npm є пакетним менеджером для мови програмування JavaScript, що дозволяє розробникам управляти залежностями та встановлювати сторонні пакети для своїх проєктів. Npm є частиною екосистеми Node.js та широко використовується у веброзробці. Він надає доступ до тисяч пакетів, які містять готовий код, модулі та ресурси, які можна використовувати у проєкті. Це спрощує розробку, оскільки дозволяє використовувати вже створені рішення та уникнути повторного винаходу велосипеда. За допомогою npm розробники можуть легко управляти залежностями проєкту. Вони можуть вказати необхідні пакети та їх версії у файлі `package.json`, а потім npm автоматично завантажить та встановить ці пакети, а також їх залежність. Це робить процес встановлення та оновлення пакетів більш зручним та автоматизованим. Крім того, npm надає команди для управління проєктом, такі як запуск скриптів, тестування, збирання та деплоймент. Розробники можуть використовувати npm для запуску сервера розробки, складання проєкту, запуску тестів або будь-яких інших завдань користувача. Npm також має широку підтримку спільноти, що означає наявність активної розробки пакетів, оновлення та підтримку. Це дозволяє використовувати актуальні та надійні рішення для розробки вебзастосунків. В цілому, використання npm у розробці вебзастосунку дозволяє розробникам ефективно управляти залежностями, встановлювати сторонні пакети та використовувати вже створені рішення для прискорення розробки та підвищення якості проєкту.

Для розробки вебзастосунку був використаний pip – стандартний пакетний менеджер для мови програмування Python. pip надає розробникам зручний спосіб встановлення, керування та оновлення пакетів, необхідних для розробки програм на Python. За допомогою pip розробники можуть легко встановлювати сторонні пакети Python із офіційного репозиторію PyPI (Python

Package Index) або інших джерел. Він автоматично керує залежностями пакетів, дозволяючи легко задовольняти вимоги до версій та підтримувати цілісність оточення розробки. Pip також надає можливість керування віртуальними середовищами Python, які дозволяють розробникам ізолювати проекти один від одного та керувати їх залежностями незалежно. Це особливо корисно, коли у вас є кілька проектів з різними вимогами до пакетів та версій. Крім того, pip забезпечує простий спосіб встановлення пакетів з локальних файлів або їх видалення, пошук та оновлення пакетів, створення та керування віртуальними оточеннями. Також він дозволяє легко керувати залежностями у проекті за допомогою файлу requirements.txt, який містить список пакетів та їх версій для встановлення. Велика перевага pip полягає в його популярності та активній спільноті розробників. Це забезпечує широкий вибір пакетів, бібліотек та фреймворків, доступних для використання у веброботці Python. У результаті, використання pip у розробці вебзастосунку дозволяє зручно керувати пакетами та їх залежностями, встановити необхідні інструменти та бібліотеки, та забезпечити гнучкість та ефективність у процесі розробки.

### 3.2 Програмна реалізація

Розробка вебзастосунку для аналізу відео за посиланням може бути розділена на наступні етапи:

а) завантаження відео за посиланням чи з відео камери. Вебзастосунок повинен мати можливість визначити джерело відео, з якого буде здійснюватись завантаження. Це може бути вбудована відеокамера на пристрої, який запускає застосунок, або зовнішній джерело, таке як відеофайл, відеопотік з вебкамери або посилання на відео з інтернету. Вебзастосунок повинен мати можливість отримати доступ до відеокамери пристрою або обробити введене посилання, якщо користувач вказав зовнішнє джерело відео.

Це може вимагати використання відповідних API або пакетів, які дозволяють отримати доступ до відео потоку або виконати завантаження відео за посиланням. Після визначення джерела відео, застосунок повинен здійснити процес завантаження відео. Це може включати зчитування відеофайлу з локального джерела, отримання відеопотоку з вебкамери або виконання HTTP-запиту до вказаного посилання для отримання відеофайлу. Після завантаження відео, застосунок може проводити різні операції обробки зображення для досягнення бажаного ефекту. Це може включати фільтрацію, корекцію кольору, виявлення об'єктів, аугментовану реальність та інші техніки обробки. Результат обробки може бути збережений в новому відеофайлі або показаний користувачу безпосередньо в вебінтерфейсі застосунку;

б) попередня обробка відео для аналізу. Цей етап включає в себе ряд дій, спрямованих на підготовку відео для подальшого аналізу. Відео може бути перетворено в потрібний формат або розмір, якщо це необхідно для подальшої обробки чи аналізу. Застосовуються фільтри або алгоритми для зниження шуму та покращення якості зображення. Відео може бути піддане корекції яскравості, контрастності та кольоровій палітрі для стандартизації його властивостей. Якщо необхідно, можуть виділятися конкретні області на кадрах для подальшого аналізу або обробки;

в) аналіз відео з використанням MediaPipe. MediaPipe надає готові моделі для виявлення різних об'єктів, таких як обличчя, руки, тіла, ознаки руху тощо. Ці моделі здатні точно визначати місцезнаходження та характеристики об'єктів на кадрі. Після виявлення об'єктів можна проводити подальший аналіз їх руху, форми, розміру, кольору та інших параметрів. Це може включати вимірювання відстаней, розпізнавання жестів, виявлення емоцій тощо. Результати аналізу можуть бути відображені на відео або відображені у вигляді графіків, діаграм або інших візуальних елементів для зручного сприйняття користувачем;

г) відправки відео на клієнт. Для відправки відео на клієнт у вашому вебзастосунку використав Flask Response Використання Flask Response дозволяє створити відповідь сервера, яка містить відео дані та налаштування для їх правильного відображення на клієнті. Ви можете встановити заголовки відповіді для визначення типу контенту, наприклад, «video/mp4» або «video/webm», і передати самі відео дані як тіло відповіді.

Інтерфейс розробленого застосунку показано на рисунку 3.1. На сайті розташовано кнопки з відповідними підписами, які відповідають за аналіз відео за посиланням.

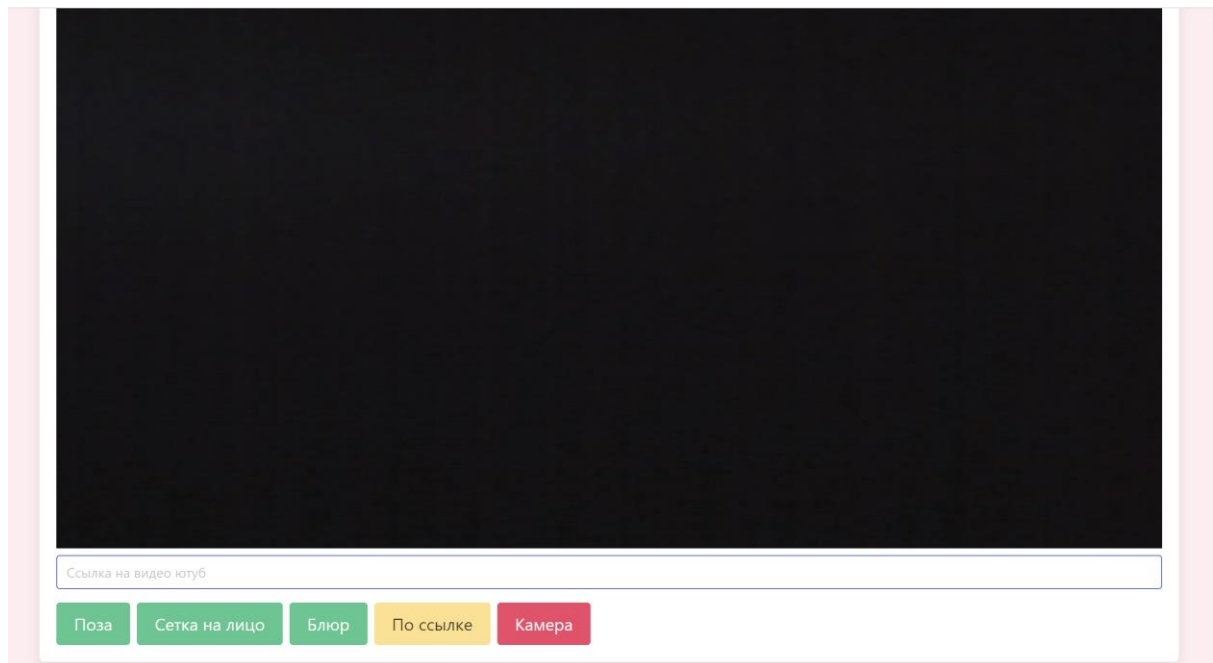


Рисунок 3.1 – Інтерфейс застосунку

Отже, при роботі з вебзастосунком користувачеві, у першу чергу, необхідно вибрати з якого джерела відео камери або за посиланням. Після вибору джерела відео для його подальшого аналізу користувачу будуть доступні різний функціонал для аналізу відео.

Дані можливості дозволять користувачеві експериментувати з різними способами аналізу відео, використовуючи розроблений функціонал застосунку.

### 3.3 Інструкція користувача

При запуску застосунку перед користувачем з'явиться сторінка, яке зображене на рисунку 3.2.

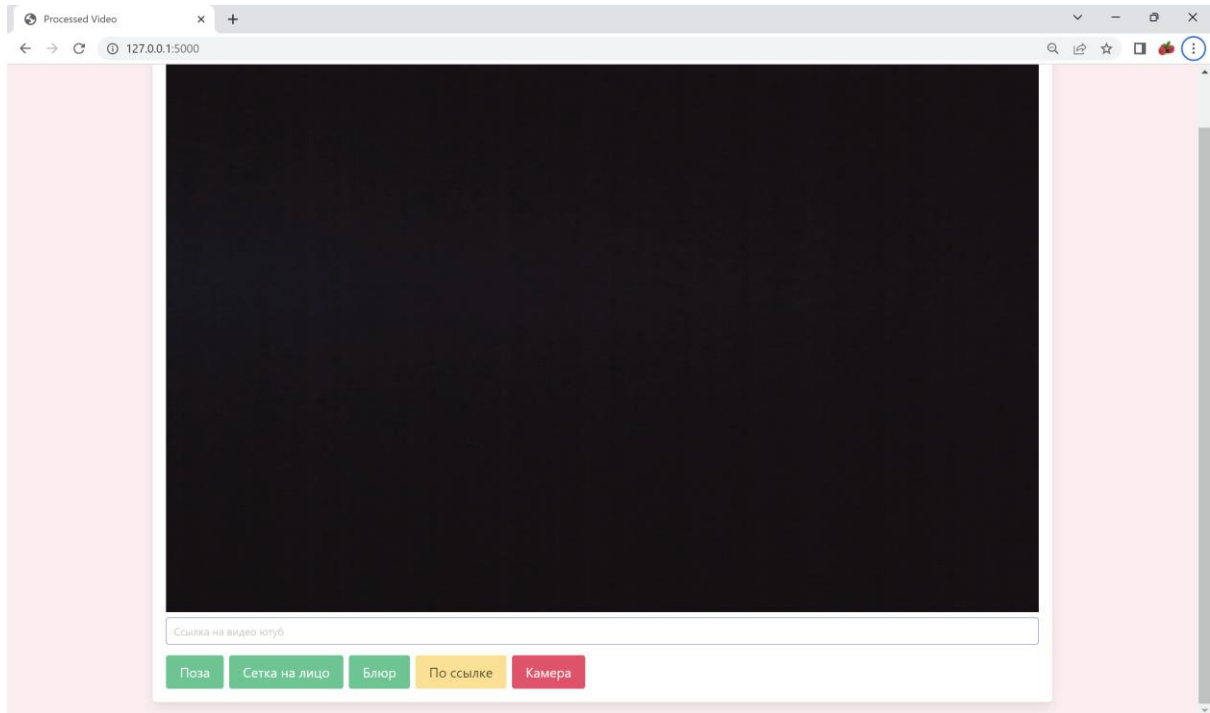


Рисунок 3.2 – Застосунок після запуску

Для того, щоб почати використовувати не обов'язково вставляти посилання на відео так як можливо застосувати камеру. Після кінцевого вибору джерела відео потрібно вибрати тип аналізу відео які зображені на рисунку 3.3.



Рисунок 3.3 – Кнопки для вибору типу аналізу відео

Після вибору який тип аналізу будуть застосований на відео, то тиснем на кнопку, наприклад на кнопку «Блур» та отримуємо результат на рисунку 3.4.

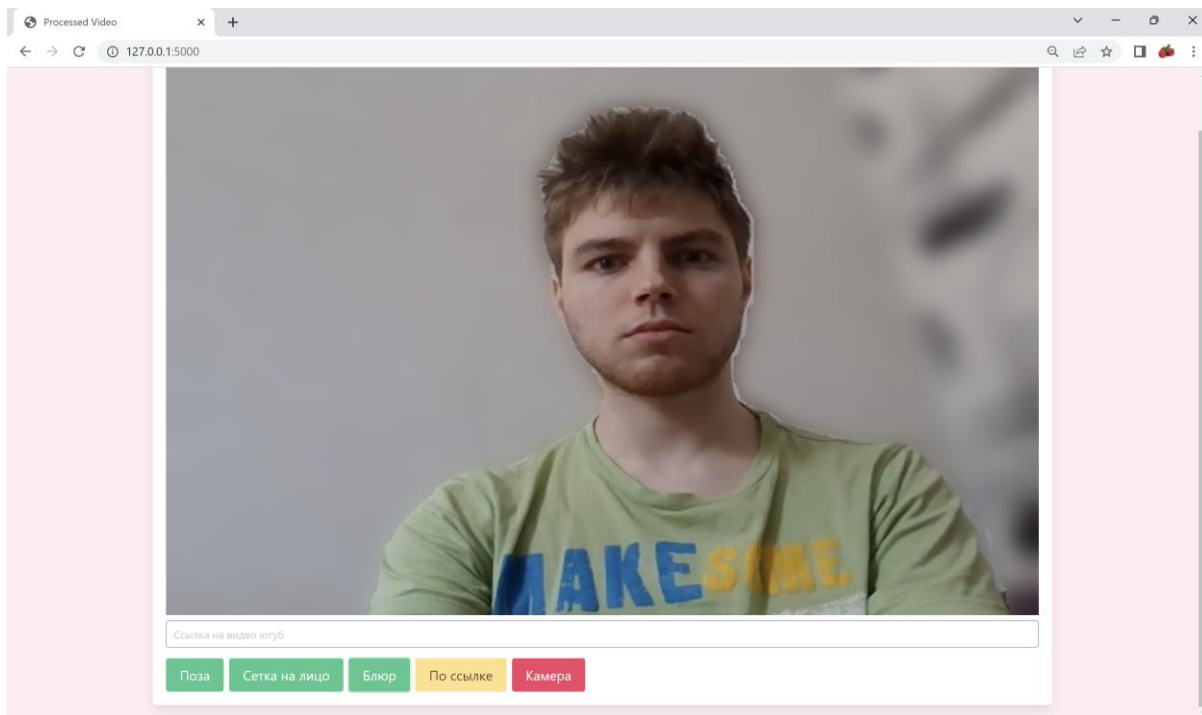


Рисунок 3.4 – Результат роботи аналізу відео типу «Блур»

Для аналізу відео за посиланням потрібно вставити посилання як на рисунку 3.5.



Рисунок 3.5 – Місце куди потрібно вставити посилання

Після того як вставили посилання, потрібно натиснути на якусь із кнопок аналізу відео які були представлені на рисунку 3.3. Після цих дій буде відображено результат аналізу відео типу «Маска на обличчя» приклад на рисунку 3.6.

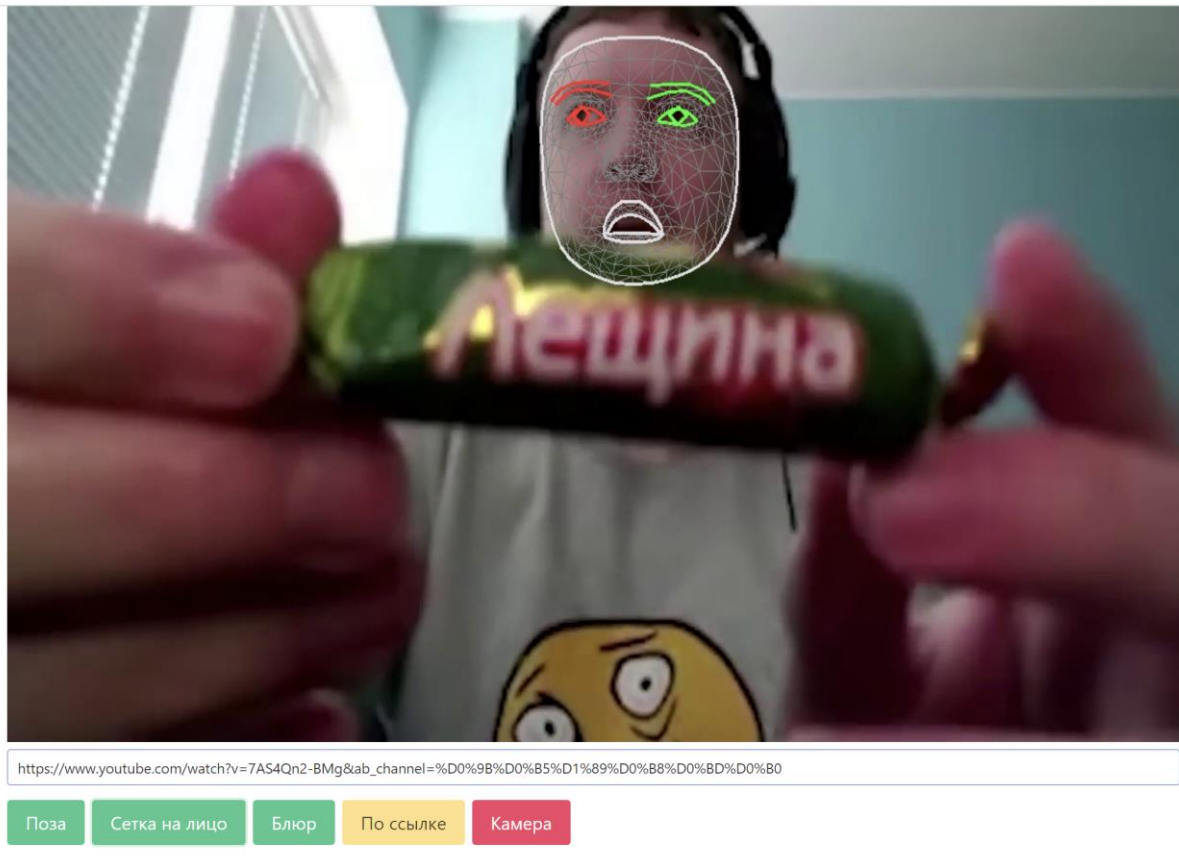


Рисунок 3.6 – Результат роботи аналізу відео типу «Маска на обличчя»

### 3.4 Тестування розробленої моделі

Тестування застосунку є важливою складовою розробки програмного забезпечення. Його основна мета – перевірити, чи працює застосунок так, як він повинен працювати, чи відповідає вимогам та очікуванням користувачів. Це допомагає виявити помилки, дефекти та недоліки, які можуть вплинути на функціональність та надійність програмного забезпечення. Тестування застосунку можна проводити на різних рівнях, включаючи модульне тестування окремих компонентів, інтеграційне тестування взаємодії між компонентами та системне тестування всього застосунку в цілому. Для цього розробники створюють тестові сценарії, дефінують очікувані результати та виконують тестові набори даних. Під час тестування важливо перевірити правильність функціонування всіх функцій та функціональностей застосунку, а також переконатися, що він працює стабільно та безперебійно. Доцільно

також проводити тестування на різних платформах та в різних умовах, щоб переконатися в його сумісності та надійності. Результати тестування допомагають виявити та усунути будь-які проблеми, а також покращити функціональність та якість застосунку перед його релізом. Це сприяє забезпеченню задоволення користувачів та покращенню їх досвіду використання програмного забезпечення.

### 3.4.1 Причини проведення тестування

Тестування проводиться з різних причин, які сприяють покращенню якості та надійності програмного забезпечення. Основні причини проведення тестування включають:

а) виявлення помилок та дефектів. Тестування допомагає виявити помилки, дефекти та недоліки в програмному забезпеченні. Це можуть бути проблеми в роботі функцій, некоректна обробка даних, помилки в алгоритмах тощо. Виявлення та усунення цих проблем перед випуском програмного забезпечення допомагає запобігти негативним наслідкам для користувачів та забезпечує високу якість продукту;

б) впевненість у функціональності. Тестування допомагає перевірити, чи працюють функції та функціональності програмного забезпечення так, як вони повинні. Це дає розробникам та замовникам впевненість, що програмне забезпечення виконує необхідні завдання та задовольняє потреби користувачів;

в) впровадження змін та покращень. Тестування допомагає перевірити, як нові зміни, вдосконалення або оновлення впливають на роботу програмного забезпечення. Це дозволяє оцінити ефективність та якість змін, а також виявити можливі непередбачені наслідки та проблеми, які можуть виникнути;

г) забезпечення сумісності та надійності. Тестування допомагає переконатися, що програмне забезпечення працює на різних платформах, в різних умовах та з різними конфігураціями. Це дозволяє забезпечити сумісність та надійність програми для різних користувачів та середовищ;

г) покращення користувацького досвіду. Тестування допомагає виявити та виправити проблеми, які можуть вплинути на користувацький досвід. Це можуть бути проблеми з інтерфейсом, швидкодією, зручністю використання тощо. Вдосконалення цих аспектів сприяє покращенню задоволення користувачів від використання програмного забезпечення.

### 3.4.2 Основні етапи тестування

Основні етапи тестування включають:

а) планування тестування: На цьому етапі визначаються мета тестування, його обсяг та стратегія. Розроблюються тестові плани, визначаються критерії успішності тестування;

б) розробка тестових сценаріїв: Створюються детальні сценарії тестування, які описують послідовність дій, що будуть виконуватися під час тестування. Вони включають в себе вхідні дані, очікувані результати та критерії прийнятності;

в) підготовка тестових даних: Забезпечуються необхідні дані для виконання тестів. Це можуть бути тестові набори даних, сценарії введення, тестові бази даних тощо;

г) виконання тестів: Тестові сценарії виконуються на програмному забезпеченні, відповідно до запланованого плану тестування. Результати тестів реєструються, включаючи виявлені проблеми та помилки;

г) аналіз результатів: Оцінюються результати тестування та аналізуються виявлені проблеми. Визначається серйозність та пріоритетність проблем для подальшого виправлення;

- д) виправлення проблем: Розробники виправляють виявлені проблеми та дефекти у програмному забезпеченні;
- е) повторне тестування: Виправлені версії програмного забезпечення піддаються повторному тестуванню, щоб перевірити, чи були виправлені проблеми та чи працює програмне забезпечення належним чином;
- є) завершення тестування: Після успішного виконання тестування, коли програмне забезпечення відповідає вимогам та надійність на достатньо високому рівні, тестування завершується.

Процес тестування може варіюватися в залежності від специфіки проєкту та методології розробки. Важливо також враховувати тестування на ранніх етапах розробки та використовувати автоматизоване тестування, де це можливо, для забезпечення більшої ефективності та повторюваності тестів.

### 3.4.3 Тестування застосунку

Тестування розробленого застосунку слід почати від самого запуску. Після запуску з'являється сторінка зображене на рисунку 3.7. На сторінці повинні бути кнопки з вибором типу для аналізу відео, кнопки вибору з якого джерела буде лічений потік відео, сторка в яку задють посилання на відео та вікно з відео.

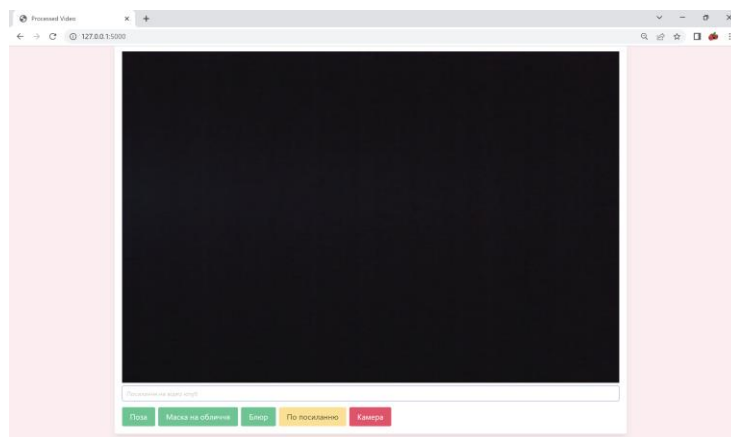


Рисунок 3.7 – Вікно застосунку після запуску

За стандартом використовується камера, так що має відобразитись відео з камери приклад роботи на рисунку 3.8.

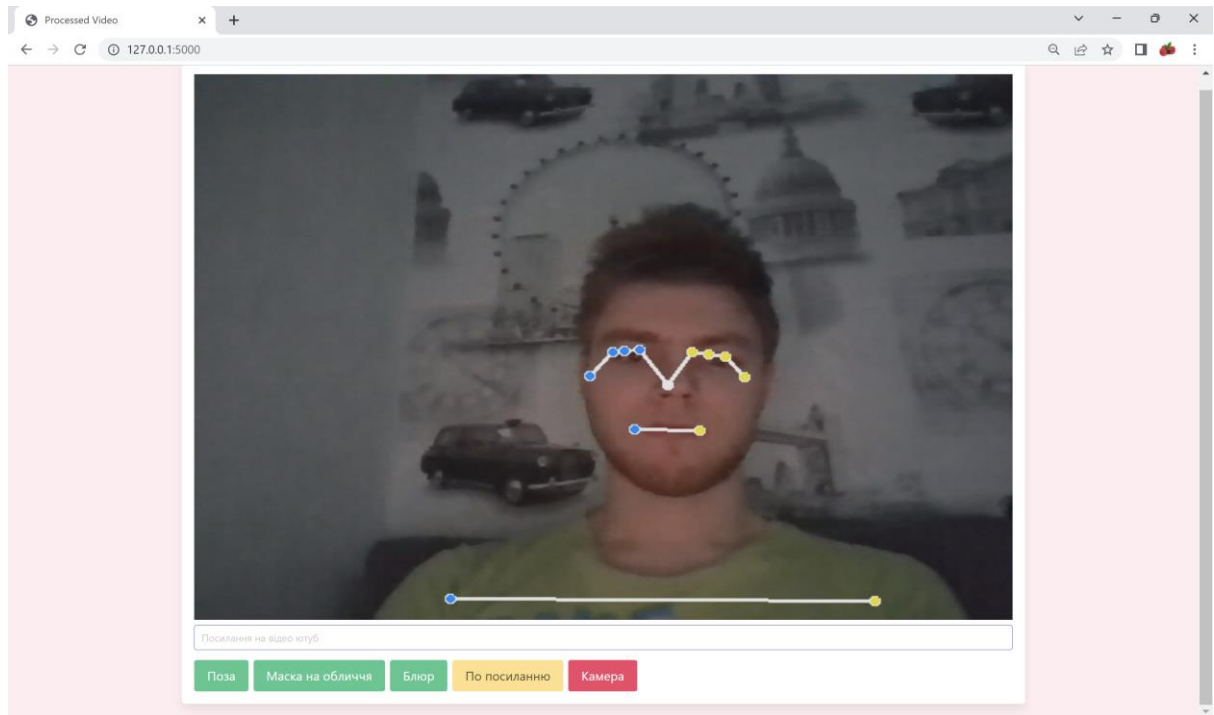


Рисунок 3.8 – Приклад роботи при натисканні на кнопку «Камера»

Під час попереднього тестування застосунку не виявлено жодних проблем з його функціональністю. Тепер необхідно спрямувати увагу на тестування аналізу відео, щоб переконатися, що ця функція працює належним чином. Завданням тестування аналізу відео буде перевірка точності, швидкості та надійності алгоритмів. Також необхідно переконатися, що відео обробляються згідно з очікуваннями користувачів і заданими вимогами.

Після натискання на кнопку «Поза» то на відео с відео почнуть малювати кості чоловіка приклад роботи зображена на рисунку 3.9.

Отже аналіз відео для будування костей на тілі чоловіка працює гарно, але на рисунку 3.10 кістки вишикувалися неправильно це могло статися з декілька причин:

а) перша причина. Це із-за того що скріншот було зроблено в час коли кістки перестроювалися;

б) друга причина. Це із-за того що алгоритм не зміг визначити тіло побачив лише нижню частину тіла.

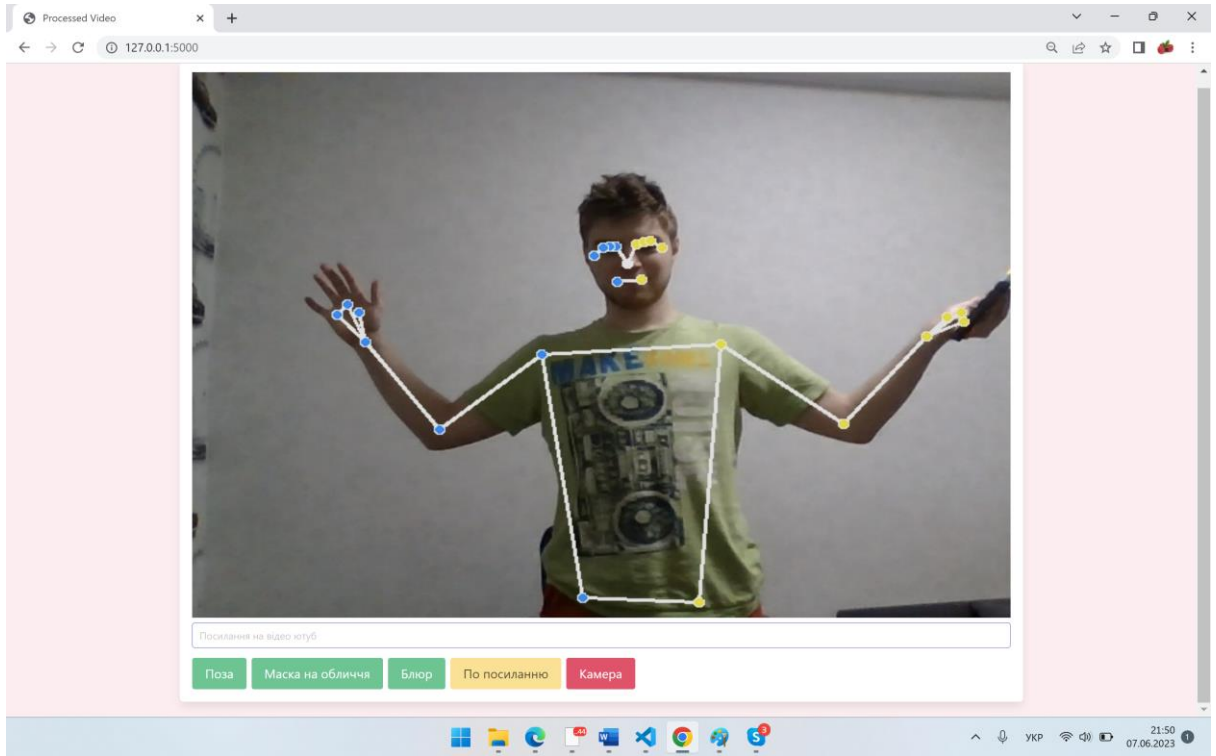


Рисунок 3.9 – Приклад роботи при натисканні на кнопку «Поза»

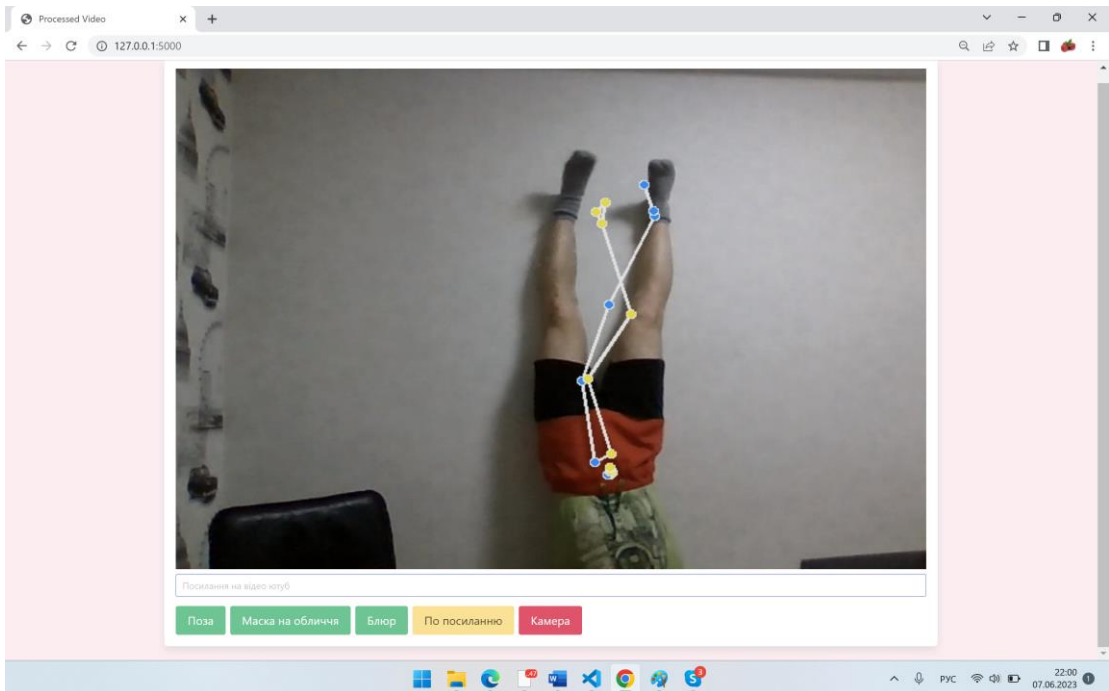


Рисунок 3.10 – Приклад роботи при натисканні на кнопку «Поза»

Після натискання на кнопку «Блор» то на відео с відео почнуть малювати кості чоловіка приклад роботи зображена рисунку 3.11.

На рисунку 3.11 бачимо, що функція по кнопці «Блор» працює так як треба.

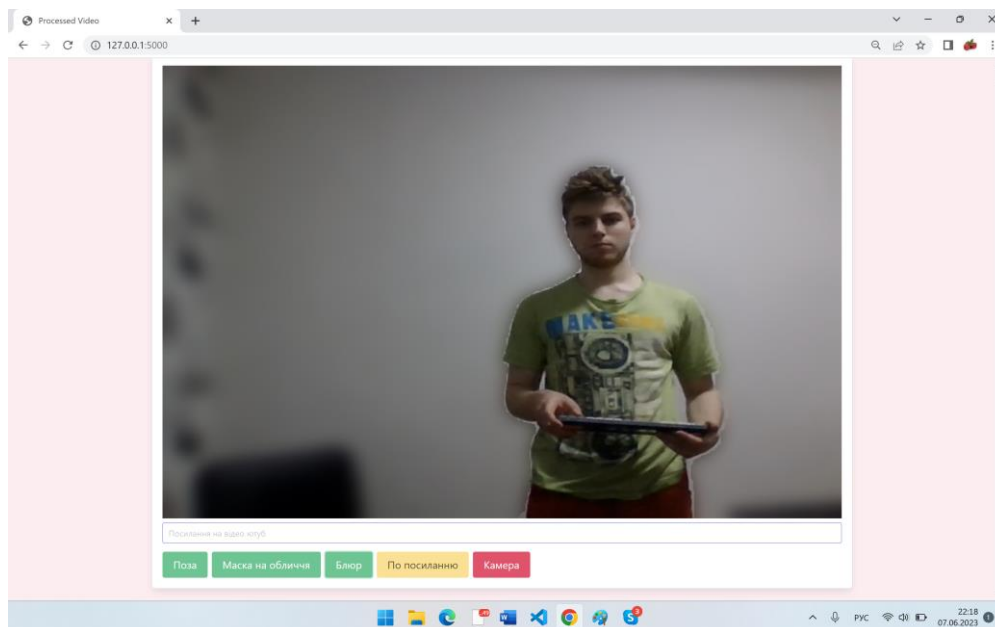


Рисунок 3.11 – Приклад роботи при натисканні на кнопку «Блор»

Після натискання на кнопку «Маска на обличчя» то на відео с відео почнуть малювати маску на лиці чоловіка приклад роботи зображена рисунку 3.12.

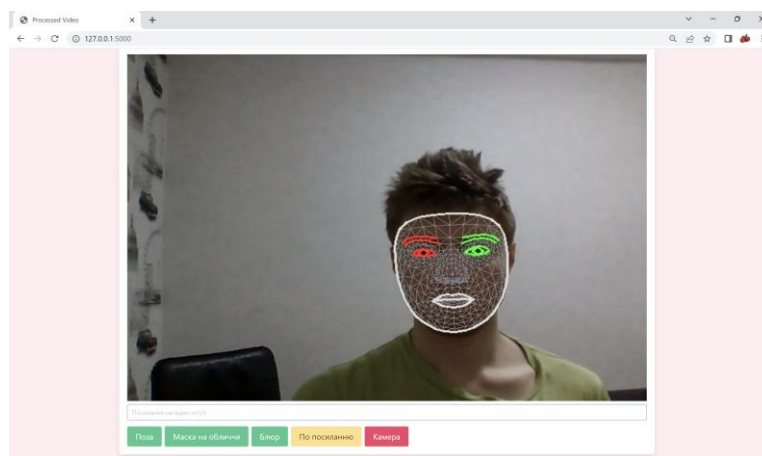


Рисунок 3.12 – Приклад роботи «Маска на обличчя»

Зараз будемо тестувати відео за посиланням. Отже для того щоб почати тестувати ми вставляємо посилання в строку, якщо цей рядок буде порожнім, то буде запускатися відео з камери.

Після натискання на кнопку «Поза» то на відео с відео почнуть малювати кості чоловіка приклад роботи зображена рисунках 3.13-3.16.

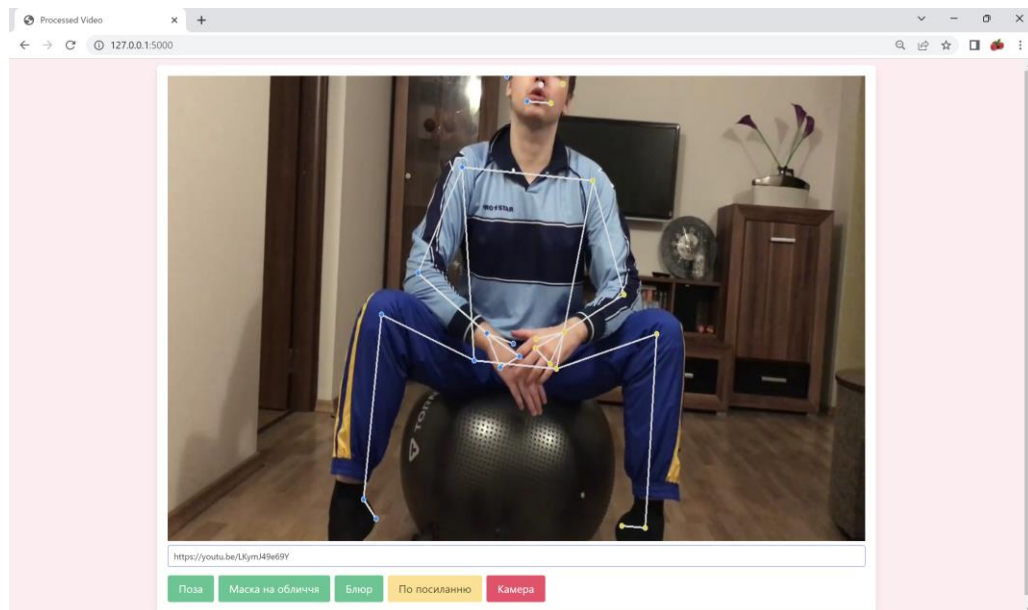


Рисунок 3.13 – Приклад роботи при натисканні на кнопку «Поза»

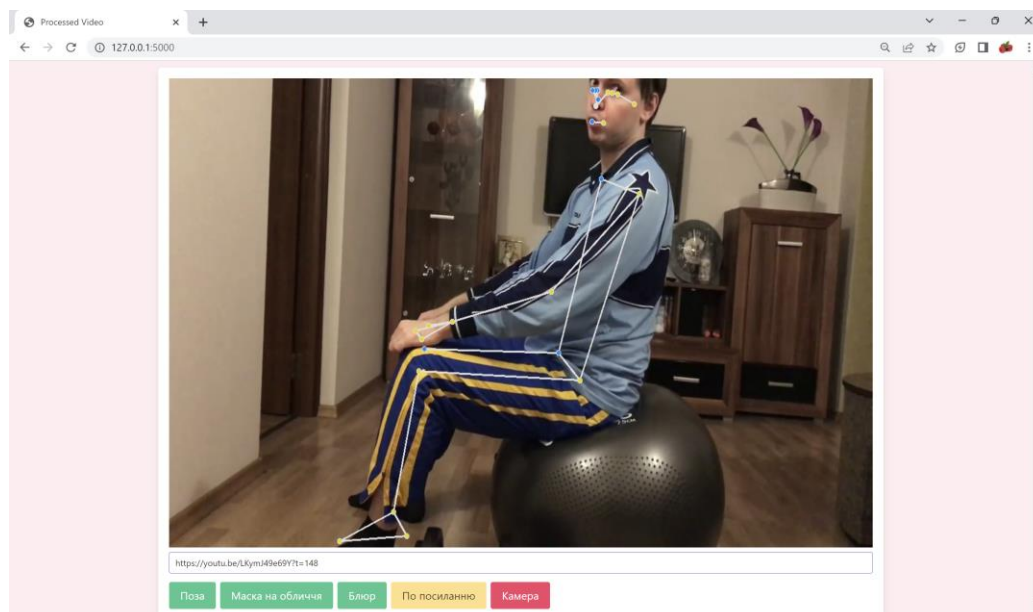


Рисунок 3.14 – Приклад роботи при натисканні на кнопку «Поза»

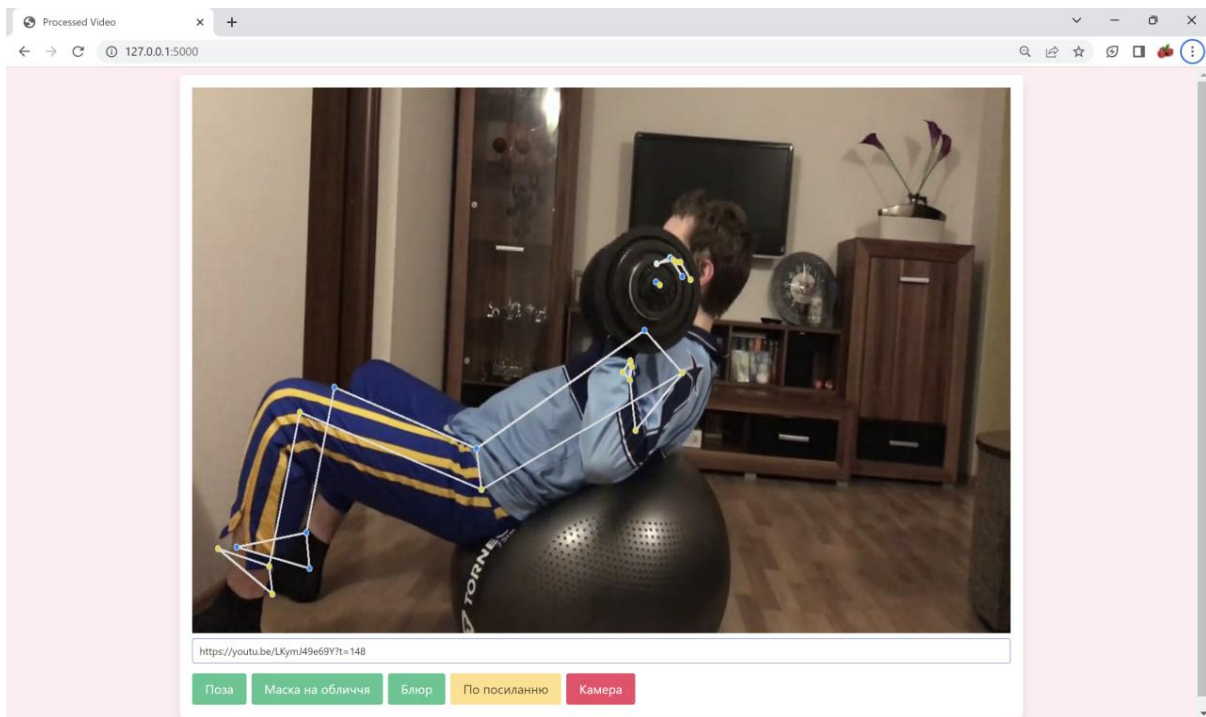


Рисунок 3.15 – Приклад роботи при натисканні на кнопку «Поза»

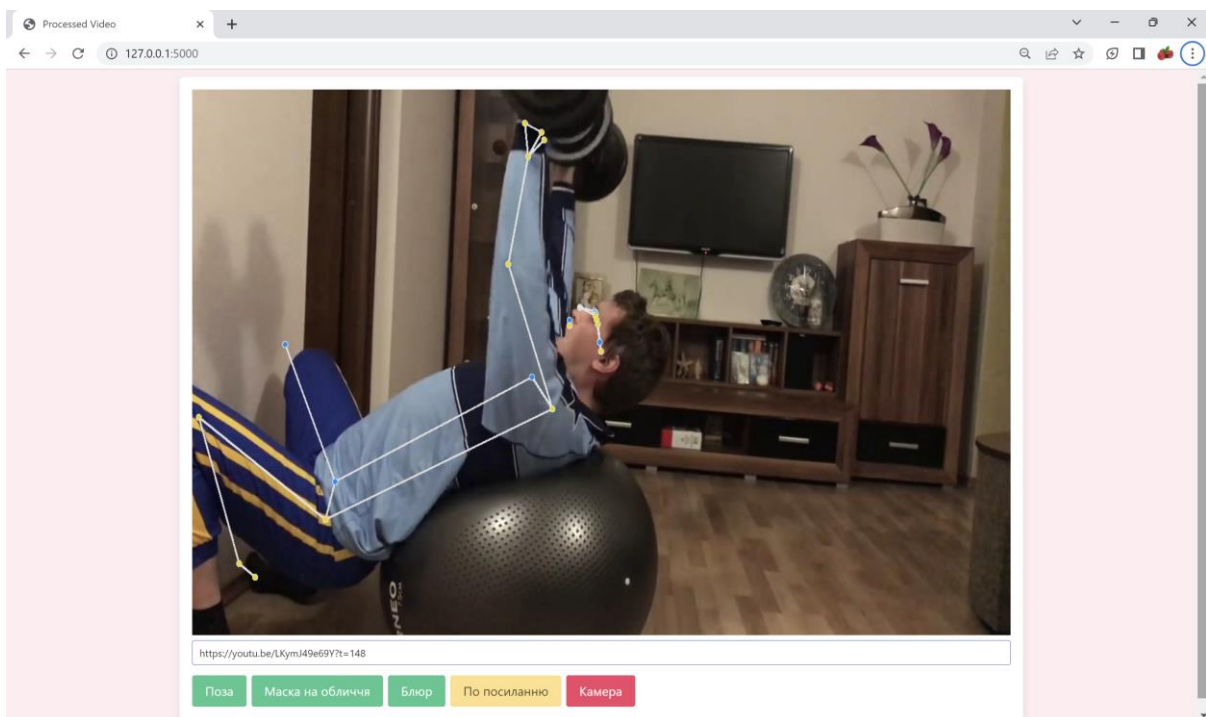


Рисунок 3.16 – Приклад роботи при натисканні на кнопку «Поза»

На рисунках 3.13-3.16 бачимо, що алгоритм справляється з поставленим завданням. навіть у складних місцях де не видно ноги.

Наприклад, на рисунку 3.14, то алгоритм домалював їй кістки, а на рисунках 3.15 та 3.16 не видно рук, але алгоритм розуміє їх розташування і малює кістки. Це свідчить про високу роботоздатність і точність алгоритму, що використовується в застосунку. Ці результати демонструють, що алгоритм аналізу відео в застосунку є ефективним і забезпечує точну реконструкцію фотореалістичних зображень навіть у випадках обмеженої видимості деяких частин тіла.

Після натискання на кнопку «Маска на обличчя» то на відео с відео почнуть малювати маску на лиці чоловіка приклад роботи зображена рисунку 3.17.

На зображеннях 3.17 та 3.18 бачимо, що алгоритм працює так, як потрібно, навіть у складній ситуації коли на голові шолом він може розпізнати обличчя людини і накласти на неї маску.

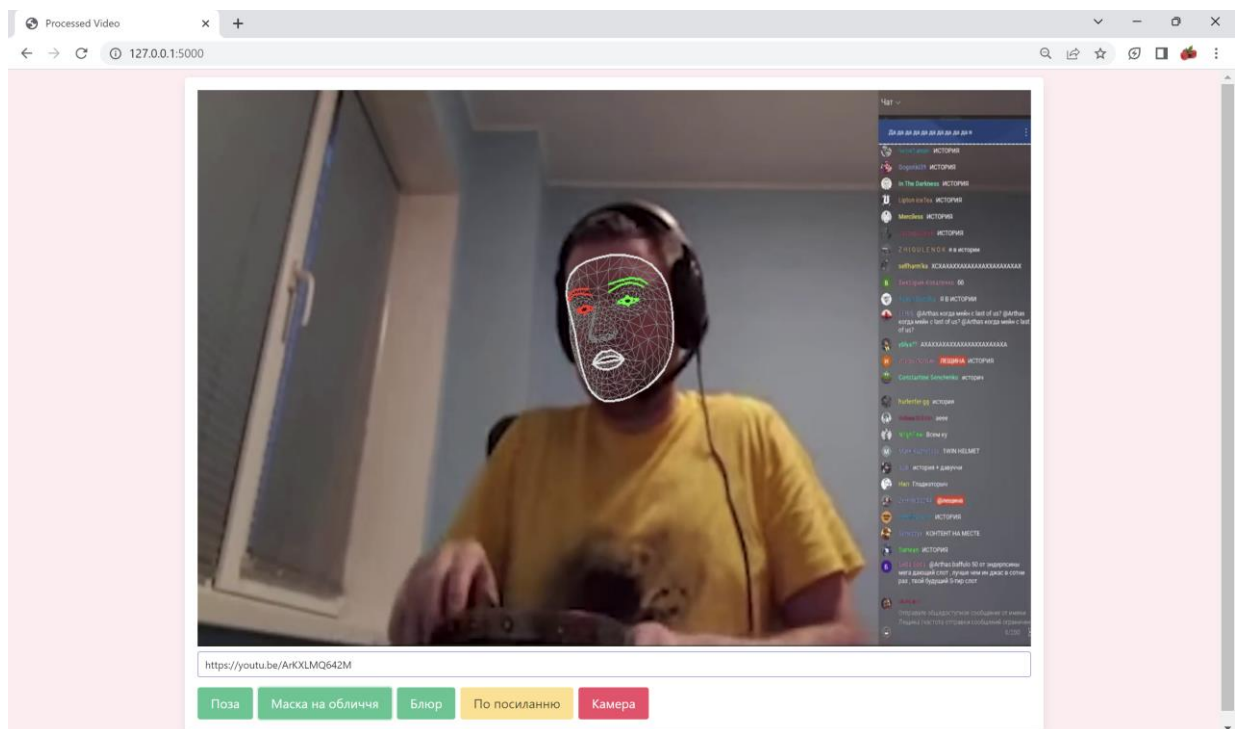


Рисунок 3.17 – Приклад роботи при натисканні на кнопку «Маска на обличчя»

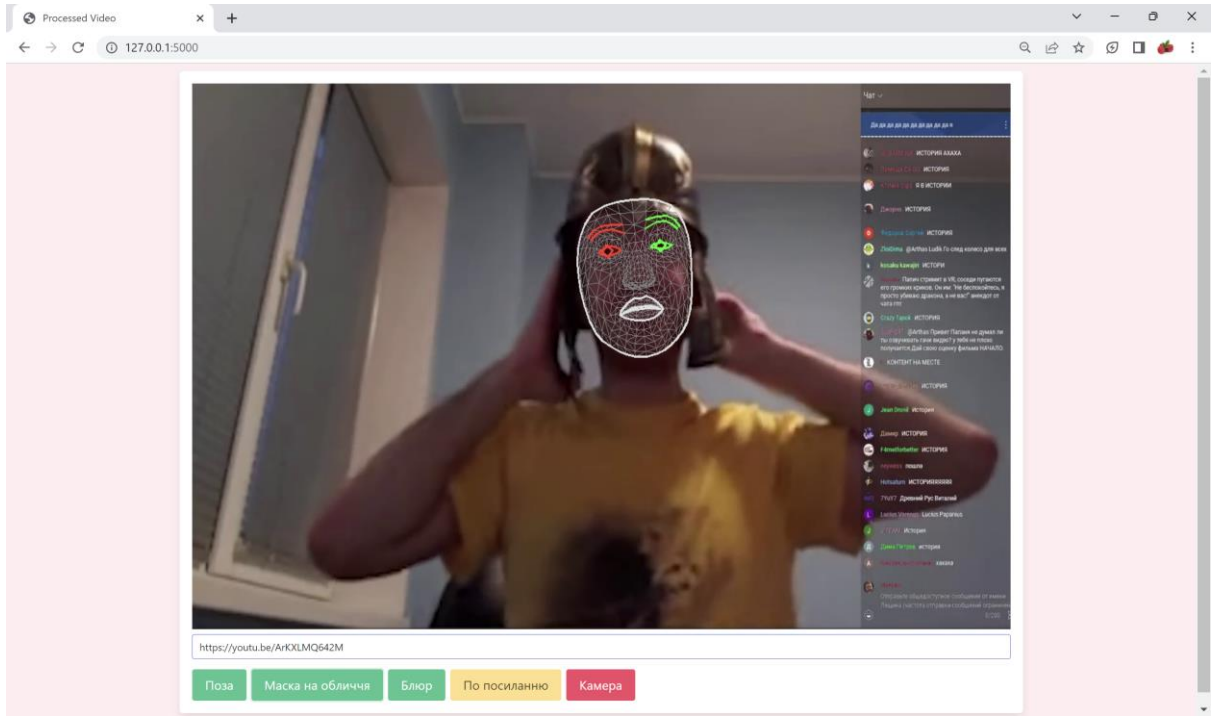


Рисунок 3.18 – Приклад роботи при натисканні на кнопку «Маска на обличчя»

## ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено застосунок для аналізу відео за посиланням з використанням алгоритму на основі Mediapipe. Метою роботи було реалізувати функціональність для обробки фотореалістичних зображень, а саме виявлення та аналіз людського тіла на відео.

Процес розробки включав кілька етапів, починаючи з завантаження відео з відеокамери або за посиланням, передання відео на клієнт, попередню обробку відео та аналіз використовуючи Mediapipe. Для створення вебінтерфейсу та розміщення застосунку на вебсервері були використані HTML, CSS і фреймворк Vulma.

Під час тестування було виявлено, що розроблений застосунок ефективно працює, забезпечуючи точну реконструкцію фотореалістичних зображень людського тіла. Навіть у випадках обмеженої видимості деяких частин тіла, алгоритм аналізу відео здатний «домалювати» відсутні елементи з використанням наявних даних.

Отже, на основі результатів дипломної роботи можна зробити висновок, що розроблений застосунок для аналізу відео за посиланням успішно виконує свої функції і має потенціал для використання у різних сферах, таких як відеоспостереження, медицина, реклама та інші. Дана робота вносить важливий вклад у розвиток області аналізу відео та візуальних систем.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is Pattern Recognition? A Gentle Introduction (2023). URL: <https://viso.ai/deep-learning/pattern-recognition/> (дата звернення 05.05.2023).
2. mediapipe/objectron.md GitHub. URL: <https://github.com/google/mediapipe/blob/master/docs/solutions/objectron.md> (дата звернення 10.05.2023).
3. mediapipe/hands.md GitHub. URL: <https://github.com/google/mediapipe/blob/master/docs/solutions/hands.md> (дата звернення 10.05.2023).
4. mediapipe/pose.md GitHub. URL: <https://github.com/google/mediapipe/blob/master/docs/solutions/pose.md> (дата звернення 10.05.2023).
5. Pose landmark detection guide | MediaPipe. URL: [https://developers.google.com/mediapipe/solutions/vision/pose\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/pose_landmarker) (дата звернення 12.05.2023).
6. Person Pose Landmarks detection using Mediapipe. URL: <https://mlhive.com/2021/11/person-pose-landmarks-detection-using-mediapipe> (дата звернення 15.05.2023).
7. Face and hand tracking in the browser with MediaPipe and TensorFlow.js. URL: <https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html> (дата звернення 17.05.2023).
8. Welcome to Flask — Flask Documentation (2.3.x). URL: <https://flask.palletsprojects.com/en/2.3.x/> (дата звернення 17.05.2023).
9. GitHub - google/mediapipe: Cross-platform, customizable ML solutions for live and streaming media. URL: <https://github.com/google/mediapipe> (дата звернення 17.05.2023).
10. Our Documentation. URL: <https://www.python.org/doc/> (дата звернення 17.05.2023).
11. [2104.11892v2] A Survey of Modern Deep Learning based Object Detection Models. URL: <https://arxiv.org/abs/2104.11892v2> (дата звернення 12.05.2023).

12. [2006.10204] BlazePose: On-device Real-time Body Pose tracking. URL: <https://arxiv.org/abs/2006.10204> (дата звернення 12.05.2023).
13. Gorokhovatskyi, V., Tvoroshenko, I., & Chmutov, Y. (2022). Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень. *Advanced Information Systems*, 6(3), 5-12.
14. brahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Cluster representation of the structural description of images for effective classification.
15. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). Image classification method modification based on model of logic processing of bit description weights vector. *Telecommunications and Radio Engineering*, 79(1).
16. Гороховатський, В.О., & Творошенко, І.С. (2021). Методи інтелектуального аналізу та оброблення даних: навч. посібник.
17. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2021). Methods of classification of images on the basis of the values of statistical distributions for the composition of structural description components. *IEEE Access*, 9, 92964-92973.
18. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Al-Dhaifallah, M. (2022). Classification of images based on a system of hierarchical features. *Computers, Materials & Continua*, 72(1), 1785-1797.
19. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image classification based on the Kohonen network and the data space modification.
20. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.
21. В.О. Гороховатський (2022). Комплекс навчально-методичного забезпечення навчальної дисципліни «Обробка зображень та мультимедіа»: навч. Посібник. *Харків: ХНУРЕ*.

22. Гороховатський, В. О., Передрій, О. О., Творошенко, І. С., & Марков, Т. Є. (2023). Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень.

23. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Al-Dhaifallah, M. (2022). Classification of images based on a system of hierarchical features. *Computers, Materials & Continua*, 72(1), 1785-1797. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.

24. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.

25. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. *Advances in Electrical and Electronic Engineering*, 21(1), 19-27.

26. Yousef Ibrahim Daradkeh, Irina Tvoroshenko, Volodymyr Gorokhovatskyi, Liza Abdul Latiff and Norulhusna Ahmad. Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic.

27. Daradkeh, Y. I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L. A., & Ahmad, N. (2021). Development of effective methods for structural image recognition using the principles of data granulation and apparatus of fuzzy logic. *IEEE Access*, 9, 13417-13428.

28. Server Sent Events. URL: <https://uk.javascript.info/server-sent-events> (дата звернення 10.05.2023).

29. MediaPipe: Google's Open Source Framework for ML solutions (2023 Guide). URL: <https://viso.ai/computer-vision/mediapipe/> (дата звернення 12.05.2023).

30. Introduction to MediaPipe. URL: <https://learnopencv.com/introduction-to-mediapipe> (дата звернення 12.05.2023).