КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра теоретичної кібернетики

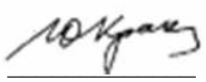
Кваліфікаційна робота

на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:
Дослідження підходів машинного навчання для розпізнавання жестів

Виконав: студент 4-го курсу
Вереновський Назарій Олександрович

(підпис)

Науковий керівник:
професор кафедри теоретичної кібернетики
доктор фіз.-мат. наук,
Крак Юрій Васильович



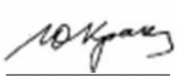
(підпис)

Засвідчую, що в цій роботі
немає запозичень з праць інших авторів
без відповідних посилань.
Студент

(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теоретичної кібернетики
«____» _____ 2022 р.,

протокол № ___
Завідувач кафедри
доктор фіз.-мат. наук, професор
Юрій КРАК



(підпис)

КИЇВ-2022

ЗМІСТ

РЕФЕРАТ 7

ABSTRACT 8

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 9

ВСТУП 10

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ 12

1.1 Аналіз задачі розпізнавання жестів 12

1.1.1 Роль жестів у людино-комп'ютерній взаємодії 12

1.1.2 Способи інтерпретації жестів людини комп'ютером 13

1.1.3 Підходи до задачі розпізнавання жестів 16

1.1.4 Складності при розпізнаванні жестів 19

1.2 Аналіз функціональних можливостей програмних систем розпізнавання жестів 21

1.2.1 DataGlove та Z-Glove 21

1.2.2 EllipticLabs InnerMagic 23

1.2.3 ArcSoft gesture recognition 23

1.3 Постановка задачі 25

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ 26

2.1. Вибір системи комп’ютерного зору 26

2.2. Вибір алгоритмів та засобів для розпізнавання жестів 28

2.2.1. MediaPipe Hands 31

2.3. Розпізнавання жестів на основі ключових координатних точок 34

РОЗДІЛ 3. ОПИС РІШЕНЬ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ 36

3.1 Реалізація загальної структури програмного забезпечення 36

3.2 Розробка основних алгоритмів програмного забезпечення 40

РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ 45
4.1 Оцінка ефективності роботи системи і умов її покращення 45
4.2 Особливості і обмеження розробленої системи розпізнавання жестів 47
РОЗДІЛ 5. ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ДИПЛОМНОЇ РОБОТИ..... 49
5.1 Планування розробки системи розпізнавання статичних жестів..... 50
5.2 Визначення витрат на розробку програми 51
5.3 Розрахунок економічної ефективності програмного продукту..... 58
ВИСНОВКИ..... 59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... 60
ДОДАТОК А. ТЕКСТ ПРОГРАМИ..... 62

РЕФЕРАТ

Об’єкт дослідження – розпізнавання жестів у реальному часі за допомогою комп'ютерного зору.

Мета роботи – розробка та застосування програмного забезпечення з графічним інтерфейсом для розпізнавання жестів у реальному часі за допомогою комп'ютерного зору з використанням цифрової камери.

Основою методу розпізнавання в розробленому програмному забезпеченні є виділення руки на зображенні за допомогою визначення її кольору і знаходження дефектів опуклості контуру руки і його опуклої оболонки.

В процесі розробки було використано мову програмування Python, інтегроване середовище розробки PyCharm Community Edition та бібліотеку комп’ютерного зору OpenCV. В якості камери була використана веб-камера Dell G3 3579, роздільною здатність 1280 на720 пікселів.

Створений програмний продукт дозволяє використовувати веб-камеру для виділення ділянки руки і розпізнавання наступних жестів: Так, Ні (тобто згоден або не згоден), Я кохаю тебе, алвафітні жести, Дякую, Надобраніч, Добрий день.

ABSTRACT

Explanatory note to the master thesis: 74 pages, 26 figures, 6 tables, 1 appendix, 9 sources.

Object of the research – real-time gesture recognition with the use of computer vision.

Purpose of the work – development and usage of software with graphical user interface for real-time gesture recognition by the means of computer vision and a digital camera.

The foundation of recognition method in the developed software is detection of hand region on the input image by using hand color values and finding the convexity defects of the convex hull around the contour of the hand.

During the development of the software,programming language Python was used, along with integrated development environment PyCharm Community Edition and computer vision library OpenCV. The camera for the development was Dell G3 3579 with the resolution of 1280 by 720 pixels.

The developed software provides the possibility for detection of hand region and recognition of such gestures as five to one fingers sticking out.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

РЖ – Розпізнавання жестів

ML – Machine Learning (машинне навчання)

LDA – Linear discriminant analysis (лінійний дискримінантний аналіз)

KNN – K-nearest neighbors (К-найближчих сусідів)

LVQ – Learning vector quantization (мережі векторного квантування)

MAE – mean absolute error (середня абсолютна помилка)

MAPE – mean absolute percentage error (середня абсолютна процентна похибка)

MSE – mean square error (середньоквадратична помилка)

VR – Virtual reality (віртуальна ріальність)

CV – Computer vision (комп'ютерний зір)

GUI – Graphical user interface (графічний інтерфейс користувача)

CNN – Convolutional neural network (згорткова нейронна мережа)

SSD – Single-Shot Detector (один із видів нейронних мереж)

IDE – Integrated development enviroment (середовище для розробки)

API – Application programming interface (прикладний програмний інтерфейс)

ВСТУП

Весь технологічний прогрес людства побудований на відкритті нових технологій і подальшій простоті їх застосування. Кожен аспект нашого життя постійно змінюється, і з'являються найкращі рішення для конкретних завдань і проблем. Інформаційна ера не є винятком. Неможливо уявити собі сучасне суспільство без Інтернету, месенджерів та інших технологій, які спрощують отримання свіжої інформації, спілкування та виконання повсякденних завдань. Однак виникає нове питання: усі ці технології, за деякими винятками, в першу чергу призначені для широкої громадськості.

Наприклад, не всі нові інформаційно-технологічні рішення доступні для людей з вадами, оскільки вони не мають цієї функції.

Звичайно, оскільки алгоритми машинного навчання та нейронні мережі продовжують розвиватися, з'являються нові методи виявлення мови, жестів і руху, але вони реалізуються лише в обмеженій кількості програмних додатків і дозволяють виконувати лише частину функціональних можливостей.

Більшість із нас є абсолютно здоровими та самодостатніми людьми, які ніколи не розглядають проблеми людей з обмеженими можливостями чи людей з обмеженими можливостями, оскільки вони здаються тривіальними, віддаленими та просто неактуальними в контексті нашого повсякденного життя.

Ми звикли бачити, чути, ходити та робити те, що хочемо, але ми не усвідомлюємо, що тисячі людей із фізичними та сенсорними обмеженнями хочуть жити повноцінним життям. Навколишнє середовище недостатнє для задоволення потреб людей з вадами. Це вимагає від таких людей використання різноманітних методів для вирішення труднощів, зокрема комп'ютерів.

Оскільки нещодавно світ усвідомив важливість інформаційних технологій для розвитку суспільства, формування інформаційної культури серед людей з обмеженими можливостями стало особливо важливим.

Акцент у цьому дослідженні робиться на програмному забезпеченні на основі камери, які можуть забезпечити спілкування між людьми. Ці інтерфейси часто

засновані на методах комп’ютерного зору, які записують обличчя або голову користувача, і спеціально створені для людей з обмеженими можливостями.

Актуальність дипломної роботи підтверджується важливістю параметрів таких інтерфейсів, а також літератури з цього питання, щоб включити ці знання в їх систему. Система ідентифікує та класифікує рухи та дії людини, оцінюючи візуальну інформацію, отриману камерою, а потім використовує їх для спілкування. Такі інтерфейси можна використовувати для будь-чого, від комп’ютерних ігор до керування роботами.

Отже, основною **метою даної роботи** є розробка веб-інтерфейсу, який використовує алгоритми розпізнавання образів обличчя та долонь, що допоможе користуватися мовою жестів людям для спілкування між собою.

Для досягнення мети дипломної роботи були поставлені наступні **завдання**:

1) Вивчити відповідну літературу;
2) Проаналізувати діючі готові програмні рішення та алгоритми;
3) Розробити основні алгоритми та їх зв'язки із сторонніми браузерами;
4) Протестувати відповідні модулі;

Об’єктом дипломної роботи є згорткові нейронні мережі для розпізнавання мови жестів.

Предметом дипломної роботи є використання можливостей згорткових нейронних мереж для розпізнавання мови жестів.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз задачі розпізнавання жестів

1.1.1 Роль жестів у людино-комп'ютерній взаємодії

З масовим збільшення кількості комп'ютерів у суспільстві, ЛКВ стає дедалі більш важливою частиною нашого повсякденного життя. Вважається, що з тим як обчислювальні, комунікаційні та відображальні технології прогресують далі, існуючі методи ЛКВ можуть стати перешкодою для ефективного використання наявного інформаційного потоку. Наприклад, найпопулярніший метод ЛКВ заснований на простих механічних пристроях клавіатурі та миші. Ці пристрої стали знайомими та загальноприйнятими, але за своєю суттю обмежують швидкість і природність з якими ми можемо взаємодіяти з комп'ютером.

Ці обмеження стали ще більш очевидними з появою нових технологій відображення, таких як віртуальна реальність. Таким чином, в останні роки відбувся величезний поштовх у дослідженні нових пристроїв і прийомів, які можуть допомогти вирішити проблему обмеженої ЛКВ. Однією з довгострокових спроб вирішення цієї проблеми є перехід до способів взаємодії, які люди використовують між собою, наприклад мова і жести[1].

Розпізнавання жестів (РЖ) - це тема в області комп'ютерних наук та мовних технологій , метою якої є тлумачення жестів людини за допомогою математичних алгоритмів. Жести можуть походити від будь-якого тілесного руху або стану, але зазвичай походять від обличчя або руки.

За межами системи ЛКВ поняття жесту не має чіткого визначення.А якщо визначення і існують, то вони особливо пов'язані з комунікаційним аспектом людських рук і рухів тіла. Однак в області ЛКВ поняття жесту дещо інше.У середовищі, керованому комп'ютером, людська рука використовується для виконання завдань, які імітують одночасно і природне використання руки як

маніпулятор, і його використання в ЛКВ (управління комп'ютером і функції за допомогою жестів). Класичні визначення жестів рідко, якщо коли-небудь, пов'язані з вищенаведеним використанням людської руки в області ЛКВ (так звані практичні жести) [1].

На даний момент галузь фокусується на розпізнавання емоцій з обличчя і на розпізнавання жестів рук. Користувач може використовувати прості жести для керування або взаємодії з пристроями, фізично не торкаючись їх. Багато підходів було зроблено за допомогою камер і алгоритмів комп'ютерного зору для інтерпретації мови жестів. Однак ідентифікація та розпізнавання положення, ходи і поведінки людини також є предметом технік РЖ.

РЖ можна розглядати як спосіб створення можливості розуміння комп'ютером мови тіла людини і взаємодії без будь-яких механічних пристроїв, створюючи тим самим ширший зв'язок між машинами та людьми, ніж примітивні текстові інтерфейси користувача або навіть графічні інтерфейси користувача, які як і раніше обмежують більшу частину вводу до вводу за допомогою клавіатури і комп'ютерної миші. Це може зробити звичайні методи вводу та пристрої навіть зайвими.

Автоматизоване розпізнавання візуальних жестів може служити дуже практичним інструментом у випадках взаємодії з інтелектуальними машинами (наприклад, роботами) або служіння соціальним цілям у випадку розуміння мови жестів.

1.1.2 Способи інтерпретації жестів людини комп'ютером

Для використання жестів для ЛКВ важливо розробити способи їх інтерпретації комп'ютером. Інтерпретація жестів вимагає щоб була можливість для машини вимірювання динамічних або статичних конфігурацій людської руки. Перші спроби вирішити цю проблему призвели до механічних пристроїв, які безпосередньо вимірюють руку, кути згинання суглобів та їх просторове положення. Ця група найкраще представлена так званими пристроями заснованими

на рукавичках. Засновані на рукавичках інтерфейси вимагають від користувача носіння громіздкого пристрою і великої кількості кабелів, які підключають пристрій до комп'ютера. Це перешкоджає легкості та природності, з якою користувач може взаємодіяти з середовищем, керованим комп'ютером. Навіть якщо використання таких специфічних пристроїв може бути виправдане у вузькоспеціалізованих областях застосування, наприклад, моделювання хірургії в умовах віртуальної реальності, звичайний користувач, безперечно, буде обмежений такими громіздкими інструментами інтерфейсу. Це породило активні дослідження в напрямку більш "природних" методів ЛКВ [3].

Потенційно, будь-яка незграбність пов'язана з використанням рукавичок або інших пристроїв може бути подолана використанням безконтактних методів взаємодії. Це підхід означає використання єдиної відеокамери або їх набору, а також методів комп'ютерного зору для РЖ [2].

Здатність відстежувати рухи людини і розпізнавати який жест вона показує може бути досягнута за допомогою різних інструментів. Кінетичний інтерфейс користувача є новим видом інтерфейсів, який дозволяє виконувати ЛКВ за допомогою переміщення об'єктів і тіл. Хоча існує велика кількість досліджень в області РЖ на основі відео та зображень, присутня певна варіація серед використовуваних інструментів і середовищ:

- дротові рукавички можуть надавати вхідні дані для комп'ютера щодо позиції і обертання руки використовуючи магнітні або внутрішні відстежуючи пристрої. Деякі рукавички можуть навіть виявляти згинання пальців з високою точністю до 5-10 градусів, або навіть надавати користувачу зворотній зв'язок, що є відтворенням відчуття дотику. Першим доступним на ринку пристроєм такого типу був DataGlove, він був здатний виявляти положення руки, рухи і згинання пальців. Він використовував оптичні кабелі, що проходили по зворотній стороні долоні. Створювались світові імпульси і коли пальці вигиналися, світло виходило через невеликі тріщини і втрата записувалась, даючи прогнозування положення руки;
- використовуючи спеціальні камери глибини, можливо створити набір значень глибини для вхідних даних камери на невеликій відстані і використовувати

ці дані для побудови тривимірного представлення того, що захопила камера. Такі пристрої можуть бути ефективними для задачі РЖ через їх високу здатність побудови представлення руки на невеликій відстані. На рисунку 1.1 зображено зразок знімку камери глибини;

- стерео камери. Шляхом використання двох камер, відношення яких одна до одної відоме, може бути побудоване тривимірне представлення. Для отримання відношення камер можна використовувати позиційне посилання, наприклад інфрачервоні випромінювачі. В поєднанні з безпосереднім вимірюванням руху можуть бути розпізнані жести;
- контролери, засновані на жестах. Такі пристрої виступають продовженням тіла, і таким чином під час виконання жестів, деякі рухи можуть бути без труднощів записані програмне забезпечення (ПЗ);
- єдина двовимірна камера може бути використана для РЖв умовах коли інші методи не можуть бути застосовані.

Хоча вважається що цей метод може бути не таким ефективним як стереокамери або камери глибини, деякі компанії досягають певних результатів, розробляючи ПЗ , яке може ефективно розпізнавати жести.

Рисунок 1.1 - Приклад знімку камери глибини

В даній роботі в якості інструменту введення даних про положення руки для розпізнавання жесту використовується двовимірна цифрова камера.

При розробці проекту була застосована камера Dell G3 3579, яка є вбудованою веб-камерою.

1.1.3 Підходи до задачі розпізнавання жестів

Деяка література відрізняє два різних підходи до задачі розпізнавання жестів: методи засновані на тривимірних моделях і методи засновані на графічних зображеннях. Перший з приведених підходів використовує тривимірну інформацію про ключові елементи частин тіла з метод отримання деяких важливих параметрів, таких як позиція долоні або кути вигину суглобів. З іншого боку системи засновані на графічних зображеннях використовують картинки або відео для безпосереднього розпізнавання [4].

Підхід з використанням тривимірних моделей може засновуватись на об'ємних або скелетних моделях, або навіть на їх поєднанні. Об'ємні моделі були широко застосовані в комп'ютерній анімації та для цілей комп'ютерного зору. Приклад тривимірної моделі наведено на рисунку 1.2. Недоліком цього методу є велика складність обчислення і системи для роботи в реальному часі все ще знаходяться на стадії розробки. На даний момент більш цікавим підходом є прив'язування простих геометричних форм до точок, що знаходяться на місці найважливіших частин тіла, і аналіз їх взаємодії між собою[4].

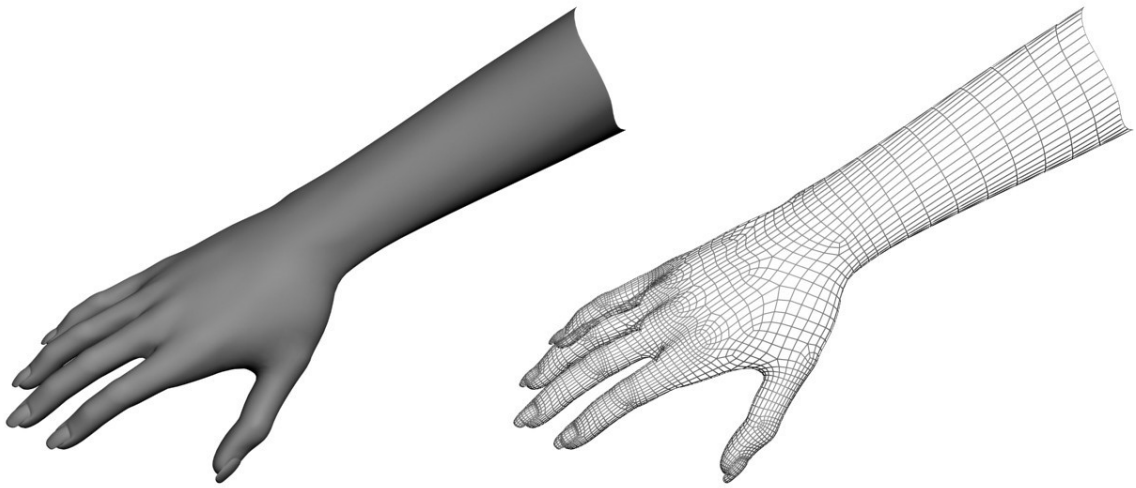


Рисунок 1.2 - Тривимірна модель руки людини

Замість використання складної обробки тривимірних моделей з великою кількістю параметрів, можливе використання спрощеної версії параметрів кута вигину суглоба разом з довжинами сегментів. Це відоме як скелетне зображення тіла, де обчислюється віртуальний скелет людини, а частини тіла прив'язуються до певних сегментів. На рисунку 1.3 зображено приклад побудованої скелетної моделі руки людини. Аналіз тут проводиться з використанням положення та орієнтації цих сегментів та співвідношення між кожним з них (наприклад, кут між суглобами та відносне положення або орієнтація).

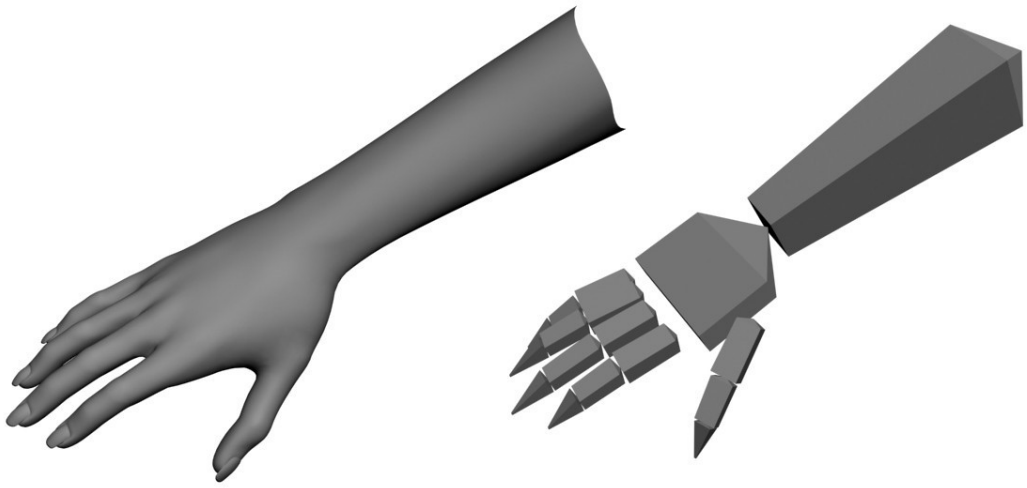


Рисунок 1.3 - Скелетна модель руки людини

Використання скелетних моделей має наступні переваги:

- алгоритми виконуються швидше, оскільки аналізуються лише основні параметри;
- можливе зіставлення шаблонів із базою даних шаблонів;
- використання ключових точок дозволяє програмному забезпеченню для розпізнавання зосередитися на важливих частинах тіла.

Моделі на основі зображень не використовують просторове представлення тіла, оскільки вони отримують параметри безпосередньо з зображень або відеозаписів за допомогою бази даних шаблонів. Деякі з них базуються на деформованих двовимірних шаблонах людських частин тіла, особливо рук.

Деформативні шаблони - це набори точок на контурі об'єкта, які використовуються як вузли інтерполяції для наближення об'єкта. Одна з найпростіших функцій інтерполяції є лінійною, яка будує середню форму від наборів точок, параметрів зміни точок та зовнішніх деформаторів. Ці моделі, засновані на шаблонах, в основному використовуються для відстеження руки, але також можуть бути корисними для простої класифікації жестів [4].

Другий підхід у визначенні жесту за допомогою моделей на основі зображення використовує послідовності зображень як шаблони жестів.

Параметри для цього методу - це самі зображення або певні функції, отримані з них. Найчастіше використовуються лише одна (моноскопічний) або дві (стереоскопічні) точки спостереження. У даному дипломному проекті був використано підхід побудови моделі на основі зображень. Після того, як регіон зображення що відповідає руці був виділений, виділяється його контур і подальші розрахунки виконуються з його допомогою. На рисунку 1.4 зліва зображено силует руки, що був отриманий після знаходження регіону, що відповідає руці, а поруч його контур що використовується при розрахунках. Програмно контур представляє собою множину точок, з яких відповідний контур складається

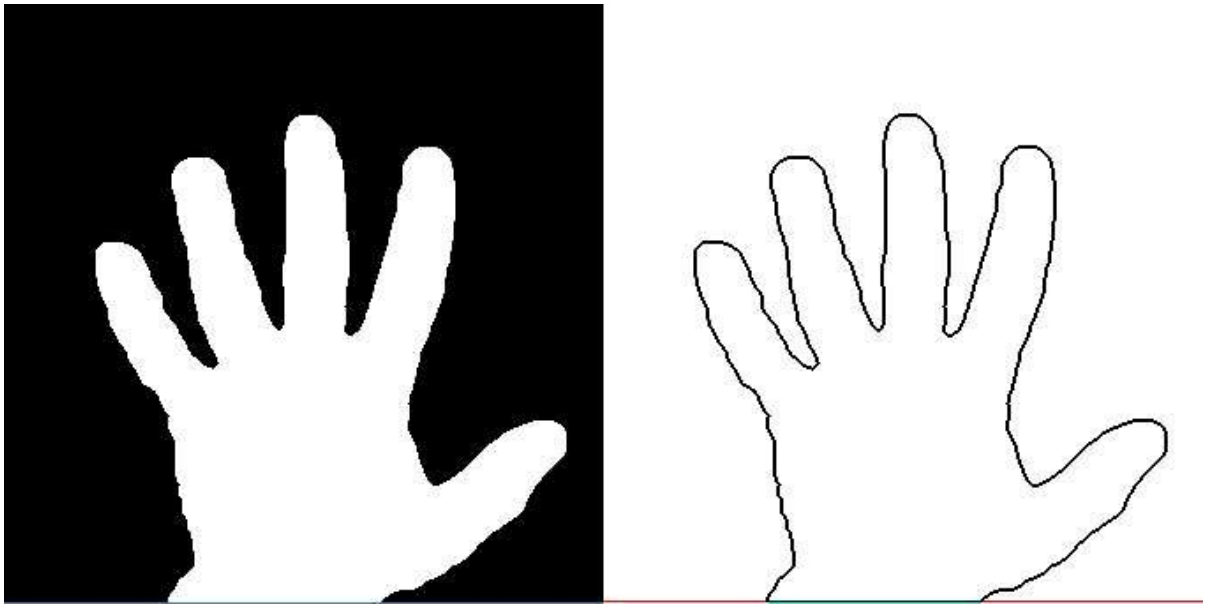


Рисунок 1.4 - Виділений регіон руки і його контур як вхідні дані для розпізнавання

1.1.4 Складності при розпізнаванні жестів

Є багато складностей, пов'язаних з точністю та ефективністю програмного забезпечення для РЖ. Для РЖ на основі зображення існують обмеження пов'язані з використанням обладнання, яке не передає зображення ідеальної якості, та візуальний шум на зображенні. Зображення або відеозапис може відбуватись під не постійним освітленням, або лише в одному місці. Фонові елементи або певні особливості користувачів можуть ускладнити розпізнавання. Саме тому в даному проекті для ефективної роботи необхідно забезпечити задовільні умови освітлення, тобто бажано розміщувати руку перед однорідним фоном і розташовувати освітлення на одній лінії з камерою і рукою, тобто або за камерою, або за рукою. Певна варіація кута допустима і не приносить значне зниження ефективності виділення контуру руки. Наприклад, розроблене ПЗ вдало виділяє регіон руки при умовах помірного сонячного світла вранці або в день на фоні звичайної білої стелі.

Різноманітність реалізацій програмного забезпечення для РЖ на основі зображення також може спричинити проблему для життєздатності технології для загального користування. Наприклад, алгоритм, відкалібрований для однієї камери, може не працювати для іншої камери. Кількість фонових шумів також спричиняє проблеми з відстеженням та розпізнаванням. Крім того, відстань від камери, а також роздільна здатність і якість фотокамери також викликають відмінності в точності розпізнавання.

Алгоритм розпізнавання, реалізований в даному проекті, був успішно випробуваний на камері початкового сегменту ринку - LogitechC170, і в відношенні ціни і в відношенні якості. LogitechC170 має роздільну здатність 640 на 480 пікселів та максимальний показник 30 кадрів у секунду.Це далеко не межа показників для веб-камер. Очевидно, що при використанні більш якісної камери, результат розпізнавання буде ще кращий.

1.1.5 Моделі жестів

Першим етапом завдання розпізнавання є вибір моделі жесту. Математична модель може виражати як і просторові, так і часові характеристики руки і жесту. Підхід, обраний для моделювання, відіграє ключову роль у природі і ефективності процесу РЖ. Після того, як буде прийнято модель, виконується перехід до етапу аналізу для обчислення параметрів моделі із особливостей зображення, які отримуються із одного або кількох відеопотоків. Ці параметри складають деякий опис положення руки або її траєкторії і залежать від обраного підходу моделювання. Серед важливих проблем етапу аналізу є локалізація руки, відстеження руки і вибір відповідних до моделі особливостей зображення. Після обчисленням параметрів моделі виконується розпізнавання жестів. На цьому етапі параметри класифікуються іінтерпретується у відповідністю з прийнятою моделлю.

Щоб повною мірою використати потенціал жестів у середовищах ЛКВ, клас доступних до розпізнавання жестів повинен бути якомога ширшим. В ідеалі будь-який жест, який виконується користувачем, повинен бути однозначно інтерпретованим, таким чином дозволяючи здобути природність інтерфейсу. Однак, навіть передові системи на основі комп'ютерного зору для розпізнавання жестів не забезпечують задовільного результату в досягненні цієї мети. Більшість систем ЛКВ на основі жестів в даний час розглядають дуже вузьку групу можливих застосувань: переважно символічні командина підставі положення руки, як і в даному проекті. Причиною цього є складність, пов'язана з аналізом та розпізнаванням жестів. Прості моделі жестів роблять можливим створення інтерфейсів управління за допомогою жестів в реальному часі. Прикладом такого інтерфейсу можна вважати і ПЗ розроблене в даному проекті [2].

Взаємодія в реальному часі, яка спирається на тривимірні моделі руки складніші в розробці через обчислювальну складність. Такі моделі зазвичай використовуються лише для відстеження руки і аналізу її положення. В той же час,

аналіз тривимірної моделі може в результаті надати можливість розпізнавати ширший набір жестів, ніж аналіз, заснований на зображенні.

1.2 Аналіз функціональних можливостей програмних систем розпізнавання жестів

Розглядаючи класифікацію систем РЖ за типом моделі, яка була наведена у попередньому підрозділі, можна виділити два основні типи систем РЖ:

- системи, що будують тривимірну модель на основі стереокамери, камери глибини або пристроїв, які є продовженням руки, наприклад спеціальних рукавичок;
- системи, засновані на зображеннях з однієї або декількох відеокамер.

Система, розроблена в даному дипломному проекті відноситься до другого типу.

Основною перевагою систем, заснованих на зображеннях, є швидкість обчислень і можливість ефективної роботи в реальному часі у зв'язку з цим.

Основною перевагою систем, заснованих на тривимірних моделях, є можливість розробити ширший перелік жестів, що підлягають розпізнаванню.

Для створення контексту для порівняння і опису розробленого програмного забезпечення слід навести короткий опис вже існуючих аналогів.

1.2.1 DataGlove та Z-Glove

Інструменти введення жестів для рук представлені тут, Z-Glove та DataGlove - це легкі бавовняні рукавички, що містять датчики згинання, які вимірюють вигин пальців, системи позиціонування та орієнтації, і тактильний зворотній відгук за допомогою вібрації.Різні системи орієнтації та позиціонування Z-Glove і DataGlove розрізняють ці дві моделі. У Z-Glove використовується ультразвукова система орієнтації та позиціонування. DateGlove використовує магнітну систему позиціонування та орієнтації [5].

До переліку можливих застосувань DataGlove і Z-Glove Відносяться розпізнавання жестів, клінічний інструмент для оцінки ручної функції, тривимірний контролер моделі руки, інтерфейс для візуальної мови програмування, контролер синтезу музики та звуку та маніпулятор комп'ютерних об'єктів. Майбутнє Застосування для додатків DataGlove та ZGlove Полягає в робототехніці, та дослідженні людських факторів та ергономіки.

Тактильний зворотній зв'язок використовується для додавання реалізму до взаємодії між комп'ютерними (віртуальними) об'єктами та віртуальною рукою. Тактильні сигнали використовуються для імітації контакту з об'єктом, твердості і поверхневої текстури. П'єзокерамічні пристрої для згинання, які керують синусоїдним сигналом частотою 20-40 Гц, (нижче пікової чутливості 250Гц для запобігання утворення звуку), розташовуються під кожним пальцем. В результаті їх роботи користувач отримує відчуття поколювання або оніміння. На рисунку 1.5 зображено DataGlove зі знятою зовнішньою рукавичкою під час використання для маніпуляції віртуальними об'єктами [5].

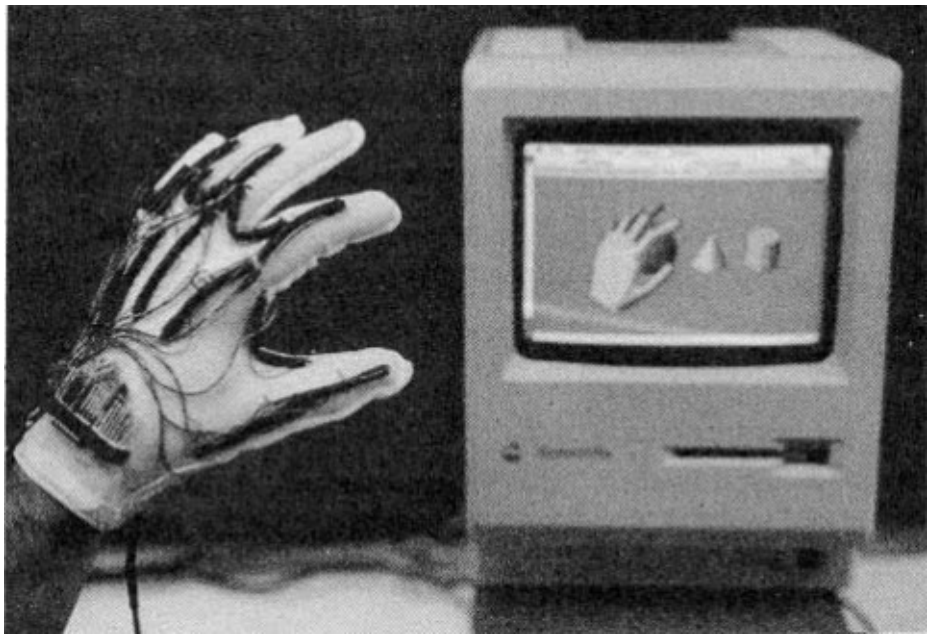


Рисунок 1.5 – DataGlove зі знятою зовнішньою рукавичкою

1.2.2 EllipticLabs InnerMagic

InnerMagic- це запатентована лінійка продуктів для ультразвукових жестів без дотику. InnerMagic складається з двигунів жестів та віртуальних датчиків жесту, що забезпечують просту, інтуїтивну взаємодію, якої прагне користувачі. Всі датчики InnerMagic мають поле зору 180 градусів, що дозволяє користувачам виконувати природні ручні рухи в повітрі над пристроєм, під пристроєм, або в бік від нього [8].

MagicSNAP - це один із віртуальних сенсорів жестів, доступний від InnerMagic. Це ПЗ для смартфону, яке дозволяє користувачам робити фотографії за допомогою жесту на відстані до 6 метрів. В останніх дослідженнях споживачів понад 90% користувачів мали позитивну оцінку можливості використання MagicSNAP на власному пристрої.

Дана система базується на відтворенні динаміком пристрою, на якому працює InnerMagic, ультразвукових хвиль, які відлітають від частини тіла користувача, яка розглядається, і повертаються до пристрою. Відбита хвиля вловлюється мікрофоном пристрою і інтерпретується віртуальними сенсорами.

Перевагами такої системи є робоча дистанція, до 5 метрів, низьке використання енергії батареї пристрою, на якому працює система, широке поле зору та можливість роботи в будь-яких умовах освітлення.

Підхід, використаний у цьому проекті є досить унікальним і має перевагу з боку швидкості роботи, але має значні обмеження з точки зору різноманіття і кількості можливих розпізнаних жестів.

1.2.3 ArcSoft gesture recognition

З моменту свого заснування ArcSoft зосереджує увагу на технологіях та програмному забезпеченні, пов'язаному із зображеннями та комп'ютерним зором на основних пристроях та платформах, таких як смартфони, камери віртуальної

реальності, камери дронів, IP-камери, розумні телевізори, роботи, інтелектуальні холодильники, інтелектуальні пилососи, автомобільні комп'ютери та планшети [9].

Даний продукт використовує одну або більше камер.

Користувачі можуть використовувати прості жести для керування або взаємодії з пристроями, фізично не торкаючись їх. Технологія ArcSoft gesture підтримує пристрої з одним об'єктивом, включаючи HD-камери, а також декілька пристроїв із об'єктивами та камери глибини. Завдяки використанню глибинних камер або пристроїв з декількома об'єктивами, глибина може бути проаналізована, щоб більш точно виявляти та підтримувати ще більший діапазон рухів рук.

Технологія ручного жесту полегшує життя користувачів, забезпечуючи взаємодію без дотику, усуваючи необхідність тримати або натискати пристрій. Застосування для технології ArcSoft gesture recognition включають прокручування або переміщення між екранами на планшетному пристрої за допомогою простих жестів рукою та змінення каналу або регулювання гучності на телевізорі без використання пульта дистанційного керування. Користувачі навіть можуть використовувати жести для навігації під час перегляду зображень або відео на своїх комп'ютерах або на великих екранах.

Багато років знадобилося для проведення дослідження природних рухів людської руки, щоб удосконалити технологію ArcSoft gesture recognition. Для того, щоб створити більш зручні умови для використання системи, було зібрано та проаналізовано великий обсяг даних, щоб можна було точно визначити звичайні рухи рук в недостатньо оптимальних або поганих умовах, таких як низький рівень освітлення, великі відстані, складні умови освітлення, тощо.

Завдяки використанню глибинних камер, технологія РЖ ArcSoft 3D дозволяє не тільки досягти тривимірного відображення людської руки, але і розпізнавати і відстежувати жести. З точною інформацією про глибину, наданою цією технологією, користувачі можуть більш природним чином взаємодіяти з комп'ютерами. Така тривимірна технологія використовується для природного і звичного для людини управління телевізором та іншими розумними домашніми

пристроями, а також забезпечує природну взаємодію з пристроями віртуальної реальності та доповненої реальності.

1.3 Постановка задачі

Задача даної роботи полягає у реалізації програмного забезпечення з графічним інтерфейсом користувача, функцією якого є зчитування відеопотоку з веб-камери, і розпізнавання жестів людини на зображеннях з кадрів. Наступні жести повинні підлягати розпізнаванню: виставлені пальці від одного до п'яти, кулак і відкрита долоня, опціональні жести: Я кохаю тебе, Добрий день, Дякую.

Програмне забезпечення повинно мати можливість розпізнавання як алфавітних символів, так і повноцінних жестів.

ПЗ повинно розроблятися з можливістю працювати з підключеною до персонального комп'ютера веб-камерою і в умовах не ідеального освітлення та фону. Калібрування координатих точок ініціююватиметься діями користувача під час тренування згорткової нейронної мережі.

Розпізнавання повинно відбуватися в реальному часі і з достатньою швидкістю для відтворення результату аналізу жестів.

Графічна оболонка програми повинна надавати можливість користувачу налаштовувати відтворювані жести, режими роботи (алфавітне або розпізнавання повноцінних жестів), а також відтворювати точки для аналізу обличчя та рук.

РОЗДІЛ 2. ПРОЕКТУВАННЯ СИСТЕМИ

Ця частина розглядає вибір та обґрунтування методів впровадження програмного продукту.

Нижче наведено основні моменти, які будуть розглянуті в цьому розділі:

• Вибір системи комп'ютерного зору, яка використовуватиметься для зчитування необхідних зображень та їх передачі на вхід модуля аналізу зображень;

• Вибір алгоритму аналізу отриманих даних та подальшої класифікації об'єктів;

• Проектування модуля, який буде виконувати певні дії з інтерфейсом браузера на основі класифікованих даних;

2.1. Вибір системи комп’ютерного зору

OpenCV пропонує багато переваг перед своїми конкурентами, коли справа доходить до бібліотек з відкритим кодом, які надають інструменти для роботи з вхідними зображеннями.

Вона розроблена на мовах програмування C/C ++ високого рівня, що позитивно впливає на продуктивність, і має модульну структуру, з певними модулями, присвяченими роботі з фотографіями, відео, калібруванням камери, виявленням об'єктів та іншими допоміжними функціями [7].

Ця бібліотека містить сотні алгоритмів комп’ютерного зору і часто використовується в мовах програмування, таких як Java, C++/C і Python. При створенні OpenCV були встановлені наступні основні цілі:

• Ефективність комп’ютерних обчислень;

• Робота з системами реального часу.

В результаті ця бібліотека була налаштована для багатопоточних ЦП, що дало їй значне підвищення продуктивності в аналізі.

Ця бібліотека також має перевагу базової та простої функціональності, яка дозволяє легко виконувати завдання [8].

На рисунку 2.1 показана блок-схема основних компонентів OpenCV.

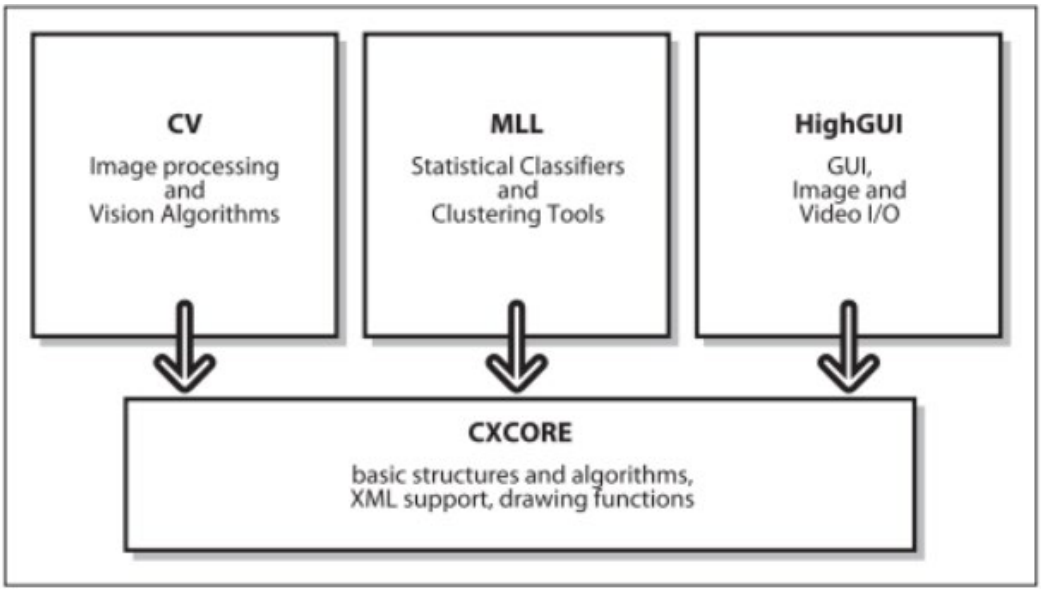


Рисунок 2.1. – Структура модулів OpenCV

Модуль CvAux, крім п’яти основних компонентів, надає експериментальні та нереалізовані інструменти [8].

Також варто відзначити, що ця бібліотека є портативною, оскільки вона працює на різних системах і архітектурах процесора (рис. 2.2).

	IA32	EM64T	IA64	Other (PPC, Sparc)
Windows	✓ (w. IPP; MSVC6, .NET2005+OMP, ICC, GCC, BCC)	✓ (w. IPP; MSVC6+PSDK.NET2005+OMP, PSDK)	± (w. IPP; PSDK, some tests fail)	N/A
Linux	✓ (w. IPP; GCC, BCC)	✓ (w. IPP; GCC, BCC)	✓ (GCC, ICC)	✗
MacOSX	✓ (w. IPP, GCC, native APIs)	? (not tested)	N/A	✓ (iMac G5, GCC, native APIs)
Others (BSD, Solaris…)	✗	✗	✗	Reported to build on UltraSparc Solaris

Рисунок 2.2 – Схема портативності для різних операційних систем та архітектур для бібліотеки OpenCV

2.2. Вибір алгоритмів та засобів для розпізнавання жестів

Згорткові нейронні мережі, які надзвичайно ідеальні для аналізу зображень завдяки своїй структурі (рис. 2.3), використовуються для вирішення таких питань, як категоризація об’єктів та інші проблеми, пов’язані з комп’ютерним зором.

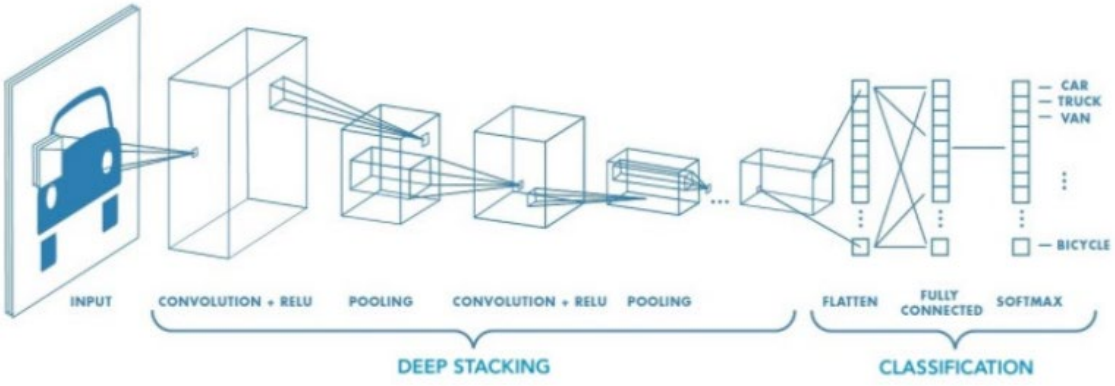


Рисунок 2.3 – Схема шарів згорткової нейронної мережі

Основна відмінність цієї мережі від інших полягає в тому, що вона може обробляти більш складні зображення, які менш залежать від якості, центрування та інших змінних. Традиційні нейронні мережі сильні в таких завданнях, як ідентифікація чисел, але не так добре в категоризації об’єктів. Існування згорткових шарів, які використовуються для фільтрації [9], є основною причиною цієї невідповідності. Спочатку проаналізуємо, як згортковий шар оцінює рівень, не використовуючи порівняння мозок/нейрон. Параметри цього шару складаються з ряду настроюваних фільтрів. Ширина і висота кожного такого фільтра крихітні, але вони покривають всю глибину забірного об’єму.

Наприклад, звичайний фільтр на першому шарі CNN може бути 5x5x3 (тобто 5 пікселів в ширину і 3 пікселі в глибину, оскільки зображення має 3 кольорові канали). Відбувається стискання фільтрів на ширину та висоту вхідного об’єму під час прямого проходу та обчислюємо локальні результати між записами фільтра та вхідними в будь-якому місці під час прямого проходу.

Після того, як буде використано переміщення фільтру по ширині та висоті вхідного об’єму, в результаті отримаємо двовимірну карту активації з результатами проходження фільтра в кожній просторовій точці. Нейронна мережа досліджуватиме фільтри, які задіюються, коли вони оцінюють певну форму візуальної характеристики, наприклад межі певного елемента або колірну пляму на першому шарі, і повні шаблони та шаблони на наступних шарах.

Кожен рівень згорткового шару тепер матиме власний набір фільтрів, кожен з яких генеруватиме різну двовимірну карту активації. Ці карти активації будуть намальовані вздовж вимірювання глибини, щоб отримати початковий об’єм.Тобто, основна функція згорткового шару полягає в тому, щоб представити крихітну колекцію пікселів, використовуючи лише один піксель.

Цей метод зменшує кількість вхідних даних, які згодом надсилаються до стандартної нейронної мережі, якій можуть не знадобитися всі пікселі, які були там до згортання [9].

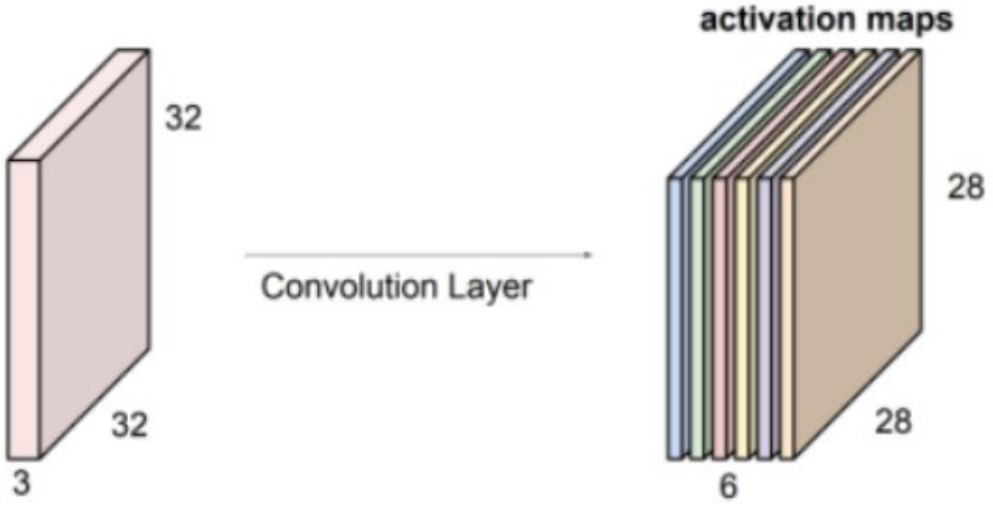


Рисунок 2.4 – Схема, яка демонструє результат після проходження згорткового шару нейронної мережі

В результаті отримаємо картки активації після проходження шару згортки, який необхідно нормалізувати за допомогою функції активації:

$$ReLU(x) = \max(x, 0), \tag{2.1}$$

формальною метою якої є видалення всіх негативних значень з кожного пікселя, тобто просто прирівнювання їх до нуля.

Основні конструкції, що використовуються в згорткових нейронних мережах [10; 11] такі:

• R-CNN - перша і найповільніша модель для виконання операцій класифікації зображень на основі окремих регіонів;

• Хоча Fast R-CNN і Faster R-CNN є кращими версіями попередньої моделі, вони погано підходять для робочих навантажень у реальному часі.

• YOLO - одна з найшвидших різновидів, яка, на відміну від інших, покладається на визначення місця розташування предмета за один прохід;

• SSD – як і попередній, цей заснований на одному проході, але використовує концепцію VGG16 [12]. Містить у своїй основі найкращі моменти YOLO, а також додаткові оновлення.

Зваживши переваги та недоліки кількох методів реалізації згорткової мережі, в результаті було вирішено побудувати систему розпізнавання на SSD. При більш детальному розгляді цієї архітектури можна описати наступні аспекти:

• SSD працює гірше на невеликих об'єктах, ніж Fast RCNN. Зрештою, їх можна розпізнати лише на перших шарах, де роздільна здатність вхідних даних залишається високою. Однак складність полягає в тому, що на цьому етапі можна класифікувати лише ознаки низького рівня, такі як кути, краї та кольорові плями, які дають мало інформації для ідентифікації.

• Точність зростає прямо пропорційно кількості граничних регіонів, хоча й повільніше.

• Конструкція фільтра має значний вплив на підвищення точності.

• SSD має менше помилок локалізації, ніж R-CNN, але з тими ж вхідними даними він може мати більше помилок класифікації.

Для більшої точності SSD можна навчати через. З точки зору розташування, розміру та співвідношення сторін, SSD дає більше прогнозів і має вищий ступінь покриття. Завдяки вищезазначеним удосконаленням SSD може знизити роздільну здатність вхідного зображення до 300 x 300 без шкоди для точності.

Модель може працювати в системах реального часу і виконувати завдання класифікації та розпізнавання значно швидше, ніж найсучасніший Fast R-CNN після видалення призначеної області та використання зображень з нижчою роздільною здатністю.

2.2.1. MediaPipe Hands

Оскільки рішення було прийнято на основі використання твердотільних накопичувачів, найкращою альтернативою є застосування нової платформи MediaPipe від Google, яка була випущена в 2020 році.

Ця структура включає набір інструментів для вирішення різноманітних проблем категоризації та ідентифікації об’єктів [13; 14]:

• обличчя та його контурів; • зіниць очей; • рук; • позиції тіла; • волосся; • об’єктів.

Використаємо інструментарій платформи MediaPipe Hands для досягнення мети, зазначеної в цій бакалаврській роботі, який відповідає за визначення рук на вхідному зображенні.

Це рішення не обмежується роботою з системами реального часу; воно також дозволяє працювати з мобільним телефоном і розпізнавати кілька рук.

Спочатку виявляється долоня, потім визначаються ключові точки (keypoints).

Тобто на першому кроці вибирається область вхідного зображення, де логікою нейронної мережі є рука, а потім ця область передається для аналізу для встановлення основних ознак долоні.

Оскільки на першому етапі область досить велика, а на другому етапі складність зростає через необхідну точність, ми надсилаємо ретельно обрізане зображення руки, що значно зводить до мінімуму необхідність збільшення введення, ця техніка економить час та ресурси.

Нейронна мережа надає 21 критичну координату в тривимірному просторі після завершення другого кроку, як показано на рисунку 2.5.

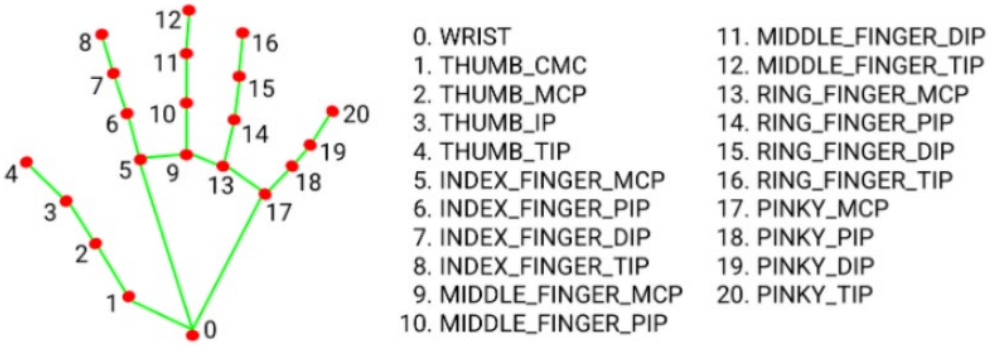


Рис. 2.5 – Екстракція ключових точок в області долоні

Окремі моделі для компонентів стійки, обличчя та руки інтегровані в конвеєр MediaPipe Holistic, кожна з яких оптимізована для власного домену. Однак вхідні дані для одного компонента не дуже підходять для інших через їхню особливість. Наприклад, модель оцінки пози використовує менший відеокадр із фіксованою роздільною здатністю (256x256) як вхідні дані. Якість зображення була б надто низькою для належної артикуляції, якби ділянки кисті та обличчя цього зображення були обрізані, щоб передати їх відповідним моделям. В результаті був створений MediaPipe Holistic як багатоетапний конвеєр, який обробляє кожну зону з оптимальною роздільною здатністю зображення для цього місця (рисунок 2.6).

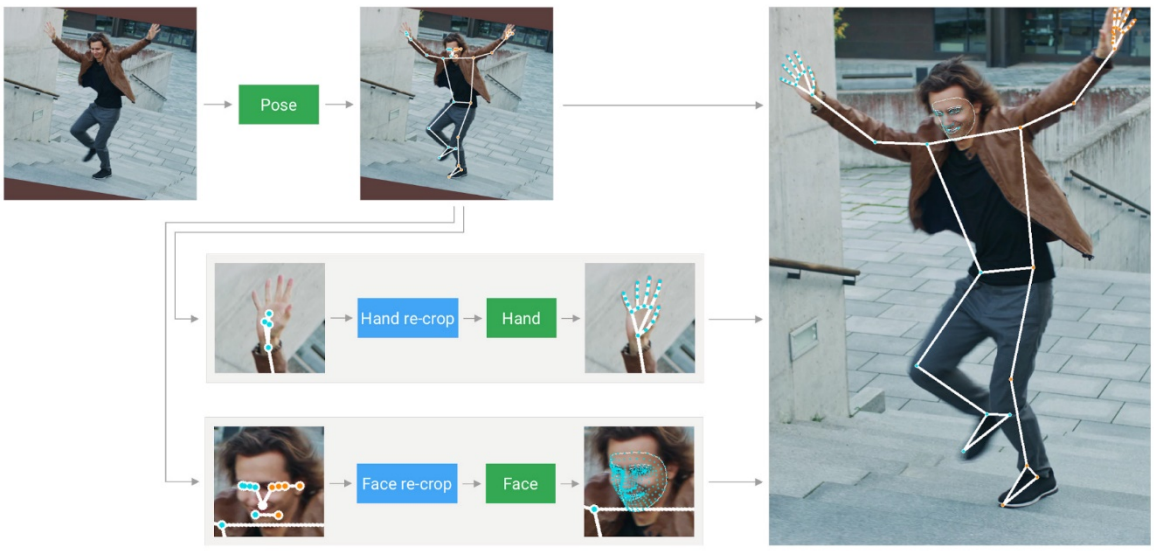


Рис. 2.6 - Огляд цілісного конвеєра MediaPipe.

Спочатку ми використовуємо детектор пози BlazePose і наступну модель орієнтиру, щоб оцінити позу людини (рисунок 2.6). Потім, спираючись на орієнтири пози, ми створюємо три кадрування рентабельності інвестицій для кожної руки (2x) і обличчя та використовуємо модель повторного обрізання для підвищення рентабельності інвестицій. ROI потім обрізаються з вхідного зображення з повною роздільною здатністю, а моделі обличчя та рук для певного завдання використовуються для оцінки пов’язаних з ними орієнтирів.

Нарешті, ми об’єднуємо всі орієнтири з орієнтирами моделі пози, щоб отримати цілих 540+ орієнтирів (рисунок 2.7).

Рисунок 2.7 – Приклад розпізнавання поз за допомогою платформи MediaPipe

При реалізації використаємо метод обстеження, подібний до того, який ми використовуємо для автономних конвеєрів обличчя та рук, щоб спростити ідентифікацію ROI для обличчя та рук. Він передбачає, що об’єкт не переміщується значно між кадрами, і використовує оцінку попереднього кадру як посилання на область об’єкта поточного кадру. Однак трекер може втратити ціль під час швидких рухів, що вимагатиме від детектора повторної локалізації її на зображенні.

Коли реагує на швидкі рухи, MediaPipe Holistic попередньо використовує передбачення положення (на кожному кадрі) як додаткову рентабельність інвестицій, щоб мінімізувати час реакції конвеєра. Це також допомагає моделі підтримувати семантичну узгодженість у всьому тілі та його компонентах, запобігаючи змішування лівої і правої рук або частин тіла однієї людини в кадрі з руками іншого.

Крім того, вхідна рамка моделі пози має досить низьку роздільну здатність, тому генеровані ROI для обличчя та рук все ще занадто неправильні, щоб керувати повторним кадруванням тих ділянок, які потребують точного вхідного кадрування, щоб залишатися легкими.

При реалізації застосовані легкі моделі повторного обрізання обличчя та рук, які діють як просторові трансформатори та коштують лише 10% часу висновку еквівалентної моделі, щоб усунути цей розрив у точності.

2.3. Розпізнавання жестів на основі ключових координатних точок

Отримавши набір координат у тривимірній системі координат, ми можемо використовувати їх для оцінки розташування долоні та окремих пальців, які згодом можна перевести в подальші інструкції.

Для категоризації жестів слід застосовувати аналітичну геометрію або формули, що стосуються розташування векторів у просторі. У нас повинно бути достатньо формул, щоб ідеально визначити довжину вектора:

$$AB = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2}, \tag{2.2}$$

де (x, y, z) з відповідними індексами – це координати точок A і B.

У цьому випадку, порівнюючи довжину векторів від верхньої фаланги до основи долоні або пальця, ми можемо визначити, який з пальців зігнутий, а який ні. Бажано використовувати пропорційне з’єднання, щоб уникнути помилок.

Наприклад, якщо відношення довжин векторів вказівного і середнього пальців більше 1,5, середній палець зігнутий.

Основними командами, які передбачені для виконання є:

• Я кохаю тебе; • Дякую; • Так; • Ні; • Добрий день;• Надобраніч; • Алфавітні послідовності;

РОЗДІЛ 3. ОПИС РІШЕНЬ ЩОДО РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

3.1 Реалізація загальної структури програмного забезпечення

Розроблене ПЗ повинно розпізнавати жести на основі кольорового відеопотоку.

Створений програмний продукт дозволяє використовувати веб-камеру для виділення ділянки руки і розпізнавання наступних жестів: Так, Ні (тобто згоден або не згоден), Я кохаю тебе, алвафітні жести, Дякую, Надобраніч, Добрий день.

Після підключення до відеопотоку, кадри конвертуються з простору кольорів RGB у простір кольорів BGR, через те що, з точки зору числових значень кольору пікселів, у форматі BGR різниця між кольором шкіри і навколишнього середовища значно відрізняється.

Для цього використовується функція бібліотеки OpenCV cv2.cvtColor з другим параметром cv2.COLOR_BGR2RGB.

На рисунку 3.1 зображено вхідне, необроблене зображення в просторі кольорів RGB.

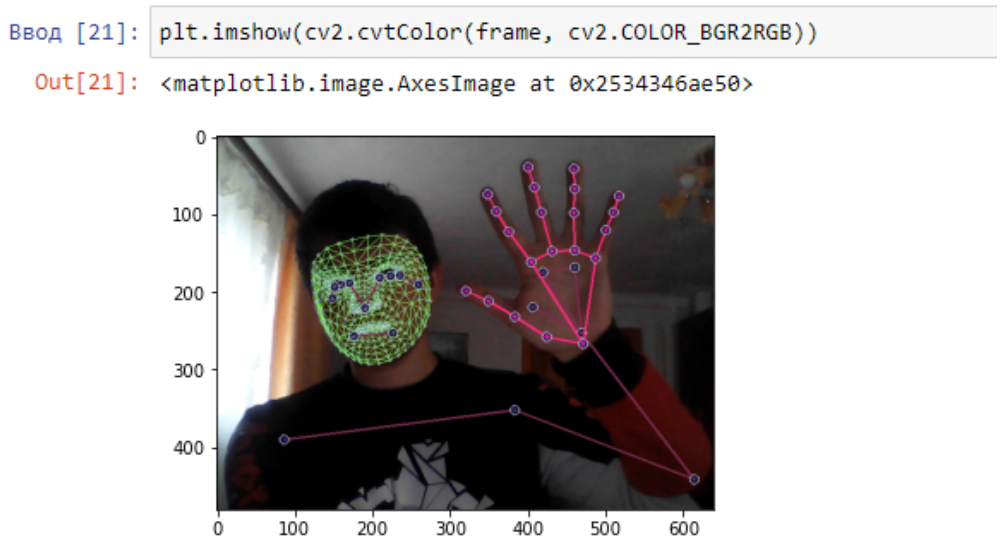


Рисунок 3.1 - Необроблене зображення у просторі кольорів RGB

В результаті його переведення в простір кольорів BGR за допомогою інструментів відкритої бібліотеки комп'ютерного зору OpenCV отримане зображення краще підходить для знаходження регіону, що відповідає руці.

На рисунку 3.2 зображено переведене у простір кольорів BGR зображення.

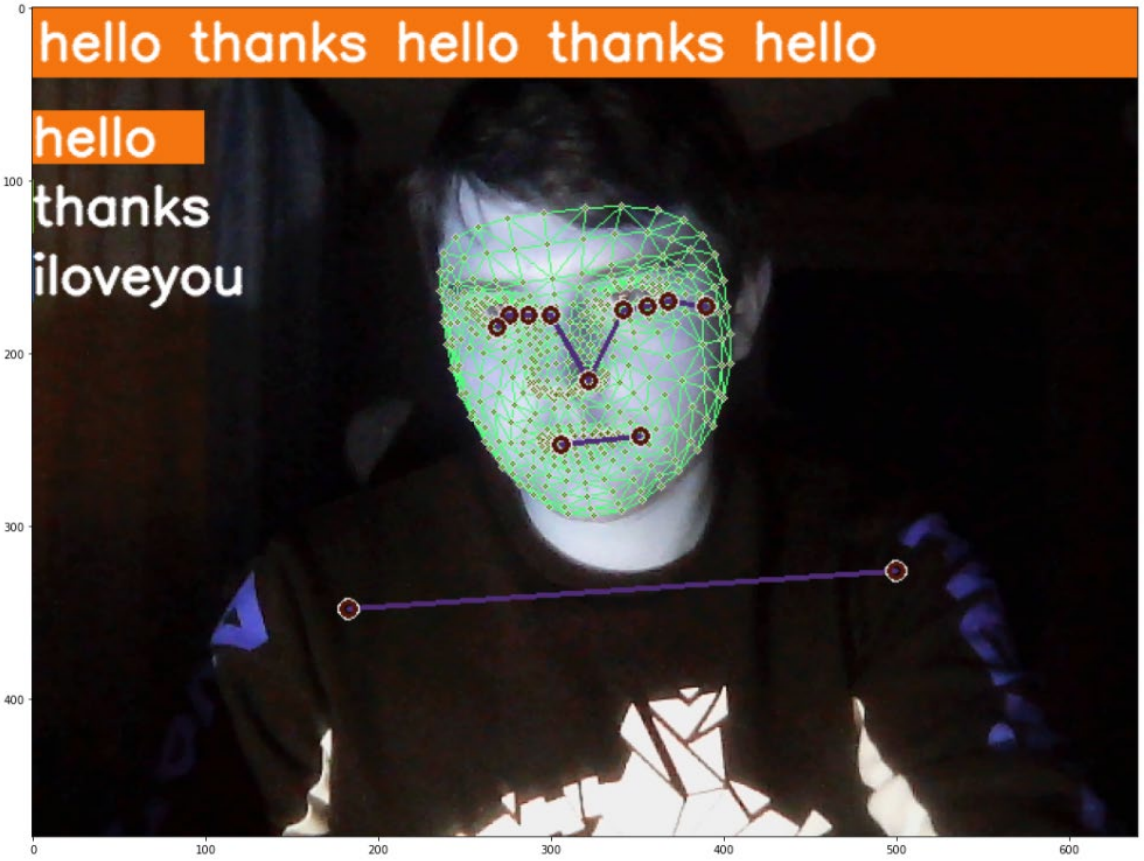


Рисунок 3.2 - Зображення, переведене у простір кольорів BGR

Калібрування ініціюється користувачем вручну і відбувається в реальному часі. Після вдалого калібрування, зображення у просторі кольорів HSV перетворюється на бінарне зображення силуету руки (рис.3.3).

На ньому білим кольором виділяється рука, а чорним - фоновий простір. Саме воно складає вхідні дані для алгоритму розпізнавання.

Рисунок 3.3 - Бінарне зображення з виділеним регіоном руки

Бінарне зображення слугує основою для знаходження контуру руки і ОО навколо нього.

Незалежно від роздільної здатності камери, яка використовується, оброблюваний регіон має розмір 340 на 340 пікселів. Він виділяється зеленою рамкою і знаходиться в лівому верхньому куті зображення.

На рисунку 3.4 зображено кадр відеопотоку з камери LogitechC170, яка має роздільну здатність 640 на 480 пікселів, і виділений зеленим кольором оброблюваний регіон 340 на 340 пікселів.

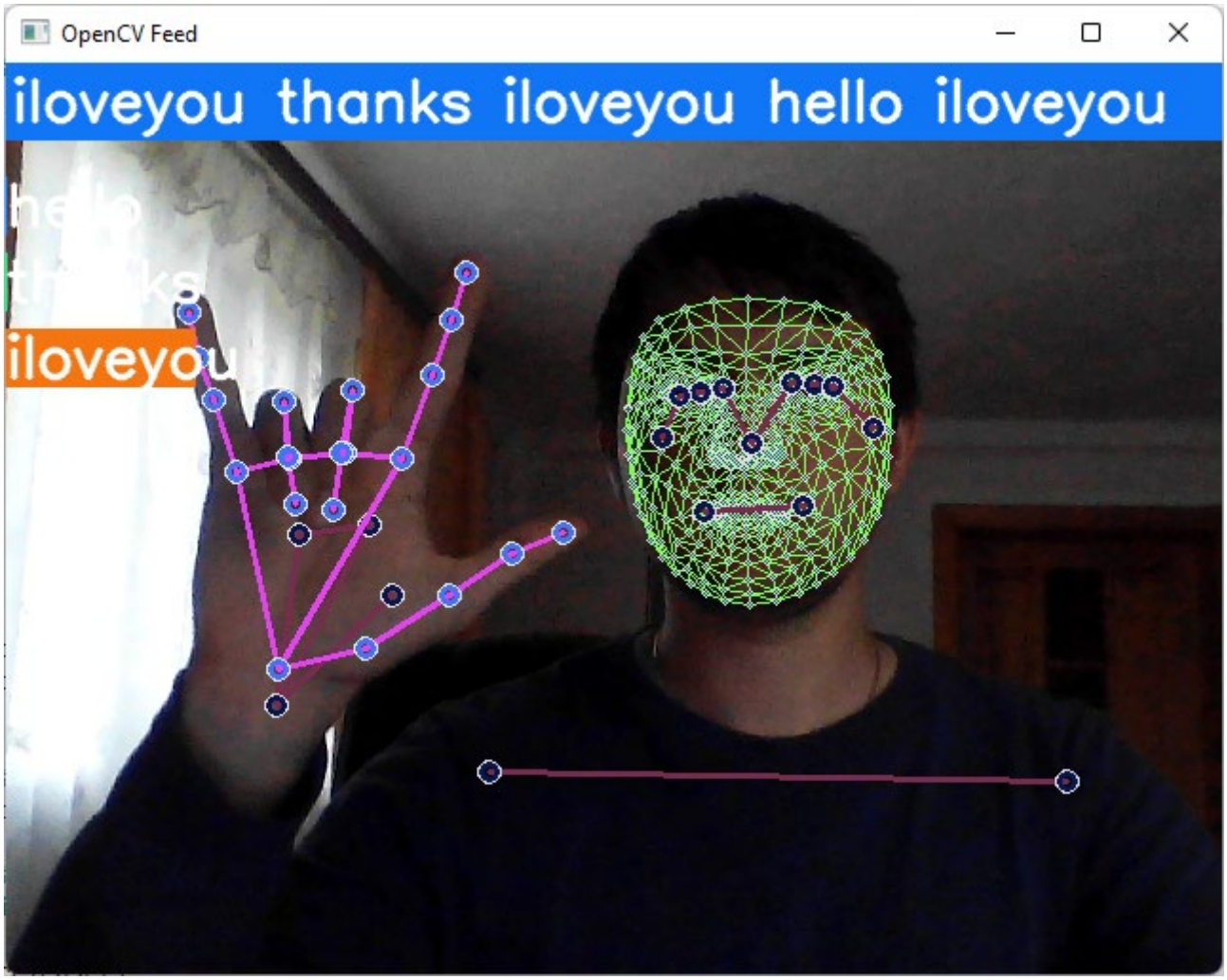


Рисунок 3.4 - Кадр з камери LogitechC170, яка має роздільну здатність 640 на 480 пікселів, і виділений зеленим кольором оброблюваний регіон 340 на 340 пікселів

Після вдалого калібрування, відбувається розпізнавання жестів. Результат розпізнавання відображається в центральній верхній частині зображення з відеопотоком.

На рисунку 3.5. зображено вікно відеопотоку з результатом розпізнавання в центральній верхній частині зображення.

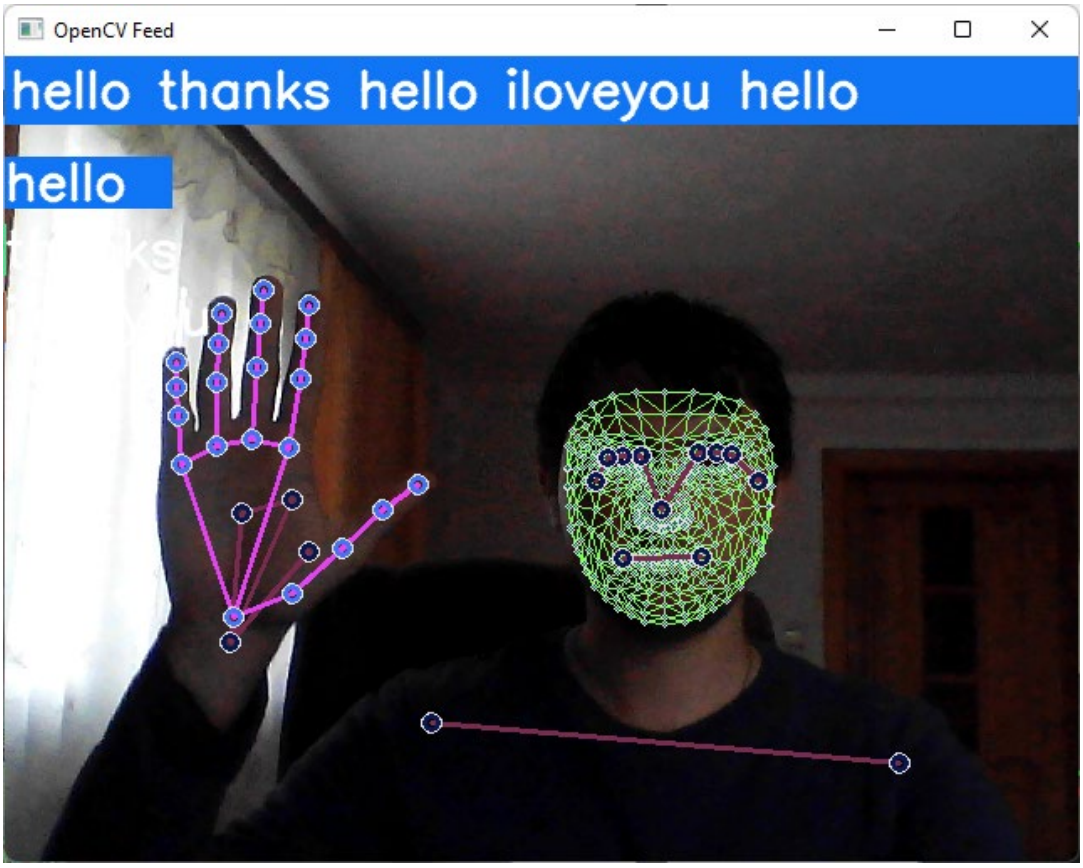


Рисунок 3.5 - Результат розпізнавання на вхідному зображенні

3.2 Розробка основних алгоритмів програмного забезпечення

У розробленому програмному забезпеченні можна виділити три окремі процедури, які мають свої алгоритми: калібрування для виділення кольору шкіри, обробка вхідного зображення відеопотоку і розпізнавання жесту на основі обробленого зображення.

Калібрування відбувається при відпусканні затиснутої кнопки 'C' на клавіатурі. Ідея така, що користувач підносить свою руку до камери, затискає кнопку 'C' і на екрані з'являються шість квадратів (рис.3.6), які відмічають регіони з яких буде братися колір шкіри, і коли користувач відпускає кнопку 'C' відбувається калібрування.

Рисунок 3.6 - Квадрати, що з'являються при калібрування кольору шкіри

Взявши шість квадратних областей на зображенні розміром 15 на 15 пікселів за вхідні дані, береться медіана значення кольору у просторі кольорів HSV із кожного з квадратів.

На основі цих значень, а також значень відхилень нижньої і верхньої границь для кожного з каналів HSV, створюється шість граничних значень кольору, відповідно до кожного з квадратів. Граничні значення кольору і є необхідними даними для виділення силуету руки.

Блок-схема алгоритму калібрування кольору шкіри зображена на рисунку 3.7.

Рисунок 3.7 - Блок-схема алгоритму знаходження граничних значень кольору шкіри

Для того, щоб в результаті створити бінарне зображення силуету руки, вхідне зображення фільтрується відсікаючи пікселі, колір яких не потрапляє в визначені граничні значення, тобто з вхідного зображення обираються лише пікселі що потрапляють в граничні значення.

Відсікання пікселів, що не потрапляють в граничні значення кольору шкіри, виконується за допомогою функції бібліотеки OpenCV cv2.inRange. Ця функція повертає бінарне зображення, на якому пікселям, що потрапляють в задані граничні значення , присвоюється значення 255 (білий колір), а усім іншим пікселям чорний колір.

Для кожного з шести граничних значень результат фільтрування зображення додається за допомогою операції побітового "або" до кінцевого результату, який зображено на рисунку 3.3. Блок-схема алгоритму створення бінарного зображення силуету руки наведена на рисунку 3.8.

В контексті бінарних зображень, що є по суті двовимірними масивами, операція побітового "або" - бінарна операція, що повертає двовимірний масив, елементи якого мають значення 255 якщо відповідний елемент у першому або другому вхідному масиві мали значення 255, а у протилежному випадку мають значення 0. Тобто, це можна розглядати як накладання зображень.

Операція побітового "або" виконується шляхом виклику функції бібліотеки OpenCV cv2.bitwise_or, яка повертає бінарне зображення, що є результатом накладання двох зображень, переданих як аргументи.

Рисунок 3.8 - Блок-схема алгоритму створення бінарного зображення силуету руки

Для розпізнавання жесту навколо контуру силуету руки будується ОО (рис. 3.9) і знаходяться дефекти опуклості.

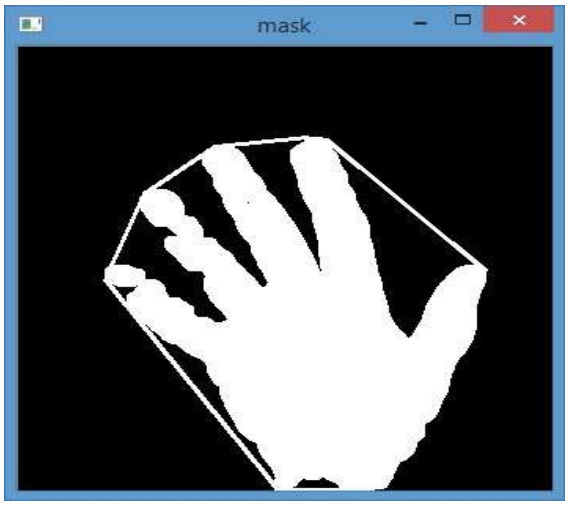


Рисунок 3.9 - ОО навколо силуету руки

Приймається що якщо кут біля дефекту опуклості менше 90 градусів, то цей дефект є точкою між пальцями. Отже на основі кількості дефектів опуклості з кутом менше 90 градусів визначається кількість показаних пальців. Наприклад, при чотирьох таких дефектах вважається що показано п'ять пальців, а при одному дефекті - два пальці.

Коли дефектів нуль, вважається що зображено жест кулака, долоні або один палець.

У момент розпізнавання публікується відповідна подія, на яку підписано обробник події, який відтворює звук. Так як загальним робочим циклом програми є зчитування зображення і обробка зображення кожного кадру з відеопотоку камери, подія розпізнавання викликається підряд під час зчитування кожного нового кадру, що викликає постійне відтворення звуків.

Для уникнення цього і відтворення звуків лише при розпізнаванні жесту, звуки відтворюються лише при розпізнаванні жесту, який відрізняється від попереднього.

РОЗДІЛ 4. РЕЗУЛЬТАТИ РОБОТИ СИСТЕМИ

4.1 Оцінка ефективності роботи системи і умов її покращення

Розроблене ПЗ при достатніх умовах освітлення і фону, захопленні усієї кисті або кисті і частини зап'ястя та позиціонуванні руки паралельно вертикальній осі камери стабільно, точно і достатньо швидко розпізнає задані сім жестів у реальному часі. Стабільне, точне і швидке розпізнавання жестів можливе при дотриманні деяких умов:

- калібрування кольору шкіри має бути вдалим;
- рука повинна знаходитись паралельно вертикальній осі камери;
- у регіон зчитування зображення для аналізу повинна потрапляти вся долоня, при чому потрапляння зап'ястя або частини передпліччя допустиме.

Вдале калібрування шкіри досягається при достатньому освітленні, правильному розташуванні руки у момент калібрування і правильно підібраних значеннях відхилення допустимого кольору шкіри. У результаті вдалого калібрування отримується чіткий контур руки (рис. 4.1). На рисунку 4.2 зображено результат невдалого калібрування, на якому присутні шуми і силует руки не виділено повністю.

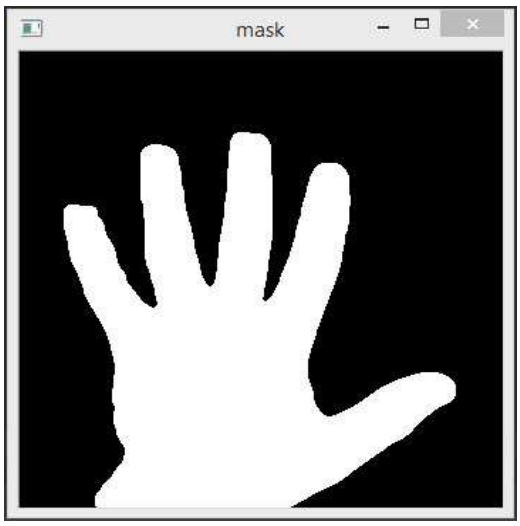


Рисунок 4.1 - Результат вдалого калібрування кольору шкіри

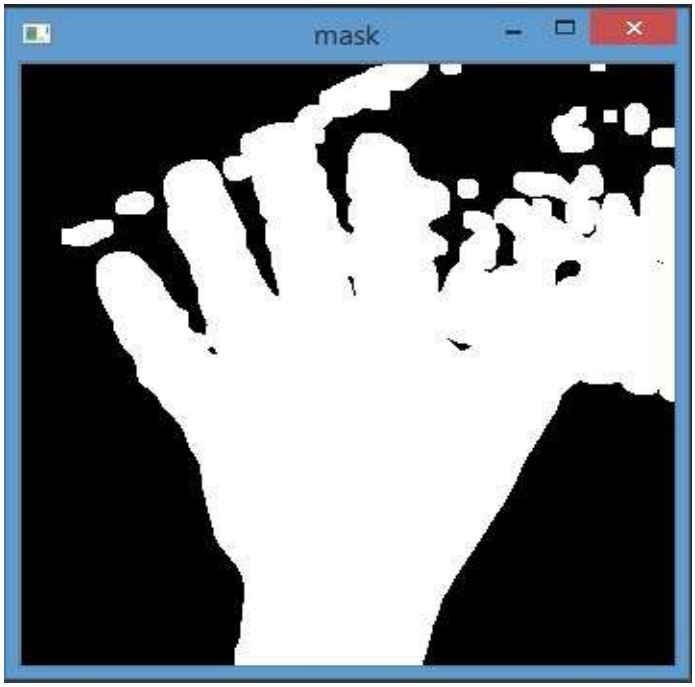


Рисунок 4.2 - Результат невдалого калібрування кольору шкіри

Із семи жестів, що вміє розпізнавати система, жести кулака і відкритої долоні вимагають вертикального положення руки, тобто положення паралельно вертикальній осі об'єктива камери. Це необхідно для того, щоб обмежуючий прямокутник, побудований навколо контуру руки, своїми сторонами правильно відбивав відношення ширини і висоти показаного жесту.

На рисунку 4.3 зображено положення руки в об'єктиві камери, яке є рекомендованим для розпізнавання усіх семи жестів.

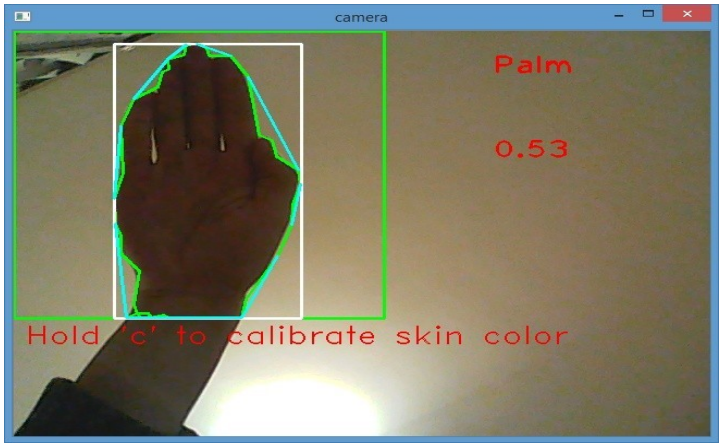


Рисунок 4.3 - Рекомендоване положення руки для розпізнавання жестів

Відтворення звуків при розпізнаванні жестів виконується без затримок у роботі системи. Після виконання певної підготовчої роботи, користувач може відтворювати прості ритми за допомогою жестів.

4.2 Особливості і обмеження розробленої системи розпізнавання жестів

При вдалому виділенні кольору руки, розроблена система не здатна здатна точно розпізнати будь-яку варіацію показаного жесту. Існують певні обмеження.

Через те, що розроблена система при розпізнаванні спирається на взаємне розташування частин руки, яка виконує жест, при певних конфігураціях вхідних даних результат не відповідає очікуваному.

По перше, при віддаленні руки від камери більше ніж на 60 см, параметр співвідношення сторін обмежуючого прямокутника контуру руки стає неактуальним В такому випадку виконується невідне розпізнавання жесту стиснутого кулака (рис. 4.4).

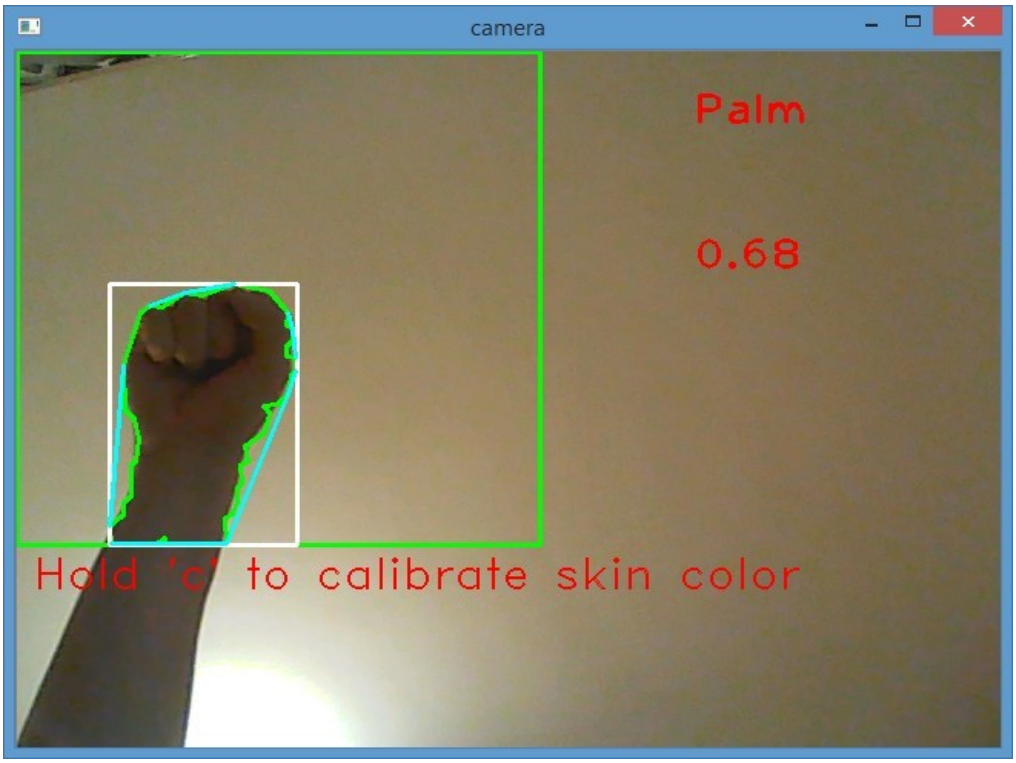


Рисунок 4.4 - Помилковий результат розпізнавання жесту кулака при віддаленні руки від камери

Оскільки для того, щоб відрізнити жест показування одного пальця від долоні або кулака, розроблена система спирається на співвідношення площі контуру руки і опуклої оболонки, то при куті обертання руки на 45 градусів при показуванні одного пальця система дає помилковий результат (рис 4.5).

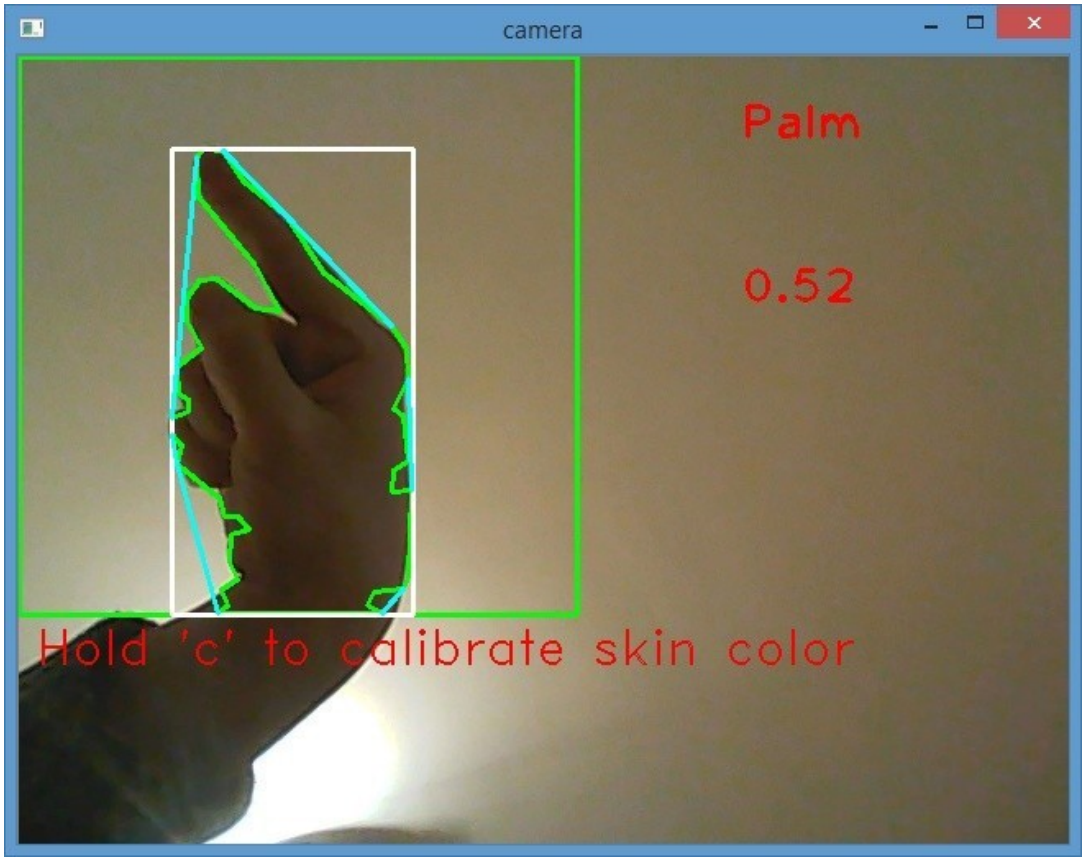


Рисунок 4.5 - Помилкове розпізнавання одного пальця при обертанні руки на 45 градусів

РОЗДІЛ 5. ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ДИПЛОМНОЇ РОБОТИ

У дипломному проекті розробляється програмне забезпечення для розпізнавання статичних жестів у реальному часі на основі визначення кольору шкіри і побудови опуклої оболонки навколо її контуру. За наявності задовільних умов освітлення, результати роботи програми отримуються в реальному часі і відрізняться високою точністю. Розпізнані жести можуть бути використані для здійснення будь-яких дій, які прив'язані до події розпізнавання у розробленій системі. У розробляємому проекті було реалізовано відтворення звуків при розпізнаванні жестів.

Існують деякі продукти, що виконують аналогічні функції. Наприклад, система Universal GestureRecognition, що розпізнає динамічні жести. Маючи беззаперечні переваги, така система має наступні недоліки в порівнянні з розробленою:

- загальна висока вартість розробки, що призводить до значної кінцевої ціни продукту;
- залучення тисяч осіб для створення бази даних для роботи системи;
- наявність потребності у продуктивних процесорах при розробці системи, що задовольняється цінними GPUплатами;
- відсутність підтримки операційної системи Windows;
- вимогливість до персонального комп'ютера кінцевого
- користувача;
- висока ціна, що складає 1450 грн. за копію;
- необхідність у великому об'ємі даних для навчання системи.

Таким чином було прийнято рішення розробити власну систему розпізнавання жестів для звичайної цифрової камери на основі безкоштовної бібліотеки OpenCV і мови програмування Python. Розроблена система повинна відрізнятися невеликої кінцевою вартістю і невибагливістю до персонального комп'ютера користувача.

Однією з переваг розробленої програми над існуючим аналогом, системою Universal Gesture Recognition, є підтримка операційної системи Windows, що дозволяє зайняти частину ринку, що складається з користувачів Windows.

Очікувана вартість складатиме 300 грн., що майже в п'ять разів нижче за вартість конкуруючого продукту і надасть можливість продати розроблену систему великій кількості користувачів.

5.1 Планування розробки системи розпізнавання статичних жестів

Весь комплекс робіт з проектування системи розпізнавання статичних жестів у реальному часі можна поділити на етапи. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. В розробці беруть участь програміст протягом 2 місяців і консультант протягом 0,2 місяця.

Тривалість робіт визначають за формулою 5.1:

$$Q = T_ц \cdot R, \quad (5.1)$$

де $T_ц$ — тривалість циклу, днів; Q – трудомісткість, людино-днів; R – кількість виконавців, чол.

Отримана інформація наведена в табл. 5.1.

Таблиця 5.1

Характеристика робіт з розробки програми

№	Найменування роботи	Трудомісткість		Виконавці	Тривалість, днів
		Люд.днів	% до підсумку		
1	Аналіз предметної області (ПО)	5	11,1%	Програміст	5
2	Аналіз, узгодження та затвердження технічного завдання	8	17,8%	Програміст Консультант	4

Продовження таблиці 5.1

3	Дослідження процесу розпізнавання жесту на основі побудови опуклої оболонки контуру	4	8,9%	Програміст	4
4	Аналіз та вибір технології	3	6,7%	Програміст	3
5	Проектування структури програми	3	6,7%	Програміст	3
6	Створення програмного коду	15	33,3%	Програміст	15
7	Тестування і відлагодження програми	5	11,1%	Програміст	5
8	Оформлення технічної документації	2	4,4%	Програміст	2
	Усього	45	100%		41

За даними табл. 5.1 складається зведений стрічковий графік Гранта для планування розробки програмного продукту, який представляє собою таблицю, в першому стовпці якої розміщені в порядку збільшення термінів початку виконання всі види роботи, а навпроти кожного з них - календарний період їх виконання.

5.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складається калькуляція кошторисної вартості робіт яка включає наступні статті:

– основна заробітна плата;

– додаткова заробітна плата;

– відрахування на єдиний соціальний внесок;

– витрати на спеціальне обладнання;

– матеріали і комплектуючі вироби;

5.2.1 Розрахунок основної заробітної плати

Витрати за цією статтею складаються з планового фонду заробітної плати всіх категорій робітників, зайнятих в розробці програми. Розрахунок заробітної плати ведеться на основі даних про трудомісткість, наведених в табл. 5.2.

Таблиця 5.2

Розрахунок основної заробітної плати

Посада виконавця	Чисельність, чол.	Місячний оклад, грн.	Кількість місяців роботи	Сума ЗП, грн.
Програміст	1	7137,00	2	14274,00
Консультант	1	9000,00	0,2	1800,00
Усього	2			16074,00

5.2.2 Розрахунок додаткової заробітної плати

Додаткову заробітну плату приймають рівною 15% від основної заробітної плати працівників і розраховують за формулою 5.2:

$$ЗП_{дод}=ЗП_{осн}*0,15 \qquad (5.2)$$

Підставивши величину основної заробітної плати в дану формулу, отримаємо:

$$ЗП_{дод}=16074*0,15=2411,10$$

5.2.3 Відрахування на єдиний соціальний внесок

Єдиний соціальний внесок визначається в процентному відношенні до суми основної і додаткової заробітної плати відповідно до встановленого нормативу на підприємстві, де виконується проект.

З 2016 року встановлена загальна ставка єдиного соціального внеску – 22%.

Розраховуємо за формулою 5.3:

$$ЄСВ=(ЗП_{осн}+ЗП_{дод})*0,22, \quad (5.3)$$

Де, $ЗП_{осн}$ – основна заробітна плата, грн.;

$ЗП_{дод}$ – додаткова заробітна плата, грн.

Отже, відрахування на єдиний соціальний внесок дорівнюють (16074 + 2411,10) * 0,22 = 4066,72 грн.

5.2.4 Визначення витрат на матеріали

До цієї статті включається вартість основних і допоміжних матеріалів (папір, модуль пам’яті USB, картридж, тонер, канцелярські товари і т.д.), необхідних для розробки проекту розпізнавання жестів.

Транспортно-заготівельні витрати приймаються рівними 3-10%

вартості матеріалів. У даній роботі транспортно-заготівельні витрати складають 5%. Розрахунок наводиться в таблиці 5.3.

Таблиця 5.3

Розрахунок витрат на матеріали

Найменування, вид, тип, марка	Одиниця виміру	Ціна за одиницю вим.,грн	Кількість од. вим.	Сума витрат, грн.
Папери для принтера, уп.	шт.	100,00	2	200,00
Тонер для принтера	шт.	50,00	1	50,00
Набір канцтоварів	шт.	50,00	3	150,00
Флеш-память	шт.	300,00	1	300,00
Разом				700,00

Транспортнозаготівельні витрати				35,00
Всього				735,00

5.2.5 Витрати на спеціальне обладнання

До цієї статті входять витрати на придбання, транспортування, монтаж, налагодження та експлуатацію спеціального устаткування, використовуваного при проектуванні для проведення розрахунків, іспитів і експериментів. До цього відносять ЕОМ, принтери, сканери, модеми й інше дороге устаткування тривалого користування (тобто основні фонди, що використовуються). Тут же враховуються витрати на оплату машинного часу вже наявних використовуваних ЕОМ.

Використовувались ноутбук Dell G3 3579, комп'ютерна миш GeniusScrollTool 200 та веб-камера Dell G3 3579.

Амортизаційні відрахування визначають за формулою:

$$A = \frac{Фб * На}{100}, \qquad (5.4)$$

де $Ф_б$ - балансова вартість устаткування, грн.;

$Н_а$ – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, %.

Балансова вартість устаткування складає 4889 + 345 + 790 = 6024грн.

Отримуємо:

$$A = 6024*0,25 = 1506 \text{ грн.}$$

Стаття "Експлуатація устаткування" включає витрати на споживану устаткуванням електроенергію і допоміжні матеріали. Визначається за формулою:

$$Се = Nн * Феф * К\,зч * К\,зп * Це, \qquad (5.5)$$

де $N_н$ –номінальна потужність ЕОМ, кВт;

$Ф_{еф}$– ефективний фонд часу роботи ЕОМ, машино-годин;

$К_{зч}$ – средний коефіцієнт завантаження по часу;

$К_{зп}$ – коефіцієнт завантаження по потужності;

$Ц_е$ – ціна однієї кВт·години електроенергії, грн./(кВт·год).

Номінальна потужність ЕОМ - 0,1 кВт.

Річний ефективний фонд часу роботи ЕОМ становить 1800 годин. Середні коефіцієнти завантаження за часом і за потужністю рівні відповідно 0,9 і 0,6.

Ціна однієї кіловат-години електроенергії складає 2,11 грн. Отримуємо:

$$C_e=0,1*1800*0,9*0,6*2,11=205,1 грн$$

Заробітна плата обслуговуючого персоналу, що складається зі спеціаліста з обслуговування ЕОМ, розраховується за формулою:

$$ЗП\ обсл=\frac{ФЗП\ р*(1+Кєсв)*tобсл}{Феф.обсл} \quad (5.6)$$

де $ФЗП_р$ — річний фонд заробітної плати (основної і додаткової) обслуговуючого працівника, грн.;

$К_{єсв}$ — коефіцієнт, що враховує виплати на Єдиний соціальний внесок; $t_{обсл}$ — час необхідний на обслуговування ЕОМ протягом року,

годин/рік;

$Ф_{еф.обсл}$ — річний ефективний фонд часу обслуговуючого персоналу.

Місячна заробітна плата обслуговуючого персоналу становить 3723 грн., а річний фонд заробітної плати відповідно дорівнює 44676 грн. Річний ефективний фонд робочого часу обслуговуючого ЕОМ працівника дорівнює 1200 год / рік. На обслуговування одного ПК витрачається по 1 годині на місяць, що в рік становить 12 годин. Отримуємо:

$$ЗП_{обсл}=\frac{44676*(1+0,22)*12}{1200}=545 грн.$$

Стаття «Поточний ремонт обладнання» приймається рівною 3% від балансової вартості обладнання і складає 180,72 грн.

Стаття «Інші витрати» приймається рівною п'яти відсоткам від суми всіх попередніх статей витрат на утримання і експлуатацію обладнання.

Сума всіх попередніх статей дорівнює 2436,82 грн., 5% від суми складають 121,84 грн.

Розраховані статті витрат на утримання і експлуатацію обладнання внесені в табл. 5.4.

Таблиця 5.4

Кошторис витрат на утримання та експлуатацію устаткування

Стаття витрат	Сума, грн.
Амортизація устаткування	1506
Експлуатація устаткування (крім витрат на поточний ремонт)	205,1
Заробітна плата основна і додаткова обслуговуючого робітника з відрахуваннями на ЄСВ	545
Поточний ремонт обладнання	180,72
Інші витрати	121,84
Усього	2558,66

Витрати на оплату машинного часу для написання і відлагодження даної програми визначаються по формулі 5.7.

$$Смо=Рекс*tмо \qquad (5.7)$$

де $С_{мо}$ — витрати на оплату машинного часу, грн.;

$Р_{екс}$ — експлуатаційні витрати на одну годину машинного часу для даної ЕОМ, грн/машино-година; $t_{мо}$ — машинний час ЕОМ для написання і відлагодження данного

програмного продукту, машино-годин.

Експлуатаційні витрати на одну годину машинного часу використовуваної ЕОМ розраховують діленням суми витрат за кошторисом «Витрати на утримання та експлуатацію обладнання (ЕОМ)» (табл. 5.5) на річний ефективний фонд часу

роботи ЕОМ. Річний ефективний фонд часу роботи ЕОМ дорівнює 1800 годин. В результаті експлуатаційні витрати на одну годину машинного часу дорівнюють:

$$P_{екс} = 2558,66 / 1800 = 1,42$$ грн/машино-година.

ЕОМ експлуатується 40 днів в одну зміну, що становить в сумі 320 годин. Таким чином, витрати на оплату машинного часу складуть:

$$C_{мо}=1,42*320=454,4 \text{ грн.}$$

5.2.6 Розрахунок накладних витрат

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямим шляхом - у відсотках до основної заробітної плати розробників. В даному випадку накладні витрати становлять 45% до основної заробітної плати розробників, що складає 7233,30 грн.

Результати визначення витрат на розробку програми у вигляді калькуляції кошторисної вартості робіт наведені в таблиці 5.6.

Таблиця 5.6

Калькуляція кошторисної вартості робіт

№	Найменування статті витрат	Сума, грн	Питома вага до підсумку, %
1	Основна заробітна плата	16074	52
2	Додаткова заробітна плата	2411,10	7,8
3	Відрахування до ЄСВ	4066,72	13,13
4	Матеріали і комплектуючі	735	2,4
5	Витрати на спец. обладнання	454,4	1,46
6	Накладні витрати	7233,3	23,3
	Разом витрати ($S_{рп}$)	30974,52	100

	Прибуток (30 %)	9292,36	
	Разом	40266,88	
	ПДВ (20 %)	8053,38	
	Усього	48320,26	

5.3 Розрахунок економічної ефективності програмного продукту

Розроблена система відрізняється невисокою кінцевою вартістю, легкістю у користуванні і невибагливістю до ресурсів ЕОМ користувача. Разом з цими перевагами, головною перевагою над існуючим аналогом, системою Universal Gesture Recognition, є підтримка операційної системи Windows, що дозволяє зайняти частину ринку, що складається з користувачів Windows.

Враховуючи поточні дані про розподіл ринку операційних систем, Windows займає 37,42% ринку. Це значна частина користувачів і розроблена система розпізнавання жестів може користуватися великим попитом.

Очікується, що 1000 користувачів у світі придбають програму після першого року запуску. Це принесе дохід у 300000 грн. Враховуючи ризик у 50%, отримуємо прогнозований дохід у 150000 грн.

Оскільки дохід від продажу суттєво перевищує ціну програмного забезпечення (48320,26 грн.), у якій було враховано рентабельність у 30%, то розробка є економічно доцільною.

ВИСНОВКИ

В результаті виконання роботи було розроблено ПЗ для розпізнавання жестів з використанням веб-камери. Створений програмний продукт дозволяє використовувати веб-камеру для виділення ділянки руки і розпізнавання наступних жестів: Так, Ні (тобто згоден або не згоден), Я кохаю тебе, алвафітні жести, Дякую, Надобраніч, Добрий день.

Виділення регіону зображення, що відповідає руці, було реалізовано шляхом калібрування допустимих граничних значень кольору шкіри у просторі кольорів RGB. Калібрування ініціюється користувачем і надає змогу ефективно виділяти регіон руки за різних умов освітлення з використанням платформи MediaPipe.

Основою методу розпізнавання є побудова ОО навколо контуру руки і знаходження дефектів опуклості, що знаходяться між пальцями руки шляхом вибору лише дефектів, у яких кут між початком дефекту і кінцем дефекту з вершиною у найвіддаленішій точці дефекту має градусну міру менше або рівну 90 градусів.

У роботі була проаналізована предметна область, були розглянуті різні підходи до вирішення поставленого завдання, вже існуючі аналоги.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review [Text] / [V. I. Pavlovic, R. Sharma, T. S. Huang] // IEEE Transactions on Patter Analysis and Machine Intelligence. - 1997.-Vol. 19, №7. - P.1

2. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review [Text] / [V. I. Pavlovic, R. Sharma, T. S. Huang] // IEEE Transactions on Patter Analysis and Machine Intelligence. - 1997.-Vol. 19, №7. - P.2

3. Gesture Recognition: The Gesture Segmentation Problem [Text] / [M. K. ViblisK. J. Kyriakopoulos] // Journal of Intelligent and Robotic Systems. - 2000.-Vol. 28, №1. - P.1

4. Gesture Recognition: The Gesture Segmentation Problem [Text] / [M. K. ViblisK. J. Kyriakopoulos] // Journal of Intelligent and Robotic Systems. 2000.-Vol. 28, №1. - P.2-6

5. A hand gesture interface device: project report (final) / VPL Research,Inc.;T. G. Zimmerman, J. Lanier, C. Blanchard et al. - Redwood CityCA, 1987. - 4 p.

6. OpenCV API Reference [Electronic resource]. - Access mode: https://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html

7. A linear time convex hull algorithm for simple polygons: project report(final) / Department of Computer Science, University of Maryland; C.E. Kim. College Park, Maryland, 1980. - 18 p.

8. EllipticLabs Products [Electronic resource]. - Access mode: http://www.ellipticlabs.com/how-it-works

9. ArcSoftGesture Recognition Technology [Electronic resource]. - Accessmode: http://www.arcsoft.com/technology/gesture.html

10 .MediaPipe Hands [Електронний ресурс] – MediaPipe, 2020 – Режим доступу: https://google.github.io/mediapipe/solutions/hands

11. Simulataneously detecting face, hand motion, and pose in real-time on mobile devices [Електронний ресурс] – Heartbeat, 2021 – Режим доступу:

https://heartbeat.fritz.ai/simultaneously-detecting-face-hand-motion-and-pose-inreal-time-on-mobile-devices-27849560fc4e

12. Understanding SSD MultiBox – Real-Time Object Detection In Deep Learning [Електронний ресурс] – towards data science, 2017 – Режим доступу: https://towardsdatascience.com/understanding-ssd-multibox-real-time-objectdetection-in-deep-learning-495ef744fab

13. VGG16 – Convolutional Network for Classification and Detection [Електронний ресурс] – Neurohive, 2018 – Режим доступу: https://neurohive.io/en/popular-networks/vgg16/

14. Bradski G., Kaehler A. Learning OpenCV - Computer Vision with the OpenCV Library — O'Reilly Media — 2008 — 580 с.

15. Convolutional Neural Networks [Електронний ресурс] – Ai in Radioogy, 2018 – Режим доступу: https://www.doc.ic.ac.uk/~jce317/introduction-cnns.html

ДОДАТОК А. ТЕКСТ ПРОГРАМИ

```
cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.
5) as holistic:

    # NEW LOOP
    # Loop through actions
    for action in actions:
        # Loop through sequences aka videos
        for sequence in range(no_sequences):
            # Loop through video length aka sequence length
            for frame_num in range(sequence_length):

                # Read feed
                ret, frame = cap.read()

                # Make detections
                image, results = mediapipe_detection(frame, holistic)
                print(results)

                # Draw landmarks
                draw_styled_landmarks(image, results)

                # NEW Apply wait logic
                if frame_num == 0:
                    cv2.putText(image, 'STARTING COLLECTION', (120,200),
                                cv2.FONT_HERSHEY_SIMPLEX, 1, (0,255, 0), 4, cv2.LINE_AA)
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(acti
on, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_A
A)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
                    cv2.waitKey(2000)
                else:
                    cv2.putText(image, 'Collecting frames for {} Video Number {}'.format(acti
on, sequence), (15,12),
                                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 1, cv2.LINE_A
A)
                    # Show to screen
                    cv2.imshow('OpenCV Feed', image)
```

```
            # NEW Export keypoints
            keypoints = extract_keypoints(results)
            npy_path = os.path.join(DATA_PATH, action, str(sequence), str(frame_num)
)
            np.save(npy_path, keypoints)

            # Break gracefully
            if cv2.waitKey(10) & 0xFF == ord('q'):
                break

    cap.release()
    cv2.destroyAllWindows()


# 1. New detection variables
sequence = []
sentence = []
threshold = 0.8

cap = cv2.VideoCapture(0)
# Set mediapipe model
with mp_holistic.Holistic(min_detection_confidence=0.5, min_tracking_confidence=0.
5) as holistic:
    while cap.isOpened():

        # Read feed
        ret, frame = cap.read()

        # Make detections
        image, results = mediapipe_detection(frame, holistic)
        print(results)

        # Draw landmarks
        draw_styled_landmarks(image, results)

        # 2. Prediction logic
        keypoints = extract_keypoints(results)
#         sequence.insert(0,keypoints)
#         sequence = sequence[:30]
        sequence.append(keypoints)
        sequence = sequence[-30:]

        if len(sequence) == 30:
            res = model.predict(np.expand_dims(sequence, axis=0))[0]
            print(actions[np.argmax(res)])
```

```
        #3. Viz logic
            if res[np.argmax(res)] > threshold:
                if len(sentence) > 0:
                    if actions[np.argmax(res)] != sentence[-1]:
                        sentence.append(actions[np.argmax(res)])
                else:
                    sentence.append(actions[np.argmax(res)])

            if len(sentence) > 5:
                sentence = sentence[-5:]

            # Viz probabilities
            # image = prob_viz(res, actions, image, colors)

        cv2.rectangle(image, (0,0), (640, 40), (245, 117, 16), -1)
        cv2.putText(image, ' '.join(sentence), (3,30),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

        # Show to screen
        cv2.imshow('OpenCV Feed', image)

        # Break gracefully
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()


model = Sequential()
model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(30,1662)
))
model.add(LSTM(128, return_sequences=True, activation='relu'))
model.add(LSTM(64, return_sequences=False, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(actions.shape[0], activation='softmax'))
```